



# A DISTRIBUTED ARCHITECTURE FOR ROBUST AUTOMATIC SPEECH RECOGNITION

*Kadri Hacioglu and Bryan Pellom*

Center for Spoken Language Research  
University of Colorado at Boulder  
[{hacioglu,pellom}@cslr.colorado.edu](mailto:{hacioglu,pellom}@cslr.colorado.edu)

## ABSTRACT

In this paper, we attempt to decompose a state-of-the-art speech recognition system into its components and define an infrastructure that allows a flexible, efficient and effective interaction among the components. Motivated by the success of DARPA Communicator program, we select the open source Galaxy architecture as our development test bed. It consists of a hub that allows communication among servers connected to it by message passing and supports the plug-and-play paradigm. In addition to message passing it supports high bandwidth data (binary or audio) transfer between servers via a brokering scheme. For several reasons, we believe that it is the right time to start developing a distributed framework for speech recognition along with data and protocol standards supporting interoperability. We present our work towards that goal using the Colorado University (CU) Sonic recognizer. We divide Sonic into a number of components and structure it around the Hub. We describe the system in some detail and report on its present status with some possibilities for future development.

## 1. INTRODUCTION

Speech recognition software grows more complex as speech scientists introduce new ideas, develop new technologies and incorporate them into their systems. For challenging tasks, like speech recognition in noisy environments with a large pool of speakers, it is not uncommon to have multiple speech recognition systems with multiple passes and techniques incorporated for environmental and voice activity detection, speech enhancement, vocal tract and variance normalization, speaker adaptation, confidence calculation and hypothesis combination. Development and maintenance of these systems are very difficult. Naturally, such a complexity raises the entry barrier for new players, and the research area becomes dominated by groups that have their own end-to-end systems. Although, some research sites provide their systems for free for research purposes, still it is difficult to get familiar with and incorporate new ideas into those systems. In addition, people are restricted to the technology available in those systems. However, people might want to share some better component technologies developed by others. Probably, lack of compatibility will hinder those efforts. A solution is to break apart the “monolithic” structure of speech recognition software into a number of standard components and define a distributed processing architecture with properly defined data and protocol standards for component communication.

Galaxy architecture has been developed, and optimized for spoken dialog systems (SDS) [1]. Initially MIT, and then MITRE have spent significant amount of effort in developing and making the system available as an open source. It has been successfully used by the sites that participated in the DARPA Communicator program [2]. MITRE has also demonstrated its plug-and-play ability by intermixing the components that has been developed at different sites [3]. The architecture has a programmable Hub that allows a flexible control of interaction among servers, and a set of libraries for rapid prototyping including a graphical user interface for controlling and monitoring the processes. We believe that the following major functions of the Hub, which have been proved to be very useful for developing SDS, are also very useful for decomposing a speech recognizer into its components [4]:

- Routing: Handles message traffic among the distributed servers
- State Maintenance: Provides a means of storing and accessing state information for all servers
- Flow control: Manages the progress of an utterance through its processing stages, server by server

Modularization of speech recognition software around such a central process has several advantages:

- Lowered entry barrier for new players
- Rapid portability
- Flexible accommodation of new components
- Easy access to information needed for debugging
- Resource sharing
- Creation of market for components
- Promoting standardization
- A nice educational tool for learning and teaching speech recognition
- Easy extension to other input-output modalities (e.g. lip reading, visual presentation)
- Fair setup for the evaluation of component technologies

The paper is organized as follows. Section 2 describes the CU Sonic recognizer as configured for the DARPA SPINE task. In Section 3, we propose and describe a modular form of Sonic within Galaxy architecture. We summarize what has been implemented so far in Section 4. Our future plan regarding the new architecture is presented in Section 5. Concluding remarks are made in the final section.

## 2. CU SONIC SPINE SYSTEM

The University of Colorado has previously participated in both SPINE-I [5] and SPINE-II evaluations. Our efforts towards the evaluation systems have focused on (1) the development of new features for robust speech recognition, (2) improved model adaptation methods and (3) an efficient, integrated approach to joint speech detection and recognition for noisy environments.

