DistributedAUDIO)))
Mobilized Speech-Audio Publishing

Why Can't You *Listen* To *Your* Information On Your Way To Work?

By Gary Tjaden, Ph.D.

Speech-Audio Publishing

Many of us are too busy to read all the things we would like to, but we do have time when we could *listen* to them. This fact has not gone unnoticed, of course. Books have been available in audio tape-cassette form for many years. More recently, newspapers, magazines and books are being produced as compressed digitized speech (MP3) files, which can be downloaded over the Internet, transferred to PDAs and MP3 players, and then listened to wherever and whenever desired. This form of publishing by downloading the speech information to client devices where it is spoken has come to be called "speech-audio" publishing. If the client devices are mobile (e.g. PDAs rather than desktop PCs), I call this "mobilized speech-audio publishing".

There is a much more cost-effective approach to speech-audio publishing than the current digitized speech approach. That is to use a text-to-speech (TTS) engine embedded in a PDA or cell phone to speak general information downloaded as small text files. Why is this approach not being used? The primary reason is that no TTS engine can speak generalized information with complete accuracy, especially if the information contains business or technical jargon, acronyms, or the names of individuals. (Try your TTS engine on the name of the Prime Minister of Israel, Ariel Sharon, for example.) It is just not feasible for the local, client-side TTS dictionary to contain all possible word pronunciations, nor for the speech analysis algorithms to always correctly interpret homonyms, hyphenations, numbers or abbreviations.

There is a simple solution to this problem, however. It is to edit the text for speaking *before* it is downloaded to the PDA or cell phone. We have found that, for example, an hour-and-a-half of spoken information from the Wall Street Journal can be edited for correct speaking by clerical personnel in 15-20 minutes. The resulting file is about 35 KB in size, and can be downloaded over a dial-up Internet connection in a few seconds.

This article tells you how to determine if TTS-based mobilized speech-audio publishing is right for you.

TTS Pronunciation Accuracy

To illustrate the TTS accuracy problem I captured the text of a sports article from the Internet, had several off-the-shelf TTS engines read it on a PDA, and counted the number of words pronounced incorrectly. The results are summarized in Table 1.

Table 1: TTS Pronunciation Accuracy

Supplier	Engine	Num. Errors	% Errors
Acapela (EU)	Elan	67	10.50%
Cepstral	Swift	71	11.13%
Fonix	DECtalk	71	11.13%
Loquendo (Italy)	Loquendo TTS	36	5.64%
Scansoft/Speechworks	RealSpeak Solo	57	8.93%
Voiceware/NeoSpeech	VoiceText	60	9.40%

The test article, entitled "Serena Struggles Past Teen in Nasdaq-100" was chosen randomly and captured from Yahoo News on 3/28/05. This article contains several unusual names, some repeated, as well as repeated tennis match scores, and repeated abbreviations. Thus, for each engine, a number of the errors counted were repeats of the same error. There are 638 words in the article. All tests were performed on a Pocket PC (Axim A30 or iPaq H3800).

Table 1 shows that the lowest pronunciation error rate achieved was 5.64%, and the highest was 11.3%. These error rates are clearly too large for any type of information except the most trivial. Corporate, business, technical or professional information upon which workers or customers depend must, of course, have very high accuracy. More general information, such as news and analysis, tends to be advertising or subscription supported. Publishers of such information who expect to derive revenues from either of these sources will, most properly, demand high accuracy.

Table 1 reveals also that speech accuracy is not a function of the underlying synthesis technology. Two types of technology are currently in use, called formant and concatenated. Concatenated speech synthesis uses snippets of actual human speech that are "concatenated" to form words. It tends to sound more natural than formant speech. All of the test speech engines use concatenated speech, except for DECtalk, which uses formant speech. Its error rate is in the mid-range between the highest and lowest rates.

