CONVERSANT® 1 VOICE SYSTEM: ARCHITECTURE AND APPLICATIONS

Robert J. Perdue and Eugene L. Rissanen

Robert J. Perdue *is a supervisor in the Conversant Systems Development Department at AT&T Bell Laboratories in Columbus, Ohio. He is responsible for developing advanced speech and signal processing systems for network and commercial customers. He received both M.S.E.E. and B.S.E.E. degrees from Massachusetts Institute of Technology in 1973.* ***Eugene L. Rissanen*** *is a member of the technical staff in the Conversant Systems Development Department at Bell Laboratories in Columbus. He is developing advanced speech capabilities for the Conversant® 1 voice system. He joined AT&T in 1968. He received a B.S.E.E. degree from Michigan Technological University in 1968, an M.S.E.E. degree from Columbia University in 1970, and a Ph.D. in electrical engineering from Ohio State University in 1976.*

A flexible and cost-effective system for introducing speech technology into the marketplace is described. The system answers telephone calls from lines and trunks and provides verbal prompts and information to the caller. The caller can request specific information and direct the transaction by using either touch-tone or voice input. Requested information is obtained either from a small data base on the system disk or from an external source. Other system functions include recognition of isolated words and connected digit strings, recording and encoding of phrases and announcements for playback during a transaction, and call transfer and origination. A special interpreted language for specifying the transaction dialog allows diverse applications of the system to be easily written and modified.

A New Venture

There exists at AT&T a tremendous knowledge base in the broad areas of speech and signal processing, telecommunications, language understanding, artificial intelligence, and systems development. Advancing technology has recently made it possible to put this knowledge to work in salable speech products and services. In 1985, AT&T formed a new venture, Conversant Systems, chartered to research the marketplace and develop speech systems to satisfy customer needs. The venture consists of colocated people from AT&T Technology Systems and AT&T Bell Laboratories, who together have expertise in marketing, development, and manufacturing.

Our market research has provided a broad perspective on available products, customer preferences, and areas where new markets could be created. For example, a significant market exists for automating telephone attendant activities in information access systems.[1,2] Desired functions of these attendant automation products include speech and touch-tone recognition, speech encoding and storage, playback of coded speech, speaker verification, and telephone call origination and handling.

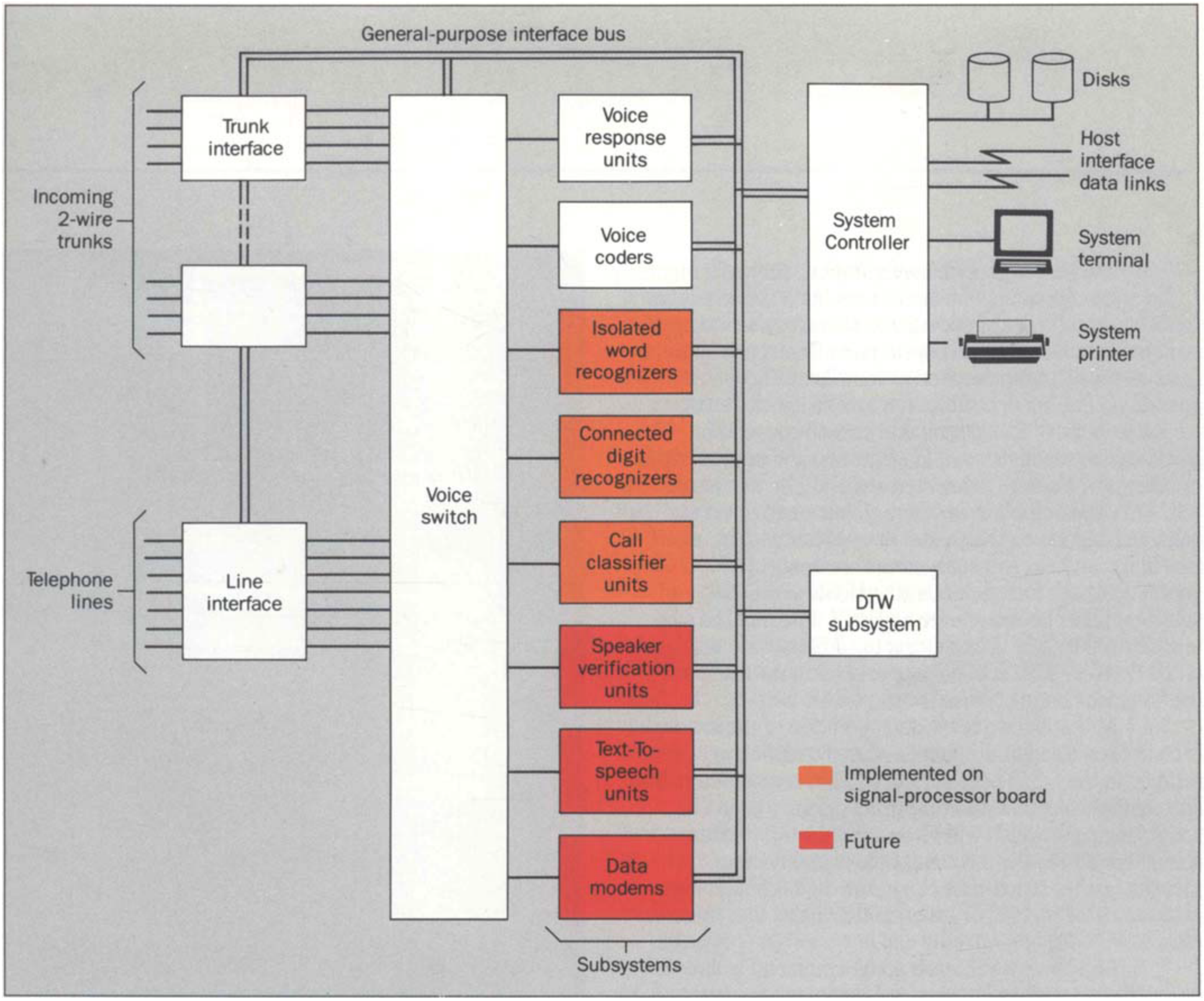

Figure 1. General architecture of the Conversant 1 voice system.