Our most recent fielded evaluation system in 2001 (SPINE-II) was based on Sonic: CSLR's large vocabulary continuous speech recognition system [6]. Sonic is based on continuous density hidden Markov (CDHMM) acoustic models. Context dependent acoustic models are clustered using decision trees. Each model has three emitting states with gamma probability density functions for duration modeling. Features are extracted as 12 MFCCs, energy, and the first and second differences of these parameters, resulting in a feature vector of dimension 39. The search network is a reentrant static tree-lexicon. The recognizer implements a two-pass search strategy. The first pass consists of a time-synchronous, beam-pruned Viterbi token-passing search. Cross-word acoustic models and 4-gram language models (in an approximate way) are applied in the first pass of search. The first pass creates a lattice of word ends. During the second pass, the resulting word-lattice is converted into a word-graph. Advanced language models (e.g. dialog-act and concept based, long span) can be used to rescore the word graph using an A\* algorithm or to compute word-posterior probabilities to provide word-level confidence scores.

Sonic provides an integrated environment that incorporates voice activity detection (VAD); speech enhancement, speaker adaptation and normalization methods such as minimum mean squared error (MMSE) speech enhancement [7], confidence weighted Maximum Likelihood Linear Regression (MLLR) [8], and Vocal Tract Length Normalization (VTLN) [9].

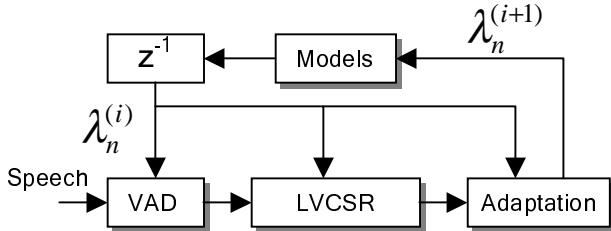


Figure 1. Block diagram of Sonic SPINE system.

Our SPINE system consists of integrated speech detection and multiple pass recognition search as shown in Figure 1. At each iteration, a voice activity detector (VAD) is dynamically constructed from the current adapted system acoustic models. The VAD generates a segmentation of the noisy audio into utterance units and LVCSR is performed on each detected speech region. The resulting output (a confidence tagged lattice or word string) is then used to adapt the acoustic model means and variances in an unsupervised fashion. The adapted acoustic models are then reapplied to obtain an improved segmentation, recognition hypothesis, and new set of adapted system parameters. The integrated adaptation procedure is repeated twice resulting in sequential improvements to both segmentation and recognition hypotheses.

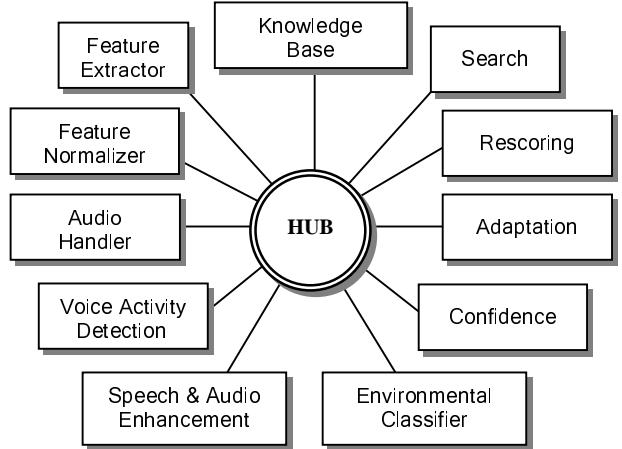


Figure 2. A distributed architecture for a speech recognition system that incorporates a Hub and eleven servers.

## 3. A DISTRIBUTED ARCHITECTURE

In this section we describe an appropriate decomposition of Sonic SPINE system. The resulting configuration is exhibited in Figure 2. We identified a total of eleven servers:

- Audio Server
- Voice Activity Detection (VAD) server
- Environmental Classification Server
- Speech Enhancement Server
- Feature Extraction Server
- Normalization Server
- Search Server
- Knowledge Base (KB) Server
- Rescoring Server
- Adaptation Server
- Confidence Server

We do not claim that this is the best decomposition of a speech recognition system. It should be noted that it is our initial configuration and intended for speech recognition in noisy environments.

### 3.1. Audio Server

We plan to have three versions of the audio server; namely, live, wireless, and batch. The live audio server will provide microphone and telephony interfaces, while the wireless audio server is intended for a system in which the features are directly transmitted to the recognizer from a handheld device. The batch audio server reads speech from files. For the sake of simplicity we have shown a single audio server in Figure 2. .

Input: Audio signals or speech files or features transmitted through a wireless channel.

Output: Audio signal will be made available through a brokered data connection to the voice activity detection, feature extraction and speech enhancement servers. Decoded features will be made available through a shared memory to the servers that need those features.