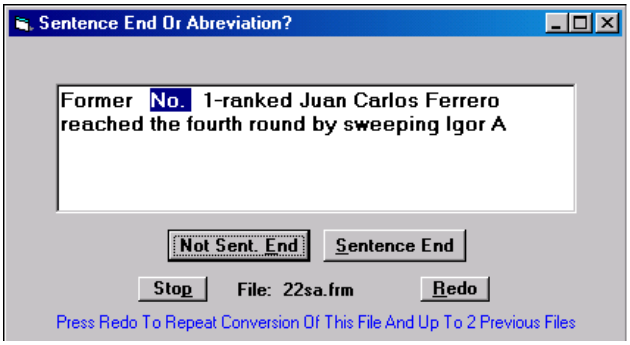

Figure 1: Parsing TTS Text Into Sentences

Some other relevant issues regarding these technologies, and their implications for speech-audio publishing are discussed later in the article. First, however, a technique for easily reducing pronunciation error rates to virtually zero for any speech engine is presented.

Editing Text For TTS Speaking

Almost all speech engines allow for adjusting the pronunciation of individual words, and then placing these new pronunciations into a custom dictionary from which they are retrieved as the engine speaks the text it is given. Sometimes this adjustment can be accomplished be merely respelling the word, or inserting a hyphen. For example, the word "Nasdaq", which appears in the test article, might need to be respelled "Naz-dack".

If this rather simple adjustment fails, the pronunciation must be defined in a "phonemic" form through the use of a special alphabet that allows each basic sound (phoneme) in the word to be explicitly specified. For example, tennis player Kim Clijsters is mentioned in the test article. In order for her last name to be pronounced correctly by the DECtalk speech engine it must be recast into its phonemic form, "[k ll'ays t rrz]". While "standard" phonemic alphabets have been defined, not all speech engines currently support the standards. Virtually all speech engines, however, have at least a proprietary alphabet.

The "custom dictionary" approach to pronunciation accuracy is sufficient for fixed vocabulary uses, such as interactive voice response systems. It is not sufficient for mobilized speech-audio publishing, however, because each client device (PDA) would have to have its own copy of the dictionary, and each copy would have to somehow be updated for each new item of information downloaded. Even if such client-side custom dictionaries were feasible, incorrect pronunciations due to improper interpretation of abbreviations, hyphenations, numbers and homonyms still would not be corrected.

As mentioned above, a solution to the problem of pronunciation accuracy for mobilized speech-audio publishing exists. It is to edit the text for speaking *before* it is downloaded to the mobile appliance*. The downloaded information is still just text, but it is text that will be spoken accurately.

A computerized editing tool can perform this editing mostly automatically. Whenever a situation is encountered that can't be handled automatically, the editing tool stops and asks the operator to select the correct option. This process is very similar to spell-checking a word processor document. There are two steps:

1. Parse the text into individual sentences, asking the operator to decide if an abbreviation is a sentence-end when necessary.
2. Check each word for correct pronunciation against a custom editing dictionary. If a word is not in the dictionary, ask the operator to approve or create its pronunciation. Also, check hyphenations and numbers.

* This concept and relevant methods are protected by patents assigned to COCOMO ID, LLC