On the basis of these market studies, the Conversant® 1 voice system was designed. The system was made flexible enough to cover a wide spectrum of call-handling applications in a cost-effective manner. A large number of calls can be handled concurrently. A centralized switch allows various subsystems, such as speech recognizers and voice response units, to be shared among calls. To allow customization for particular applications, the transaction dialog is written in a special language that is interpreted in real time. Speech processing algorithms, subsystems, and other hardware already available have been relied upon heavily.[3-13] Where necessary new algorithms and subsystems have been developed.

System Architecture and Operation

An underlying goal of the system architecture is that it accommodate a large variety of applications with a minimum of adaptation effort. This goal is reflected in both its hardware configuration, as described here, and its generic and application-specific software, as described later in this article.

In its most general configuration, shown in Figure 1, the system consists of a set of telephone trunk and line interface units, a set of specialized speech subsystems, a switch to connect these subsystems to the trunks and lines, and a system controller to coordinate the whole operation. The speech subsystems, which include voice response units, voice coders, and speech recognition units, can be readily shared among calls, and several types of subsystems can be connected sequentially to a single call. The type and number of subsystems and interface units will depend on the particular application. The interface units, switch, and subsystems are connected to a general-purpose interface bus (the IEEE 488 GPIB) and will sometimes be referred to as GPIB devices. The system controller (SC) uses the bus to communicate with the GPIB devices. The SC also supports optional data links to one or more host machines.

At the highest level, the operation of the system is controlled by application-dependent "transaction scripts" residing in the SC. These scripts contain commands which specify the dialog and flow of events during a call. The script language, which will be described later, includes commands to answer incoming calls, speak various phrases, gather touch-tone or spoken digits, obtain data items from an internal or external data base, and transfer calls.

At a lower level, each script command is decomposed into a set of tasks when the command is executed. For example, a command to speak a particular phrase requires the following tasks:

- Parsing the phrase into words or subphrases
- Determining if the voice response unit (VRU) serving the call has the needed subphrases in its cache
- Downloading subphrases not already in the VRU cache
- Sending a message to the VRU specifying the subphrases to be played
- Awaiting a completion message from the VRU.

While awaiting the VRU completion message, the script interpreter program will execute script commands for other calls in progress. Hence, many calls can be handled simultaneously and independently in a multitasking manner.

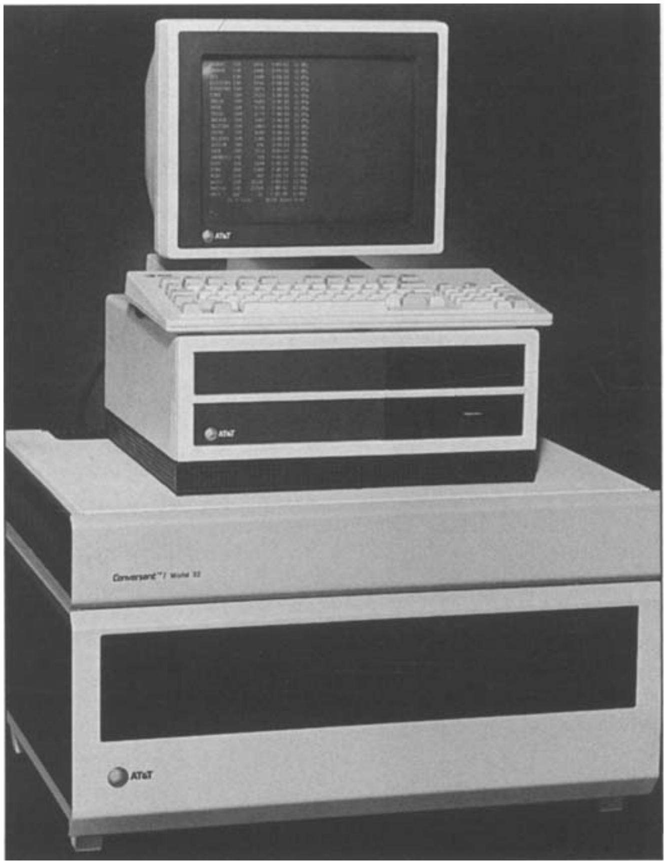

Figure 2. The Conversant 1 voice system, model 32.

Depending on the script commands, different subsystems on the voice switch can be connected to the call, as needed. The script can also obtain application-dependent data via a data interface process.

