t2sp Terminals Talk

Talking Terminals

published in Byte September 1982

Text-to-speech translation involves looking at the problem from a different "viewpoint."

David Stoffel
Scion Corporation
12310 Pinecrest Rd.
Reston, VA 22091

About the Author
David Stoffel has participated in the research and development of voice-response technology for six years. He has built his own talking terminal as a research tool and for his personal and professional use

Imagine for a moment that you are sitting in front of a computer video terminal working on a program when suddenly the screen goes blank. The display tube has failed. Could you continue to work on the program even though you couldn't see the screen display? That's exactly the problem that faces many visually disabled persons when they try to use standard microcomputers.

An answer to that problem is the "talking terminal." Simply, a talking terminal resembles a conventional computer terminal except that it speaks information instead of, or in addition to, displaying that information visually. This article aims to offer an understanding of the human factors involved in selecting a talking terminal and to compare current talking-terminal products.

In addition to conventional terminal capabilities, a talking terminal requires several additional features and capabilities. First, of course, the terminal must be able to talk intelligibly for you to understand its speech. So, we want to assess the intelligibility and acceptability of the product's speech. Second, speech is an elusive method of communication; once those sound waves are heard, it's up to the listener to remember what was said. So, just as many video-display terminals provide local editing and memory, a talking terminal has to provide a "say again" feature. Finally, consider, for a moment, how you would read this article aloud to someone. Would you read the punctuation as pauses, or would you say the names of the punctuation symbols? Would you pronounce acronyms, such as ASCII, or would you spell them out letter by letter? Would you read the string of digits 1234 as "one thousand, two hundred, and thirty-four," or "one, two, three, four," or use some other method? A talking terminal should be able to present the information in a variety of ways,

suited to your needs and preferences.
Today's commercially available talking-terminal products (see table 1) represent two different design strategies. The speech-related features and capabilities have either been built into an existing conventional cornputer terminal, as with the Total Talk and the FSST-3, or are in a selfcontained accessory module connected in series on the communication line between the computer and the terminal, as with the VERT. These two design strategies have sigftificant ramifications in two of the three areas of comparison: speech review and speech-parameter control.

Table 1: Manufacturers of talking terminals.

Product: VERT (Verbal Emulation In Real Time)
Self-contained speech unit connected in RS-232C communication line between computer and any terminal.
Price: $5900 (with educational discount $4990)
Manufacturer: Automated Functions Inc.
Suite 813
4545 Connecticut Ave. NW
Washington, DC 20008
(202) 362-6292

Product: Total Talk (other models are available)
Hewlett-Packard HP-2621 terminal with added speech circuItry.
Price: $4990
Manufecturer.
Maryland Computer Services Inc.
2010 Rock Spring Rd.
Forrest Hill, MD 21050
(301) 838-8888

Product.. FSST-3 (Free-Scan Speech Terminal)
ZenIth Z-19 terminal with added apeech circuitry.
Price: $4495
Manufacturer: Triformatlon System. Inc.
3132 Southeast Jay St,
Stuart. FL 33494
(305) 283-4611

Speech is an elusive method of communication; once those sound waves are heard, it's up to the listener to remember what was said.

Translation Algorithms

An exhaustive comparison of the intelligtbility and acceptability of the speech output-measures of listener comprehension and preference-requires rigorous performance measures. Such scientific evaluation is beyond my resources. Nevertheless, I can offer some useul obserrrations on the different text-to-speech

algorithms used in these talking terminals.
Though some manufacturers do not acknowledge the ancestry of the text-to-speech algorithms they use, it is reasonably safe to infer that both the VERT and Total Talk use the McIlroy (Bell I,aboratories) algorithrn, as enhanced by NIH (National Institutes of Health), and that the FSST-3 uses the NRL (Naval Research laboratory) algorithm. The Mcllroy enhanced algorithm uses about 1000 rules, and the NRL uses about 600 in performing the letter-to-phoneme or word-to-phoneme translation. (A phoneme is the smallest sound unit of speech. When we speak, we string phonemes together to produce words.)
Both algorithms are quite adequate, with translation accuracy, linguistically speaking, of about 90 percent. In my experience, I find that the Mdlroy algorithm handles difficult words correctly more often than the NRL. Neither of them makes any particularly eggious errors in the text-to-speech translation.

