# BBN Real-Time Speech Recognition Demonstrations

*Steve Austin, Rusty Bobrow, Dan Ellard, Robert Ingria, John Makhoul, Long Nguyen, Pat Peterson, Paul Placeway, Richard Schwartz*

BBN Systems and Technologies
Cambridge MA 02138

Typically, real-time speech recognition – if achieved at all – is accomplished either by greatly simplifying the processing to be done, or by the use of special-purpose hardware. Each of these approaches has obvious problems. The former results in a substantial loss in accuracy, while the latter often results in obsolete hardware being developed at great expense and delay.

Starting in 1990 [1] [2] we have taken a different approach based on modifying the algorithms to provide increased speed without loss in accuracy. Our goal has been to use commercially available off-the-shelf (COTS) hardware to perform speech recognition. Initially, this meant using workstations with powerful but standard signal processing boards acting as accelerators. However, even these signal processing boards have two significant disadvantages:

1. They often cost as much as the workstation they are plugged into.
2. The interface between each board and workstation is complicated, and always different for each combination of workstation and board.

To make speech recognition available to a broad base of users at an affordable cost, we have eliminated these disadvantages by developing algorithms that are able to operate in real-time on COTS workstations without requiring additional add-on hardware and without decreasing recognition speed and accuracy. An additional advantage is that we are able to benefit from the improvements in workstation price and performance, with very minimal porting effort. The BBN RUBY$^{TM}$ system, a robust commercialization of the BYBLOS$^{TM}$ speech recognition technology, is the result of this development effort. At the workshop, we demonstrated two example systems that employ the RUBY speech recognition system.

Both demonstrations run on Silicon Graphics workstations (Personal IRIS 4D/35 and Indigo), which contain a built-in programmable A/D-D/A. The signal processing and vector quantization, which runs in a separate process from the recognition search, communicates with the recognition search via network sockets. We have reduced the computation required for this front end processing to the point where it requires little enough of the CPU so that there is enough left over to perform the more expensive search in real time. Since accuracy is our primary concern, we have verified that this signal processing results in the same accuracy as our previous signal processing software.

## 1. REAL-TIME ATIS SYSTEM

The ATIS demonstration integrated BBN's DELPHI natural language understanding system with the RUBY speech recognition component. RUBY is used as a black-box, controlled entirely through an application programmers interface (API). The natural language component is our current research system, which runs as a separate process. Both processes run on the same processor, although not at the same time. The NL processing is performed strictly after the speech recognition, since competing for the same processor could not make it faster. (If two separate processors are available, the processing can be overlapped as described in [1].)

The speech recognition component performs three separate steps. First, it uses a forward Viterbi-like computation to find the 1-Best speech answer in real time as the user is speaking. Very shortly after the user stops speaking, it displays the 1-best answer. Then, it performs a backwards N-Best pass to find the N-Best hypotheses. Finally, we rescore each of the text hypotheses using a higher-order n-gram statistical class grammar and reorder the hypotheses accordingly. In this application, we use a trigram model; this rescoring computation requires very minimal processing. (Note that at this time we have omitted the acoustic rescoring stage in which we could rescore with between-word triphone models and semi-continuous HMM models.) After that, the N-Best answers are sent to DELPHI which searches the N-Best answers for an interpretable sentence and then finds the answer to the question.

Our goal in this effort was to produce a system that was as close to real-time as possible. However, it is of little interest to us to demonstrate real-time at the expense of accuracy. We have verified that the speech system, when operating in real-time, degrades only marginally from the very high performance figures reported in the evaluation. In particular, the word recognition error rate degrades from 9.6% to 11.7% – a 20% increase in error rate.

Normally, the system displays the 1-Best answer within a half second of detection of the end of speech. This is fast enough that it feels instantaeous. (This speed is in marked contrast with the other near real-time demonstrations shown at the workshop, which all required from two to five times real time, resulting in a delay that was at least equal to the length of the utterance – usually several seconds.) Next it performs the N-Best recognition, and then interprets the answer. The N-Best recognition usually runs in less than 1 second, since it is sped up immensely by the forward pass. The time required for the interpretation depends on how many of the speech answers must be considered, and on how complicated a retrieval results. In most cases, this phase requires only another second or two.

To operate the demonstration system, the user clicks on the "Push to Talk" window at the top of the screen. The status will change from "ready" to "listening". As soon as the user begins speaking, the status will change to "beginning of speech". When (s)he stops speaking, it will change to "end of speech". The system briefly displays its status as "First-Best" and "N-Best" while it completes these phases of the recognition. Finally, the system will "Interpret" the query, which includes all parsing, semantic interpretation, discourse modeling, and data retrieval from the actual database.

The answer displayed in the speech window first contains the answer from the 1-Best pass, then the top-choice of the N-Best, and finally the sentence chosen by DELPHI. The N-Best hypotheses are displayed at the bottom of the screen for information only. Then, the answer to the query is displayed under the recognized sentence. If the answer to be displayed is larger than will fit in the window, it can be scrolled. A history of the previous four sentences are shown in a window that can scroll all the way back through the previous questions. If the user wishes to review any of the previous answers in more detail, they may mouse on the arrow to the right of the question, which brings back a copy of the question and answer in a separate window that may be placed and sized as desired, and then used for reference as long as needed.

To the right of the main display, we also display the *discourse state*, which consists of the set of constraints that were used to answer the query. In this way, the user can verify how much of the previous context was actually used to answer the question. As each successive query is interpreted, the system may add new constraints, modify old ones, or completely reset the context. The user may also reset the discourse state by speaking either of the commands, "BEGIN SCENARIO", "END SCENARIO", or "NEW SCENARIO".

## 2. REAL-TIME SPEECH RECOGNITION FOR AIR-TRAFFIC CONTROL APPLICATIONS

We also demonstrated RUBY configured for air-traffic control (ATC) applications. This system is notable for its very high speed, accuracy, robustness, and reliability, all necessary qualities for ATC applications requiring human-machine interaction. Such applications include training systems, where the trainee controller interacts with a simulated world, and operational environments, where a controller's interaction with a pilot could automatically generate a database retrieval request for flight plan information.

In this demonstration, the system extracts the aircraft flight identification from an utterance as soon as that information has been spoken. For example, if the controller says, "Delta three fifty seven descend and maintain 2000", the flight information could be captured for display or other uses by the time the controller has said the first syllable of "descend". To achieve this immediate response, the speech recognition detects when the controller has completed the flight identifier and is speaking the rest of the utterance. This requires a different process than is usually used for speech recognition. Normally we wait until the end of the sentence to determine the most likely word string for the complete utterance. For this application, the system stops the recognition process as soon as it determines that it is most likely to have the complete flight information.

Another unique capability that is demonstrated here is the capability to reject the flight ID if it is not in a specific closed set. Again, this is done by explicitly modeling the likelihood that the user has spoken a flight ID other than the set that is expected at any given time. Further development of these systems is continuing.

## REFERENCES

1. Austin, S., P. Peterson, P. Placeway, R. Schwartz, J. Vandergrift, "Toward a Real-Time Spoken Language System Using Commercial Hardware", *Proceedings of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann Publishers, Inc., Jun. 1990, pp. 72–77.
2. Schwartz, R., S. Austin, "Efficient, High-Performance Algorithms for N-Best Search", *Proceedings of the DARPA Speech and Natural Language Workshop*, Morgan Kaufmann Publishers, Inc., Jun. 1990, pp. 6–11.