Variations of the general configuration of Figure 1 have been used. For example, in many voice response applications, a VRU must be dedicated to each call for the

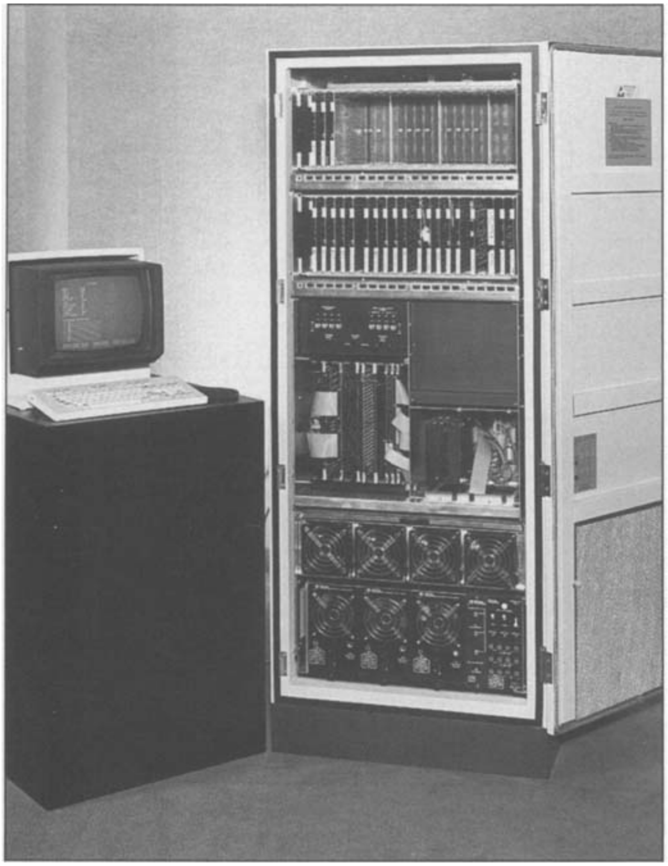

Figure 3. The Conversant 1 voice system, model 80.

entire duration of the call. Hence, one of our standard configurations has each trunk interface unit wired to a dedicated VRU. The voice switch is used to connect additional subsystems to the call.

Physical System Design. There are currently two models of the Conversant 1 voice system, model 32 and model 80 (Figures 2 and 3). The main difference between them is the SC and the physical housing. The call-handling capabilities of both are functionally equivalent.

The model 32 uses an AT&T PC6300 Plus computer as its SC while the model 80 SC is based on a commercial single-board computer. With redundant power supplies, alarm systems, and automatic reboot capability, the model 80 is suited for larger applications that demand an extra margin of system availability.

Both models use AT&T Fastech® backplane technology for the GPIB devices. The model 80 packs its GPIB devices and SC on the shelves of a tall floor-standing cabinet, whereas the model 32 consists of the PC6300 Plus and one or more single-shelf tabletop cabinets for the GPIB devices.

The System Controller. For simplicity, only the model 32 system controller is described here. The PC6300 Plus was chosen because of its availability, UNIX environment, GPIB compatibility, and low cost. The PC6300 Plus is based on the Intel 80286 processor and comes equipped with 1 megabyte of memory, a 20-Mbyte hard disk, a 1.2-Mbyte floppy disk, a GPIB interface board, and a 4800-baud asynchronous data port.[13]

Depending on the specific application, optional features that will be used include:

- An additional megabyte of memory with future expansion to 7 Mbyte
- A faster and larger (40, 67, or 135 Mbyte) hard disk for the speech phrases and any internal application data bases
- A streaming tape drive for program backups
- An additional GPIB interface board to access subsystems
- Four-port asynchronous communication boards for host data !inks
- Synchronous protocol communication boards for host data links
- A printer.

GPIB Device Descriptions. A short functional description of several GPIB devices is given in this section. Each device consists generally of a single circuit board.

Trunk interface circuits. Each trunk interface circuit (TIC) can handle up to four 2-wire incoming trunks from a central office. Currently, ground start and direct inward dialing reverse battery signaling systems are accommodated.

Voice response units. Each voice response unit (VRU) can handle up to four simultaneous calls. The VRU decodes and plays either multipulse linear prediction coded (MPLPC) speech or sub-band coded speech, as described under "Speech Synthesis Capability." An on-board cache can hold up to 400 seconds of MPLPC speech. Phrases not in cache can be downloaded from the SC disk in real time. The VRU also detects touch-tone signals.

Line interface units. Each line interface unit (LIU) can initiate calls or receive calls on up to six ordinary dial pulse or touch-tone telephone lines. Calls are initiated or answered via commands from the SC.

Voice coders. A voice coder (VC) provides real-time encoding of the caller's speech for storage and later playback through a VRU. This allows features such as entering announcements and voice messaging to be implemented.

Signal processor. The signal processor (SP) is a general-purpose programmable subsystem containing an MC68000 computer, a 2-Mbyte memory, and up to four DSP32 signal processor chips. Each SP can handle up to two simultaneous calls. Programs for the SP enable it to perform speaker-independent speech recognition (both connected digits and isolated words), speaker verification, and originating call classification, as described later in this article.

Dynamic time warper subsystem. Speech recognition is accomplished by matching (or comparing) the unknown spoken word to stored templates of known words. In order to provide a timely response to the speaker, the SP speech recognizer by itself is limited to about 200 to 250 template comparisons. The dynamic time warper (DTW) subsystem can be used to extend the number of template matches possible in real time to about 4000.

The DTW subsystem consists of a controller board and eight boards for doing 32 dynamic time warps and template matches in parallel. The results of the comparison are returned to the SP speech recognizer that requested the job. A single DTW subsystem is shared among all speech recognizers.