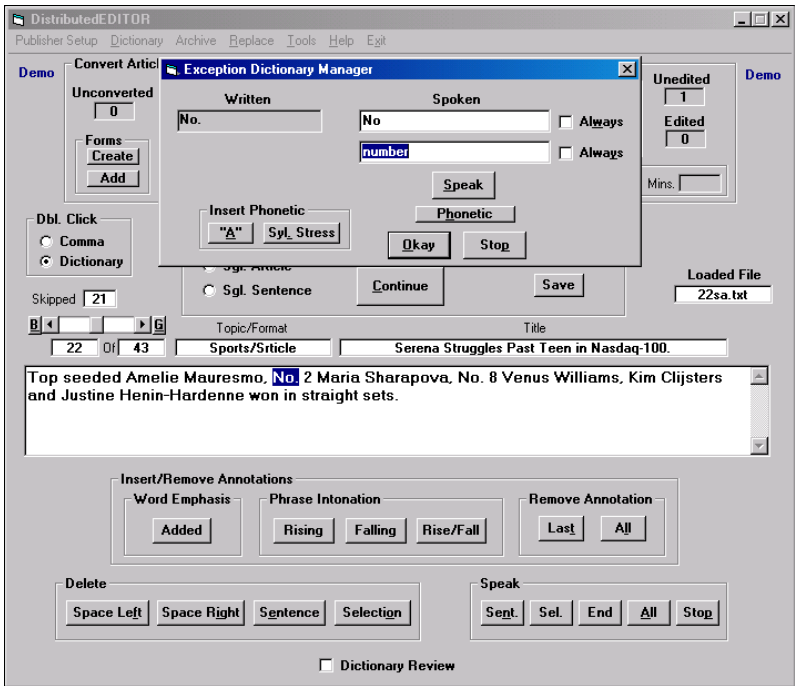

Figure 2: Editing TTS Text For Homonyms

A desktop software TTS Editing tool, called DistributedEDITOR, has been developed and integrated with several speech engines. The editing process will be illustrated here using the DECtalk version, which is the most current, as it edits the test article.

Figure 1 shows a typical dialogue window displayed during the Step 1 process of parsing the test article into sentences. Here the operator is being asked to decide whether the abbreviation "No." is the end of a sentence, or is just an abbreviation embedded within a sentence. Since abbreviations end with periods, and so do sentences, the occurrence of an abbreviation can be ambiguous. The correct answer here is that it is not a sentence end. Two of the test speech engines (Elan and DECtalk) handled this particular abbreviation incorrectly.

The second editing step of checking the pronunciation of each word requires several kinds of tests. One is for words that are spoken differently depending on context, called homonyms. While some speech engines do a good job of correctly pronouncing "regular" homonyms, such as "record" or "close", there are many words, such as "No.", that are not normally considered homonyms but actually are. If "No." appears at the end of a sentence it should probably be pronounced "no", otherwise probably "number".

The DistributedEDITOR keeps a list of all homonyms encountered for the first time as editing is performed and the operator creates alternative pronunciations. When one of these words, called "Exception Words" is encountered during editing, a dialogue, such as shown in Figure 2, is displayed giving the alternative pronunciations so the operator can choose the correct one.

Another kind of Step 2 test is for hyphenated or slashed (e.g., "and/or") words. These types of words also need to be pronounced differently depending on context, but there can be more than two alternatives.

For example, in the test article the tennis match scores are given as "6-4, 6-2". These scores should be pronounced "6 4, 6 2". That is, the games won in each set should be pronounced as pairs of individual numbers with a pause between pairs.

However, this pronunciation will not be correct for a player's win-loss record, as illustrated in Figure 3. Here the hyphen should be replaced with the word "and", so it is pronounced "6 and 7". In other situations, such as when reporting a baseball score of "5-4", the hyphen should be replaced with the word "to" so the score is pronounced "5 to 4".