Choosing Synthesizers
The only viable synthesizers to date are those that use phoneme synthesis, rather than synthesis by analysis (speech encoding), because the synthesizer must be able to speak an unrestricted vocabulary. The speech-encoding synthesizers, such as Texas Instruments' TMS5221 LPC (linear-predictive coding) synthesiser or National Semiconductor's Digitalker, are still limited to fixed, prerecorded vocabularies. Both the VERT and the Total Talk use the Votrax VSB single-board speech synthesizer; while the FSST-3 uses the older Votrax VSA.
Both Votrax synthesizers are capable of independent variation in speech rate and pitch, under either manual or program control. The VERT takes advantage of the programmable-speech-rate control to enhance the pronunciation duration of very short and very long words, while also providing you with manual speech-rate and pitch controls. The Total Talk and the FSST-3 offer you manual speech-rate and pitch controls.
The Votrax VSA and VSB synthesizers seem quite similar with respect to their phoneme production, but the FSST-3, which uses the VSA, definitely sounds inferior; whether this is an artifact of the VSA synthesizer or poor audio amplification, I don't know. You may wonder why none of these products uses the new Votrax SC-1A integrated circuit, which is
less expensive. The singlequantity cost of the VSB is about $800, while the SC-01A is $70. But there are two major reasons why the SC-01A is not used. The speech-rate and pitch controls are both dependent on the same clock signal or timing circuit, affecting the ease with which intelligible speech may be produced. Also some people are concerned about the acceptability of the SC-1A's sound quality. Only scientific performance measures can determine which Votrax synthesizer is ultimately more intelligible. (For a description of an application using the Votrax SC-0IA speech-synthesizer chip see Steve Ciarcia's article on page 64 in this issue.)

Speech-Review Capabilities
Imagine that a talking terminal is reading this article to you. Suddenly, you wonder at what you just heard-either a terrible pronunciation of a proper name (like 'Ciarcia" perhaps) or maybe just a word that you don't recognize. You would like to

stop the speech, perform some review functions to repeat the last few lines or words, or spell the word in question, and then continue the speech just where you stopped it.
Stopping the speech output of a talking terminal requires that the stream of characters coming from the host computer to the terminal be halted. (Some remote computers make this very difficult.) Only the VERT attempts (when the feature is enabled) to tell the host computer not to send any more text when reviewing. The Total Talk loses data after receiving 120 characters of yet-untranslated text from the host computer. The FSST-3 loses data after accumulating 1920 characters of yet-untranslated text.
All three products allow you to review the text saved in memory. The VERT saves the most recent 12,000 characters, the Total Talk saves two screens (48 lines of 80 characters each) in the HP-2621's display memory, and FSST-3 saves from one to three screens (depending upon the amount of memory installed) in the Zenith Z-19's display memory. All three products can repeat the text in its entirety or by character, word, or line. In addition, the VERT can repeat text by phrase, sentence, or paragraph.
The Total Talk and the FSST-3 perform their review functions as a result of using the standard cursor-movement and screen-print functions of the HP-2621 and Z-19 terminals. The VERT responds with its review function to an ASCII (American Standard Code for Information Interchange) escapecode sequence from any dataterminal equipment.

The integration of speech capability with an existing, popular terminal design-the case for both the Total Talk and the FSST-3-has positive and negative consequences. Such integration negates the need to acquire a computer terminal separately when you shop for a talking terminal. On the other hand, building the speech circuitry into terminals has resulted in a performance characteristic especially annoying to programmers: both the FSST-3 and the Total Talk (Z-19 and HP-2621 terminals, respectively) never speak cursor, character-attribute, or print-function codes.
Anyone who buys a VERT must also acquire a standard computer terminal. This terminal is connected to one of the VERT's two ports, while the computer (or modem) is connected to the other. The VERT transmits all characters received from the host computer to the terminal, while translating and speaking if appropriate. The VERT can also transmit all characters received from the terminal to the host computer, though usually some are trapped as the VERT function codes. This black-box filterlike approach to the problem of providing a talking terminal is modular and well formed.

Speech Parameter Control
A talking terminal should give you the option of setting speechcontrol parameters. It should either decide the most appropriate way to translate and speak segments of text where machine-based decisions are competent or provide you with the capability of manually setting those decision parameters which cannot be successfully handled by a machine. A program can decide whether to pronounce or spell IBM, NIH, or ASCII.
The VERT uses truth tables for prefixed and suffixed letter pairs to determine whether to spell or pronounce alphabetic tokens. It is rather more difficult for a program to decide whether to say 370 as "three seven zero," "three hundred seventy," or "three seventy." If the text is referring to an IBM 370 mainframe

computer, the choice wfll be obvious to you. But a translation program has no way of "knowing" the correct pronunciation of a number or word on the basis of the context in which it was used. The Total Talk and the FSST-3 simply speak numbers digit by digit. The VERT does the same or says numbers as whole words depending on your parameter setting.