System Controller Software

Except for the transaction scripts and optional data-base-access modules, the SC software is generic, and consists of the UNIX™ operating system, several permanent processes, and some special commands for maintenance and diagnostics, system administration, and system usage statistics. As depicted in Figure 4, the permanent UNIX processes include:

- A transaction state machine to run transactions as specified in the transaction script
- A GPIB interrupt handler to field device requests for service
- A GPIB output process to send commands to the devices
- A data interface process to access local and/or host data bases
- A call data handler to generate and store statistics on processed calls
- An error tracker to collect and report various hardware and software errors
- A maintenance process to run on-demand and automatic diagnostics on the subsystems.

An application is built upon the generic software in three basic steps:

1. Define the dialog and write the transaction script.
2. Define any data request messages and write the application layer subroutines needed to access data bases.
3. Record and encode the speech for any phrases to be spoken to the caller.

The first two steps are discussed in the remainder of this section and phrase encoding is covered in the section, "Speech Synthesis Capability."

Transaction Scripts. The transaction state machine (TSM) controls the transactions in progress by executing commands of a transaction script written for a particular application. TSM can handle many calls, i.e., run many transaction scripts concurrently. Since the transaction

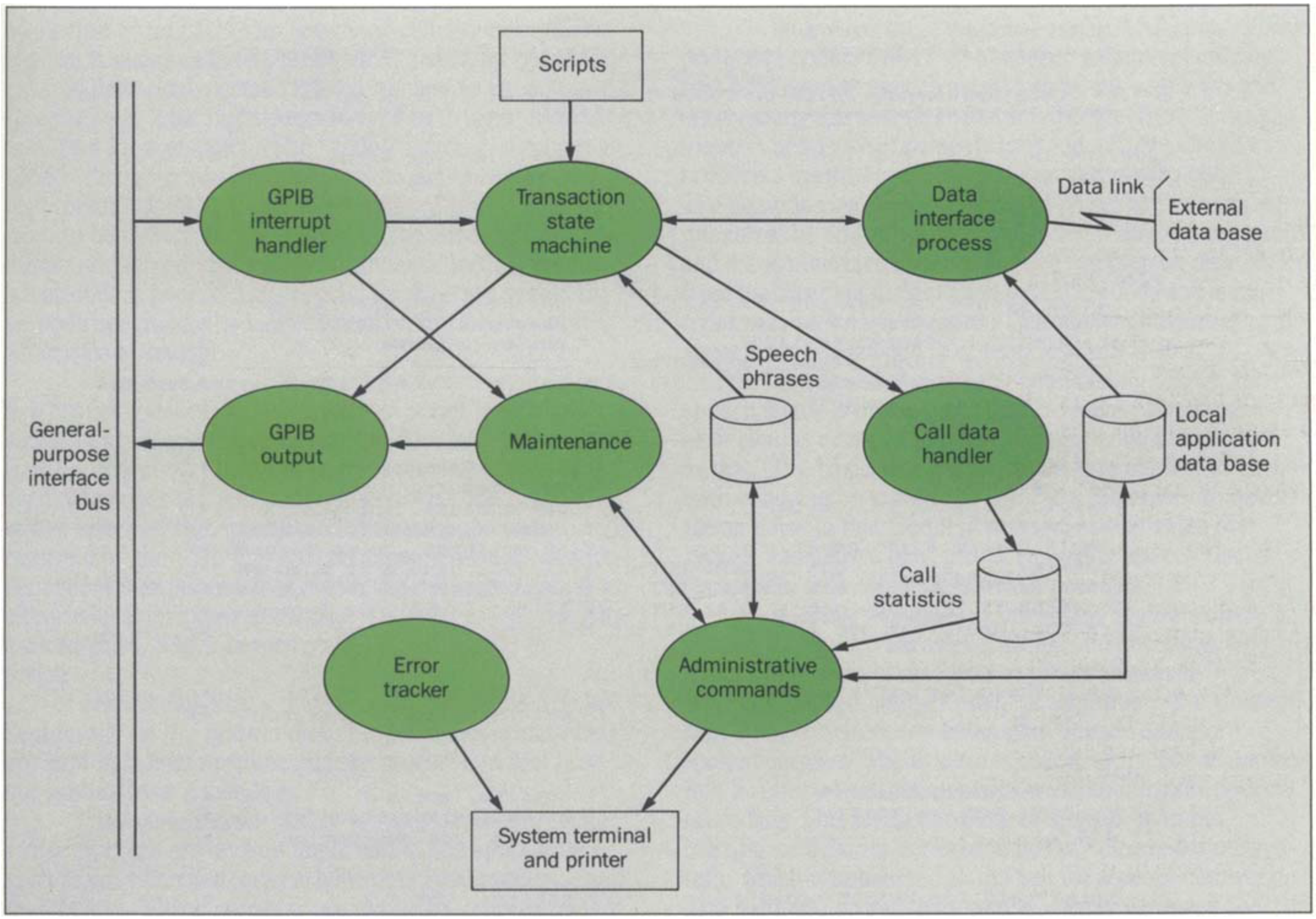

Figure 4. System controller software for the Conversant 1 system.

scripts may be different, several independent applications can be implemented on a single system. The script to be run on a particular call is normally determined by either the trunk the call uses or the direct inward dialing digits forwarded to our system by the central office.

An annotated example of a simple transaction script for a call routing application is shown in Panel 1. The panel represents a sample script for airline reservations. The script is essentially a set of instructions that tell the system how to carry out transactions. The column at the right of the panel contains the script writer's comments on the instructions.