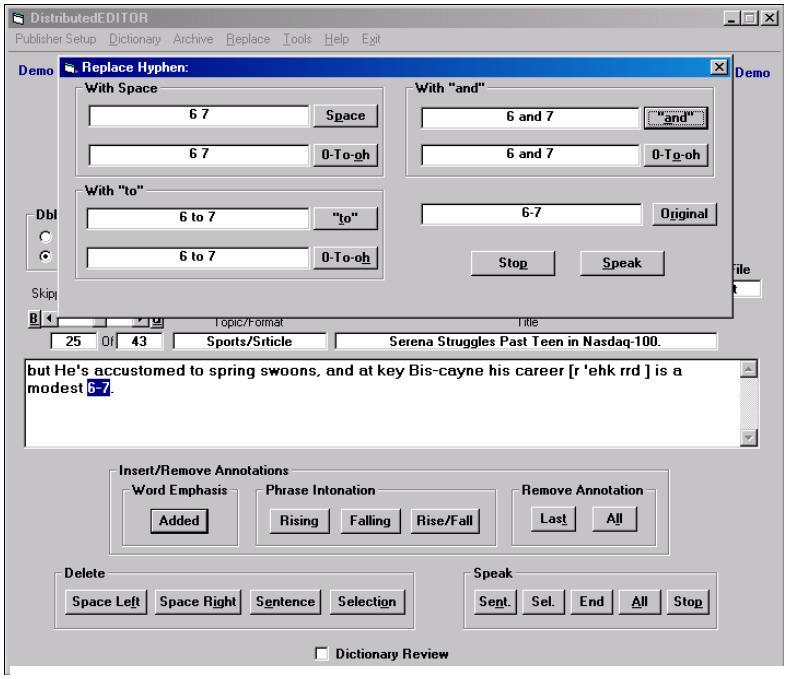

Figure 3: Editing Hyphenated Words For TTS Speaking

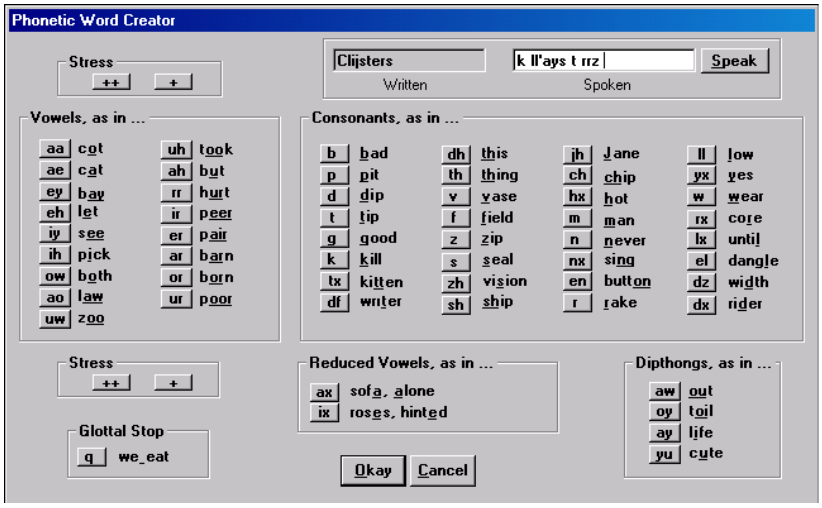

Figure 4: Creating Phonemic Pronunciations

None of the test speech engines handled either the match scores or the win-loss record pronunciations correctly. As shown in Figure 3 the DistributedEDITOR detects these situations and provides the operator with a quick and easy way to select the correct pronunciation.

The editing tool also checks, in a similar fashion, the pronunciation of numbers. For example, the operator is asked to decide if the number "747" should be pronounced "7 47", as it would when referring to Boeing's airplane model 747.

Finally, the editing tool makes it easy to create new word pronunciations for adding to or updating in the editing dictionary. Figure 4 shows the window displayed to the operator when it is necessary to create a phonemic version of the word, as described above for the last name of tennis player Kim Clijsters. All possible phonemes are shown with an illustration of their associated sound, and the operator need merely press the button for the required phoneme to add it to the word. Only two of the test speech engines, Loquendo TTS and VoiceText pronounced the word "Clijsters" correctly.

To illustrate the power of DistributedEDITOR I measured the amount of time required to edit the test article with the DECtalk version. A special demonstration dictionary, having only about 50,000 words, was used. Thus, many "common" words that most speech engines will pronounce correctly are already in the dictionary. Only less common words will be brought to the operator's attention. For these tests DistributedEDITOR was running on a laptop computer with a 1.66 GHZ CPU and 480 MB if RAM. The results are summarized in Table 2.

There are twenty-seven words in the test article that are not in the demonstration dictionary. Of these, nineteen were pronounced correctly by the DECtalk speech engine without change, seven required slight respelling (e.g., insertion of a hyphen or extra letter, and one, "Clijsters" required creation of a phonemic version.

The total editing time of the article the first time through was 3 minutes 28 seconds. Virtually all of this time was operator action time. The operator never had to wait for the DistributedEDITOR to perform its functions. The total speaking time for the article is about 4 1/2 minutes. Thus, even with a relatively limited dictionary size, the editing time is less than would be required for a human to speak the article to record it for digitizing.