### 3.2. Voice Activity Detector (VAD) Server

This server detects speech vs. non-speech regions in the audio.

Input: A continuous channel of audio is streamed via a direct-brokered data connection from the audio server.

Output: The server outputs begin and end times of speech activity to the Hub for use by the environmental classifier, speech enhancement server, and search server.

### 3.3. Environmental Classification Server

This server is responsible of detecting, classifying and tracking environmental changes. It passes this information to the servers that need to adapt their behavior according to the type of environment. Examples to these are the speech enhancement and KB servers. The speech enhancement server might apply different enhancement techniques depending on background noise-shape and type, and the KB server might provide environment dependent or adapted acoustic models to cope with environmental changes.

Input: The input consists of features extracted from the feature extraction server and begin/end times of speech/non-speech regions from the VAD server.

Output: A symbolic representation or classification of the noise type with time-markings. For example, in in-vehicle systems, this might include car conditions (windows up, windows down, car passing noise, radio state, etc.).

### 3.4. Speech Enhancement Server

This server is essential for the recognition of speech in noisy environments. This server is responsible for tracking and attenuating background noise present in the audio channel.

Input: The input to this server is the audio data channel and the voice activity detection timing information from the VAD server. The non-speech regions are used to update spectral estimates of the noise.

Output: A noise attenuated audio channel is output for use by the feature extraction module and search server.

### 3.5. Feature Extraction Server

This server extracts features from the audio channel and makes them available to the search, adaptation, normalization, rescoring and confidence servers, if needed, through shared memory. We envision that this server will extract spectrally motivated feature types (e.g., MFCC, PLP, Root Cepstrum) as well as prosodic features (e.g., F0, degree of voicing, etc.).

Input: An audio channel from the speech enhancement or directly from the audio server.

Output: A stream of feature vectors that can be accessed by various modules through a shared memory module.

### 3.6. Feature Normalization Server

The feature normalization server compensates for channel and speaker conditions by applying various transformations on the speech features. Examples types of normalization include cepstral mean subtraction (CMS), vocal tract length normalization (VTLN) and cepstral variance normalization. This server will also be responsible for accumulating statistics

needed for histogram equalization of the filter bank energies in MFCC calculation.

Input: Unnormalized features from feature extraction module (potentially through a shared memory access)

Output: Channel and speaker normalized features

### 3.7. Search Server

This server builds and searches the recognition network. During search it generates a word lattice from which a word graph can be created.

Input: time-synchronous stream of feature vectors

Output: word-lattice representation of the search space

### 3.8. Knowledge Base (KB) Server

This server provides access to knowledge sources needed by the system. These include pronunciation lexicons, grammars, language models and acoustic models. The server interacts with the search server by providing observation probabilities from system Gaussians as well as language model probabilities. The server is also responsible for dynamic switching of the task-based language model, acoustic model and lexicon on demand. It is also responsible for applying transformations of system parameters as dictated by the adaptation server.

Input: Environmental classification labels, transformation matrices, knowledge base requests.

Output: Knowledge base information as lexical items, grammar rules, acoustic and language model indices/probabilities.

### 3.9. Rescoring Server

This server is responsible of performing the second pass of search. It accepts a compact representation of the search space from the search server in the form of a word lattice. The lattice is converted into a word graph and further refined through rescoring using higher-order knowledge sources.

Input: word lattice from search server (could be accessed through shared memory).

Output: An N-best list of word strings

### 3.10. Adaptation Server

Given the best string, or the word graph (possibly augmented with confidence and model alignments), this server determines sets of transformations to apply to the acoustic models to minimize mismatch between training and testing conditions.

Input: N-best list or word-graph and access to extracted features through shared memory.

Output: A set of transformations that can be applied to the acoustic models (e.g., class-conditioned MLLR matrices).

### 3.11. Confidence Server

This server is responsible of generating confidence values at different levels; namely, HMM-state, phone, word, concept and sentence levels. For their portability and descent performance we are in favor of confidence estimation methods based on the posterior probabilities of word graph edges that can be easily computed by a forward/backward like algorithm [11].



Input: Word graph, possibly model marked, with acoustic and language model scores from the rescorer server.

Output: Word graph annotated with posterior probabilities.