In this example, it is assumed that the system is

Panel 1: Sample Script

A sample script for an airline reservation application is shown below. The script is essentially a set of instructions that tell the system how to carry out transactions. The right-hand column contains the script writer's comments on instructions.

```
   /****************************************************************/
   /*                        Sample Script                         */
   /*  Call Routing by VOICE or TOUCH-TONE Entry of 1, 2, 3 or 4   */
   /****************************************************************/

#define IN           0                    /* location for input        */
#define EXT          2                    /* location for extension    */
#define VOICE_OR_TT 16                    /* for getdig, voice or TT   */
tfile("list.callrt")                      /* call routing phrases      */


PLAY_MENU:
    talk("thanks for calling" "sil250")   /* phrase & 250 msec silence  */
    talk("please speak or press")         /* phrase & no silence        */
    talk("1 for reserv" "sil250")         /* phrase & pause             */
    talk("2 for flight sched" "sil250")   /* ditto                      */
    talk("3 for freight info" "sil250")   /* ditto                      */
    talk("or 4 for assist")

GET_INPUT:
    addeq('s')                            /* attach speech recognizer   */
    tttime(5, 5)                          /* set timouts                */
    getdig(VOICE_OR_TT, ch.IN, 1)         /* get 1 digit, into ch.IN    */
    dropeq('s')                           /* release speech recognizer  */
    jmp(r.0 < im.0, no_good)              /* if no input, go to no_good */
    case(ch.IN, im.'1', got_1, DIAL_OUT)  /* if "1", handle it & go on  */
    case(ch.IN, im.'2', got_2, DIAL_OUT)  /* if "2", handle it & go on  */
    case(ch.IN, im.'3', got_3, DIAL_OUT)  /* if "3", handle it & go on  */
    /* got input, but not 1, 2 or 3; treat the same as no input         */

  no_good:
    talk("hold for assistance")           /* no good input              */
    load(ch.EXT, im.'0174')               /* extension for assistance   */
    goto DIAL_OUT

  got_1:
    talk("hold for reserv")               /* confirm, got a 1           */
    load(ch.EXT, im.'0171')               /* extension for reservations */
    rts()                                 /* return from subroutine     */

  got_2:
    talk("hold for flight sched")         /* confirm, got a 2           */
    load(ch.EXT, im.'0172')               /* extension for schedules    */
    rts()

  got_3:
    talk("hold for freight info")         /* confirm, got a 3           */
    load(ch.EXT, im.'0173')               /* extension for freight      */
    rts()

DIAL_OUT:
    liu('f')                              /* flash the line             */
    liu('d', ch.EXT)                      /* dial the extension         */
    quit()
```

connected by an LIU to an automatic call director (ACD) and can transfer calls to specific ACD attendant pools. The caller is prompted via the "talk" command to enter: 1 for reservations, 2 for flight schedules, 3 for freight information, or 4 for assistance. The "addeq" command causes a speech recognizer to be connected to the call. The "getdig" command asks for one touch-tone digit or one spoken word to be collected. Depending on the received digit, the call is transferred via the "liu" commands to the appropriate attendant pool. If 1, 2, or 3 is not detected within the 5 seconds specified, the call is transferred to the general assistance attendant.

Data Base Access. Many transactions require access to a changing data base. The "dbase" script command sends a data request message to the data interface process (DIP) of Figure 4. Since the contents and structure of such data bases are application-dependent, the script writer specifies the structure of the messages sent to and returned by the DIP. The DIP contains several application-dependent subroutines that obtain the requested data from either a local or a host data base. The DIP passes the data back to TSM, which in turn makes it available to the script.

Small data bases that are relatively stable can be kept locally on the system disk. Larger dynamic data bases are kept by a host machine and are accessed in real time by our system over a data link.

Measurements. A facility is available for the script writer to count any events important to the application, such as calls handled successfully, data base requests, and user errors. These counters are essential for determining system usage and performance.

Speech Synthesis Capability

The phrases spoken by the VRU during a transaction are concatenations of one or more recorded phrase files. In order to conserve disk space, encoding techniques are used to compress the recorded speech. Two encoding algorithms being used are described briefly below and more thoroughly in References 6 and 7.

Multi-Pulse Linear Predictive Coding. Multi-pulse linear predictive coding (MPLPC) of speech is implemented by first breaking the speech signal into 10-ms segments and determining the best filter that fits the spectrum of the segment. A sequence of pulses is then fed into the filter to produce a synthesized speech signal at the filter output. The positions and amplitudes of the pulses are adjusted to minimize the difference (the error) between the real speech and the synthesized speech. Speech can then be reproduced from the filter coefficients and pulse positions and amplitudes saved for each segment. With this technique, quality speech can be encoded at rates as low as 9.6 kb/s.

Sub-Band Coding. In sub-band coding (SBC), the speech signal is divided into four frequency bands, and each band is encoded in adaptive pulse-code modulation format. The basic idea is that the bit rate needed to encode each band can be tailored to the ear's sensitivity to quantization noise in that frequency range. For bit rates of 16 kb/s and above, SBC produces quality speech that is compatible with other AT&T voice services.

Encoding Procedures. Conversant Systems offers a service for speech phrase recording and encoding. Customers may send phrase lists to be professionally recorded, edited, and encoded. Alternatively, the Conversant 1 voice system can be used to encode and store spoken phrases. The system is called with a phone number that invokes a transaction script written to handle speech recording. The actual encoding of speech, in either MPLPC or SBC, is performed by the voice coder subsystem, which is connected to the call via a script command. The recorded speech files can be audibly reviewed and visually edited via a video terminal attached to an SC port. Edit commands currently available replay the speech and reduce leading and trailing silence.