Ironically, it's often desirable to make your talking terminal remain silent, while continuing to display and save text. The reasons are many, varied, and a matter of preference, but the capability is important. Total Talk will remain silent when you depress its Silence key. The VERT can be made to remain silent until a new line, speech command, or predefined text pattern is received. The FSST-3 can start or stop speaking on cornmand.

No matter what the accuracy and proficiency of a text-to-speech translation system, there will always be words or symbols that you would like to have spoken your own way. For example, it is becoming popular in academic computer-science circles to use the word "bang" or "shriek" for the exclamation-point character (!). I am sticking with the conservative "exclamation," even though the new-comers are shorter and can be spoken more quickly. The VERT offers you the power to define, in English, your own translation preferences. You simply define a rule that says ! - 'bang," or whatever.

On the Horizon

We may see the cost of talking terminals either decrease as new speech synthesizers are used, or increase as speech capabilities are integrated with personal computers. Whatever the result, the cost of a talking terminal will remain a serious problem for visually disabled persons. Talkingterminal manufacturers should expand the market for their products-not limit it to the visually disabled. Increased sales will lower costs and benefit everyone in the long run.

One perplexing problem remains. The rapid advance of video-display technology has promoted the ever-increasing use of video dependent software. Users of talking terminals will require programmed solutions for describing essentially visual information. Unfortunately, information science is still far from providing accurate verbal descriptions of two dimensional space, thus, for instance, making it impractical to run a screen oriented program like Wordstar solely from spoken output.

Though the sound quality of available phoneme synthesizers is deiinitely far from human-sounding, I've found that visually impaired persons find it intelligible and acceptable with use. I believe that computers with natural-sounding speech and more sophisticated algorithms for translation will be achieved in this decade.E

Text-to-Speech Translation

Several independent efforts have resulted in various grapheme-to-phoneme translation systems for speech synthesis. Graphemes are letters or other characters, and phonemes are the sounds of speech. There are two approaches to the problem of translating written language (orthography) to its spoken (phonetic) form. All current efforts to create artificial speech use either one or both of these approaches.

The first approach searches a dictionary of words and/or word fragments (morphernes) for corresponding phonetic representations. Such dictionaries that are expected to satisfy a wide variety of contexts must be quite large. The software responsible for searching a dictionary must be able to account for various forms of a given entry. When dictionaries of morphemes are used, the software must be capable of separating the words to be translated into their constituent morphemes.

The second approach uses grapheme-to-phoneme translation rules. Such rules attempt to describe a correspondence between the orthographic and phonetic forms of the language. Some efforts have resulted in a combination of these two methods of translation, resorting to the second when the first fails to satisfy a translation request.

Unrestricted Text

In order to remove all restrictions on the content of the text being translated, the translation system must be able to distinguish among English words, acronyms, mnemonics, abbreviations, etc. The input stream of text to be translated is parsed into tokens that contain characters of the same type. Tokens may be divided into types alphabetic, punctuation, numeric, or symbolic. A token is complete when a character in the input stream of another token type is encountered. The type of a token determines the classification of nales used in translating the token. The selection of the rule set is dependent on the token type. There are currently rule sets for English, numerals, punctuation, and spelling. Spelling is the English pronunciation of a single characterts name You must also consider that alphabetic characters do not always represent an English word.

Frequency tables representing the occurrence of letter pairs (digrams) or triplets (trigrams) offer significant help in deciding whether a group of characters represents an English word, an acronym, or a mnemonic. The frequency tables currently in use were derived from a lexicon of about a quarter of a million words. The digram-frequency table is reduced to a binary table that represents the occurrence or nonoccurrence of letter pairs in the lexicon. The use of digram or trigram tables could be expanded to the detection of specific subsets of English vocabulary. One case where this is useful: frequency tables derived from a common-usage dictionary and a lexicon of medical terms are significantly different.

Rule-Directed Translation

Orthographic representations of text are translated to phonetic representations by means of a production system. The niles used in the English-tophoneme translation match context-sensitive patterns to the word or word token. The rules are of the form:

left-context [current-token]
right-context = phonemes

The current-token is the character(s) that is currently being translated by a rule. The left-context and the right-context are the text in which the current-token must be matched. These left- and right-contexts may contain special symbols that define arbitrary patterns of characters. The current-token may

not contain these special symbols and must match, character for character, the token of the word being translated. The right-hand part of a rule gives the phonetic symbols representing the current-token, English phoneme rules are classified in subgroups of alphabetic, numeric, punctuation, and spelling rules. The phonetic replacements selected by the successful matching of rules are used to drive a speech-synthesizing device.