#### 4. CURRENT IMPLEMENTATION

We have begun development of the system architecture outlined in Section 3. The implementation uses version 4.0 of the Galaxy Communicator Infrastructure (GCI) [10] to implement the Hub and server interaction. Currently the distributed ASR system has been built for single-pass recognition using file-based input (although it can easily be extended to live-mode interaction using the Galaxy Communicator Infrastructure microphone based audio server). The implementation consists of the Hub connected to the following servers:

- Audio Server
- Voice Activity Detector (VAD) server
- MMSE Speech Enhancement Server
- Feature Extraction & Normalization Server
- Search, Knowledge Base (KB), Adaptation Server

The implementation performs single pass recognition and online incremental speaker adaptation using the MLLR technique. Specifically, the audio server reads recorded utterances and streams the samples of audio via a brokered connection to both the voice activity detection (VAD) server and the speech enhancement server. The resulting speech begin/end times from the VAD are then passed as input to the speech enhancement server. The speech enhancement server performs MMSE based signal estimation with knowledge of the speech begin/end times for noise estimation. The enhanced (noise attenuated) audio stream is then passed to the feature extraction and normalization server via a direct brokered connection. The feature extraction server computes a stream of 39-dimensional MFCC features that are normalized through cepstral mean subtraction. The features are then streamed to the search server, which then performs a time-synchronous token passing beam Viterbi beam search through a static reentrant tree lexicon. The server sends the single best word string to the Hub for logging. It also performs incremental online MLLR adaptation after each utterance and utilizes the updated linear transform for decoding the next utterance.

Our current effort consists of refinement of the server modules into smaller processing blocks as suggested in Section 3. Based on the Sonic ASR engine, we feel that the next step is to separate the adaptation module from the search server. We will also place the acoustic and language models into a separate knowledge base server. It should be noted that converting state-of-the-art speech recognition systems that have been tightly integrated in the past does require significant architectural changes. However, we feel that the distributed and modular architecture has many benefits for future system development.

#### 5. CONCLUSION

We have described a speech recognition system that has been decomposed into its components and configured using the DARPA Galaxy Hub architecture that supports the plug-and-play paradigm to rapidly develop efficient and effective interfaces among the components. The complete implementation of the system that we have mentioned in Section 3 is still under development. However, we have shown the feasibility of the proposed system by implementing an initial system in which some servers have been collapsed into single servers. We expect a complete system for demonstration be available at the time of conference. Our goal is to invoke an interest in an architecture that would define a standard for speech recognizer component interoperability.

#### 6. REFERENCES

- [1] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, V. Zue, "Galaxy-II: A Reference Architecture for Conversational System Development," *Proc. ICSLP*, Sydney Australia, Vol. 3, pp. 931-934, 1998.
- [2] J. Abardeen et. al., "Darpa Communicator Program Tackles Conversational Interface Challenge," The Edge; MITRE Advanced Technology Newsletter, 1999.
- [3] S. Bayer, C. Doran, and B. George "Exploring Speech-Enabling Dialogue with the Galaxy Communicator Infrastructure," *Proc. HLT*, pp. 114-116, San Diego, California, March 2001.
- [4] A. Goldchen, D. Loehr, "The Role of the DARPA Communicator Architecture as a Human Computer Interface for Distributed Simulations," Simulation Interoperability Standards Organization (SISO) Spring Simulation Interoperability Workshop, Orlando-Florida, 1999.
- [5] J. H. L. Hansen, R. Sarikaya, U. Yapaner, B. Pellom, "Robust Speech Recognition in Noise: An Evaluation using the SPINE corpus", *Eurospeech 2001*, Sept. 2001.
- [6] B. Pellom, "Sonic: The University of Colorado Continuous Speech Recognizer", Technical Report TR-CSLR-2001-01, CSLR, University of Colorado, March 2001.
- [7] Y. Ephraim and D. Malah, "Speech Enhancement using a Minimum Mean Square Error Short Time Spectral Amplitude Estimator," *IEEE Transactions ASSP-32*, pp. 1109-1021, 1984.
- [8] C. J. Legetter, and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, Vol. 9, pp. 171-185, 1995.
- [9] L. F. Uebel, and P.C. Woodland, "An investigation into Vocal Tract Length Normalization", *Proceedings of Eurospeech-99*, Budapest, Hungary, 1999.
- [10] <http://communicator.sourceforge.net>.
- [11] K. Hacioglu, W. Ward, "A Concept Graph Based Confidence Measure," *Proc. ICASSP*, pp. I-225-I-228, Orlando-Florida, May 2002.