Speech Synthesis. The encoded speech files can be played out by the VRU. The DSP20 digital signal processor chip on the VRU is programmed to convert the encoded speech back to 64-kb/s pulse-code modulation format. The digital speech signal is then converted to its final analog form.

Speech Recognition Capability

The speech recognition capability is a major attraction of the system. A limited vocabulary of spoken isolated words and connected digit strings can be recognized. Speech recognition consists of first detecting and determining the endpoint of an unknown utterance, and then classifying it as a particular word in the vocabulary. For connected digit strings, a third step involves error correction. Some details of the algorithm are given here.

When requested to recognize *N* words, the speech recognizer digitizes received speech and generates an energy contour. An endpointing algorithm identifies the intervals containing words based on a top-down analysis of the energy contour.

Each endpointed interval (unknown word) is classified by comparing it to models (or templates) of all the words in the vocabulary being used. The comparison involves a dynamic time warping procedure to account for variations in speaking rate.[3] For speaker-independent word recognition, each word must be represented by a set of templates to cover various voices and dialects. The unknown word is classified the same as the template set it most closely matches. Words that do not closely match any template set are rejected. The templates used in the system have been generated from hundreds of speech samples collected over the telephone network from diverse regions of the country.

Isolated Word Recognition. The speech recognizer subsystem can perform about 200 to 250 template matches in real time. For speaker-independent recognition, this corresponds to a vocabulary size of about 10 words. Larger vocabularies and connected digit recognition require the assistance of the DTW subsystem.

Excellent recognition accuracy has been obtained on a large and diverse telephone speech data base consisting of several control words. A large fraction of the words not correctly recognized are normally rejected. Some specific accuracy numbers are given in "Applications," below.

Connected Digit Recognition. Recognizing connected words is inherently more difficult than isolated words for two reasons: first, the endpointer is more likely to miss spoken words and interpret noise or word fragments as words; second, any misrecognized word in the string causes the entire string to be wrong. To compensate for this added degree of difficulty, the digit strings recognized by the system contain redundant digits to permit error correction. Strings of 3, 6, and 9 digits (including redundant digits) have been recognized in various applications.

Without error correction, a digit recognition accuracy, over the telephone network, of under 90 percent is obtained on our large data base. Although consistent with other similar experiments,[14] this digit accuracy is much too low for recognizing digit strings. With error correction, the effective per-digit accuracy jumps to over 99 percent. About half the strings not correctly recognized were rejected and half were misclassified. To obtain this accuracy boost, 4 redundant digits are appended to each 5-digit string, and certain resulting 9-digit strings that prove difficult to recognize are not used.

In practice, after a redundant 9-digit string is recognized at the start of a call, the decoded digits are used to label the raw digits. This allows speaker-dependent templates to be made and used for the remainder of the call. Repeating this on the second spoken digit string and so on provides a bootstrapping mechanism, causing recognition accuracy to improve considerably as the call proceeds.

Call Generation Capability

Many business systems have banks of attendants placing calls to customers. The attendants dial the phone number, listen for the call progress tones (ring, busy, etc.) and talk to the customer when the call completes. Sizable savings in attendant time can be achieved by automating the call placement. The savings would result from automatic dialing and autonomous handling of calls that do not complete. Only calls that do complete are transferred to an attendant.

To meet this need, the system is designed to place outgoing calls and transfer those calls that successfully

Table I. Financial Information Transaction Dialog

Caller	System
Dial 800 number	Answer call. “Hello, Fidelity Automated Quotation Service, please enter the nine-digit ID code.”
“494327365”	“Initials are JQP, correct?”
“Yes.”	“At 10:08 the Dow was at 1864.21, down 5.73. Stock watch list?”
“No.”	“Stock quotations?”
“Yes.”	“All quotes are delayed 15 minutes. Stock ID?”
“541364672”	“XON is at 60 1/4, down 1/4, volume 29,200. Next stock ID?”
“No.”	“Option quotations?”
“Yes.”	“All quotes are delayed 15 minutes. Stock ID?”
“697753147”	“NSM last sale was at 11 7/8. First month code?”
“795”	“Strike price code?”
“158”	“NSM August 15 calls are at 1/2. Second month code?”
“No.”	“Next stock ID?”
“No.”	“Thank you for using the Fidelity Automated Quote Service. Goodbye.”