However, once a word is in the dictionary it need never again require operator time for editing, unless it is a homonym. Thus, the total editing time for subsequent editing of this article reduces to only 1 minute and 41 seconds. This is the time required for the operator to parse the sentence for abbreviations (13 seconds), and deal with homonyms, numbers and hyphenated or slashed words (1 minute 28 seconds).

It is clear, then, that there are significant cost benefits for TTS-based mobilized speech-audio publishing relative to the digitized speech approach. Production of content for publishing will take less time and require less skilled personnel, delivery time will be much shorter, and content storage size much smaller.

Selecting A Speech Engine

By now it should be clear that selection of the right TTS speech engine for use in a specific mobilized speech-audio publishing application need not and should not require considering the accuracy of pronunciation. All speech en-

Table 2: Measured Editing Time

Editing Step	Elapsed Time (mins:secs)	
	First Edit	Next Edits
Sentence Parsing	0:13	0:13
Word Pronunciation	3:15	1:28
TOTAL	3:28	1:41

gines are significantly inaccurate, but pre-editing for TTS speaking makes this fact irrelevant.

I believe the selection of a speech engine requires making a trade-off between three factors: pleasantness and clarity of the speech, and memory footprint on the target mobile appliances. Of these, only memory footprint can be measured objectively.

Table 3 shows the memory footprints of the test speech engines on a Pocket PC. The memory footprint of the only formant speech technology engine, DECtalk, is significantly smaller than for the other concatenated technology engines. This footprint difference is true in general, and is a basic property of the different technologies.

As mentioned above, concatenated speech is generally perceived to be more "natural sounding", or "pleasant". I would agree. However, in my tests I observed that the concatenated speech was not necessarily easy to understand. That is, it was lacking "clarity". However, I observed that the formant speech could be more easily understood than several of the concatenated engine's speech. These observations are, of course, subjective.

Notice from Table 3 that the Total Running Footprints of some of the concatenated speech engines are very large (>30MB). And these sizes are for just one voice (male, female, young, mature, etc.). Using additional voices in an application could easily be infeasible if the target client platform has limited memory, such as cellphones, or even smartphones without a memory card. Formant speech, on the other hand, doesn't require extra memory for additional voices.

In some cases there is a fourth consideration in selecting a speech engine. If the publishing is to be advertising supported, for example, great care will need to be taken to make the advertising messages as dramatic and emotionally compelling as possible. Many speech engines support the insertion of special word or phrase emphasis annotations into the text. Some offer more emphasis possibilities than others, and the realism of the emphasis effect varies from engine to engine also. The bottom of Figure 3 shows DistributedEDITOR controls for supporting word and phrase emphasis for DECtalk. DECtalk's capabilities in this area are somewhat limited relative to some other speech engines.

Conclusion

If you are a publisher, wouldn't you like to give your "readers" the option to *listen* to your content wherever and whenever they choose? It might be the only way they could find the time to receive it at all.

If you are a provider of TTS technology, shouldn't you be trying to expand into markets requiring the mobilized publishing of generalized information? Or, if you are responsible for mobilizing the information systems of an enterprise, shouldn't you be offering your clients mobilized speech-audio publishing capabilities?

Perhaps you are an individual who doesn't have time to read all the information you would like. Then, why not contact the publisher of that information and ask them to publish with mobilized speech-audio?

There is really nothing to keep them from saying yes!

Table 3: TTS Memory Footprint (MB)

Supplier	Engine	Storage (common)	Storage (per voice)	Program	Total Running Footprint*
Acapela (EU)	Elan	5.47	17.51	7.82	30.80
Cepstral	Swift	0.00	10.97	5.18	16.15
Fonix	DECtalk	0.91	0.00	4.10	5.01
Loquendo (Italy)	Loquendo TTS	2.97	6.38	7.24	16.59
Scansoft/Speechworks	RealSpeak Solo	2.09	5.38	3.83	11.30
Voiceware/NeoSpeech	VoiceText	3.88	25.90	4.36	34.14

*For one voice, unless size is voice independent