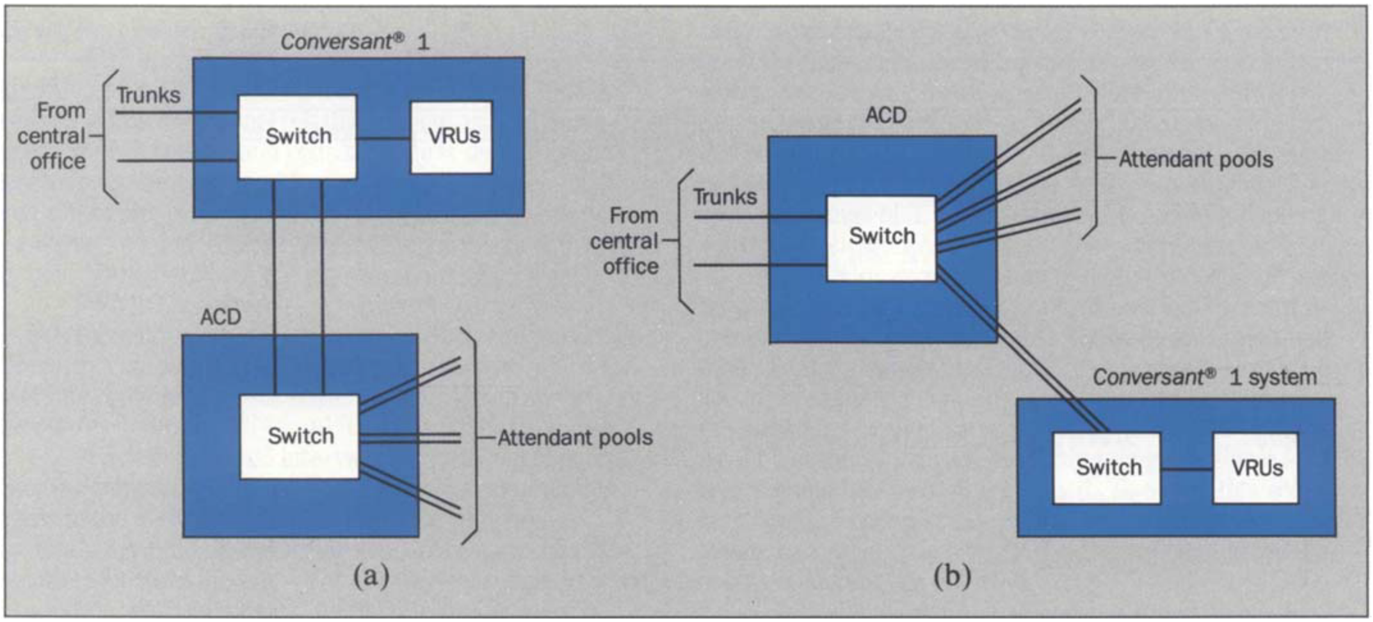

Figure 5. Interface between an automatic call director and a Conversant 1 system. (a) Voice system in front of ACD. (b) Voice system behind ACD.

complete to an attendant. This is accomplished by commanding the TIC (or LIU) to generate a call to a specified number and connecting a call classifier to the call after end of dialing. The call classifier is an SP subsystem programmed to classify the call, i.e., determine its fate, and report back to the SC. The program uses call progress tones and voice detection to classify the call in a manner similar to that currently used by the No. 2 Service Evaluation System.[15] Possible call classifications include:

- Complete—the call has been answered and voice detected.
- Announcement—the call has been answered and a network announcement detected. This can be distinguished from the "complete" classification by the special information tones (SIT) that precede announcements.
- Didn't answer—the called party's phone rang N times with no answer.
- Busy—the called party is off-hook.
- Reorder—the call encountered a congested network.
- High and dry—the call could not be properly handled by the network.

On some trunks, answer supervision is available from the central office. The TIC reports detected answer supervision directly to the SC. In these cases, the call classifier is used to distinguish types of unsuccessful attempts.

Applications

Although the capabilities and operation of the Conversant 1 system are well documented, close customer contact is considered essential in developing applications to solve customer problems. Such contact allows us to understand customer needs and to assist in developing application scripts, generating speech files, etc. Important applications are also being developed by value-added resellers. Several Conversant 1 system applications described in this section demonstrate its versatility.

Financial Information. In its trial application, the system was installed on the premises of Fidelity Brokerage Services, Inc., in Boston. It provided callers with various stock market information, such as the Dow Jones industrial stock market average and prices and volumes of requested stocks and stock options. The stock information was retrieved by the script in real time, via a data link, from a Fidelity host computer, which in turn was fed by a quote source from the major stock exchanges.

The 500 or so Fidelity customers who participated in the trial were each given a 9-digit user identification number, a catalog of 9-digit numbers for the most active 6000 stocks, a list of 3-digit month and price codes for stock options, and the 800 number of the system. In addition, customers could select up to five stocks they regularly watch. Records of user identification number and stock watch lists were kept in the host computer. The transaction works as shown in Table I. The flow of the transaction is controlled by the caller's "yes" or "no" response to questions posed by the system.

About 70 percent of the customers seemed to prefer touch-tone input over voice. The yes/no accuracy was about 97 percent. Nearly 12 percent of the time, the 9-digit strings were misspoken, i.e., the caller said a wrong digit or skipped a digit. When correctly spoken, only about 3 percent of the strings were misrecognized, another 6 percent were rejected (the caller was asked to repeat the string), and the remaining 91 percent were correctly recognized.

Call Screening and Routing. In this application, the caller is prompted for a digit and the call is routed according to the received digit, as described in the example in Panel 1. An extension of this idea is also planned, where the received digit is used to specify a particular announcement to be played. The caller is then reprompted for a second digit for routing to an attendant or an announcement, and so on.

Two basic configurations for routing screened calls are possible. In Figure 5a, calls come directly into the system, are screened, and either handled autonomously or routed to the ACD or PBX. This configuration is appropriate for applications in which the majority of incoming calls can be automatically handled by the system. The configuration of Figure 5b is desirable in situations where the majority of calls cannot be automatically handled, perhaps because of the complexity of the transaction or lack of automatic access to a data base. Here the system merely screens the call and disposes of it by transferring it to the proper attendant pool.

Outward Call Management. Other customers are using the call system's generation capability to call credit card holders with account irregularities. The system obtains a list of particular card holder phone numbers from a host computer and automatically places the calls. The progress of each call is monitored. When a call is answered, it is immediately transferred to an idle attendant and a message is sent to the host computer, which displays the person's account information on the attendant's video screen.

Calls failing to complete are handled according to the detected disposition. Calls encountering a busy signal are redialed after a few minutes. Calls encountering no answer are redialed after a much longer period. Those calls encountering network announcements—for example, calls to disconnected numbers—are not retried but reported to the host computer.

Voice Messaging. The voice coder subsystem is used to encode and store on disk a message spoken by the caller. The caller also touch-tone dials the phone numbers of the intended recipients. The system makes the outgoing calls and verbally delivers the message. To allow for mass distribution of messages, the caller can maintain a list of recipient phone numbers on the system disk.

Future Enhancements

Planned enhancements to the system include an expanded interface to the telephone network and additional subsystems. A digital T1 line interface to the system is being developed to simplify the interface to the central office and significantly reduce the amount of per-trunk hardware. Other enhancements involve adding new subsystems that can be switched onto a call. Two such subsystems with emerging markets are described below.

Speaker Verification. With speaker verification, the system will be able to ask a caller to enter an identification number and speak a special password or phrase. The entered identification will be used to access a file containing the same password or phrase spoken earlier by the caller. A comparison, similar to a template match in speech recognition, is then made between the spoken password and the one on file. If the match is close enough, the caller is assumed to indeed be the owner of the entered identification and the transaction proceeds. Otherwise, an announcement informs the caller that his or her voice does not match the entered identification. In a practical system, the caller would normally be given two or three verification chances before the system would disconnect.

The algorithm used to make the voice comparisons will be implemented on the SP board and is similar to that described in References 11 and 12.

Text to Speech. A text-to-speech subsystem would allow written information to be spoken to a caller. This eliminates the time and effort required to record and edit speech files and allows arbitrary messages to be easily added or modified. In addition, textual messages require much less disk space than spoken messages.

Algorithms that synthesize speech from text use various pronunciation rules and a large data base for converting letter sequences or phonemes to speech data. At present, text-to-speech algorithms produce somewhat unnatural sounding speech. Efforts are under way to improve the speech quality of text-to-speech systems.

Summary

The Conversant 1 voice system is the basic vehicle leading AT&T into the arena of commercial speech systems. Its flexibility in attaching various subsystems to calls allows it to incorporate new technology as it emerges. The commercial use of voice systems is still in its infancy but rapidly expanding, and expectations for its growth over the next decade are very high. AT&T intends to play a significant role in shaping this market in the coming years.

References

1. Tsutomu Takahashi et al., "The SR-2000 Voice Processor and Its Applications," *Speech Technology*, Vol. 2, No. 4, February-March 1985, pp. 22-29.
2. Lynette Gutcho, "DECtalk—A Year Later," *Speech Technology*, Vol. 3, No. 1, August-September 1985, pp. 98-102.
3. L. R. Rabiner and S. E. Levinson, "Isolated and Connected Word Recognition—Theory and Selected Applications," *IEEE Trans. on Communications*, COM-29, No. 5, May 1981, pp. 621-659.
4. J. G. Wilpon and L. R. Rabiner, "A Modified K-Means Clustering Algorithm for Use in Speaker Independent Isolated Word Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-33, No. 3, June 1985.
5. J. L. Boddie et al., "Digital Signal Processor: Architecture and Performance," *Bell System Technical Journal*, Vol. 60, No. 7, September 1981, pp. 1449-62.
6. B. S. Atal, "A New Model of LPC excitation for Producing Natural-Sounding Speech at Low Bit Rates," *Proceedings*, Intl. Conference on Acoustics, Speech, and Signal Processing, 1982, Paris, pp. 614-17.
7. R. E. Crochiere, S. A. Weber, and J. L. Flanagan, "Digital Coding of Speech in Sub-bands," *Bell System Technical Journal*, Vol. 55, No. 8, October 1976, pp.1069-1085.
8. R. E. Crochiere, R. V. Coy, and J. D. Johnson, "Real-Time Speech Coding," *IEEE Transactions on Communications*, Vol. COM-30, No. 4, April 1982, pp. 621-634.
9. J. G. Josenhans et al., "Speech Processing Applications Standards," *AT&T Technical Journal*, Vol. 65, No. 5, September/October 1986, pp. 23-33.
10. H. L. Andrews, "Speech Processing," *Computer*, Vol. 17, No. 10, October 1984, pp. 315-24.
11. Sadaoki Furui, "Cepstral Analysis Techniques for Automatic Speaker Verification," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, ASSP-29, No. 2, April 1981, pp. 254-272.
12. F. K. Soong et al., "A Vector Quantization Approach to Speaker Recognition," *Proceedings*, Intl. Conference on Acoustics, Speech, and Signal Processing, 1985, Tampa, pp. 387-390.
13. "AT&T Personal Computer 6300 PLUS, Hardware Reference Manual" and "UNIX™ System V Release 2.0 Operations

Guide," AT&T, 1985. Order numbers 845 657 048 and 845 657 758, respectively. Available by telephone order from 800 432-6600.

14. J. G. Wilpon, "A Study on the Ability to Automatically Recognize Telephone-Quality Speech From Large Customer Populations," *AT&T Technical Journal,* Vol. 64, No. 2, February 1985, pp. 423-451.
15. S. D. Hester, "Taking the Pulse of the Network", *Bell Laboratories Record,* Vol. 60, No. 3, March 1982, pp. 70-74.

(Manuscript received July 17, 1986)