

# A Software Architecture Supporting In-Car Speech Interaction

Andrew L. Kun, W. Thomas Miller III, Albert Pelhe, Richard L. Lynch

Electrical and Computer Engineering Department  
University of New Hampshire  
Durham, NH 03824  
{andrew.kun, tom.miller, apelhe, richard.lynch}@unh.edu

## Abstract

*Project54 is a research and development effort that involves integrating electronic devices in police cruisers into a single system. Officers primarily interact with the Project54 system through its voice interface. The Project54 system software architecture is completely modular and is based on Microsoft's Component Object Model (COM). The software includes specialized modules that allow interfacing to commercial speech recognition (SR) and text-to-speech (TTS) engines. Modules can communicate by means of one-to-one text messaging. Recognition results from the SR engine are routed to the appropriate destination application using text messages. Similarly, speech output requests are routed from applications to the TTS engine in the form of text messages. The system is currently used in over 100 cruisers of the New Hampshire State Police and other police departments in New Hampshire.*

## 1. Introduction

In [1] Mark Weiser described a vision of the future of technology in which ubiquitous computers blend into our everyday lives. We interact with them constantly but we do not have to concentrate on the interaction itself. Instead we can focus on the goals we are trying to accomplish through the interaction. Our world, of course, is very different from the world of Weiser's vision – our computers are centers of attention and we often need extensive training to be able to interact with them. One place where this deficiency of current computing technology is on full display is the inside of a police cruiser. Technological advances have introduced many new electronic devices in police cruisers, for example radar equipment, video equipment, sirens, emergency lights, radios, etc. Each of these devices has its own user interface. Each interface in turn acts as a center of attention and can be operated only by trained individuals.

Work at the University of New Hampshire Consolidated Advanced Technologies Laboratory (Catlab) concentrated on making the devices inside a police cruiser behave more like devices in Weiser's vision. In the Project54 system [2] the cruiser's devices are controlled by integrated software components running on an embedded computer. The integrated software components also implement an integrated user interface – the system allows the officer to have control over all the electronic devices either through a touch screen or through a voice interface. A key element of the Project54 system is its voice interface. While officers still need to be trained to use the devices, gone are the multitude of individual user interfaces that added complexity to the interaction. Officers can use voice commands to tell the system what they want to accomplish.

## 2. Background

The idea of integrating speech recognition into the car environment is not a new one. Researchers at Dragon Systems UK R&D published a paper in 1999 [3], discussing their experiences with in-car speech recognition. As the Dragon researchers and other researchers observe, cars present a unique challenge to speech recognition systems due to their frequently low SNR [4]. This can be combated through a variety of techniques. Project54 combats this problem by using a directional microphone. As Smolders et al [5] discovered, microphone placement also plays an important role in in-car speech recognition. Their results indicated the optimal position for the microphone is on the roof, in front of the driver. This position resulted in up to a 7dB SNR improvement over other positions in the car. In the Project54 system the directional microphone is installed on the roof by attaching it to the visor (see Figure 2). In addition to a low SNR, mobile speech recognition has to deal with the Lombard effect [6] – the tendency of people to speak differently in the presence of noise. Although this effect is beneficial for people attempting to understand the speaker, it can confuse speech recognition

engines since the speaker is deviating from his/her normal speaking style.

In [7] Kun et al describe Version 1 of the Project54 system software. Version 1 of the software implemented a speech user interface. It was a modular system based on Microsoft’s Component Object Model (COM). However, it did not allow direct, one-to-one communication between applications. Messaging between applications was many-to-one, and it was done anonymously (no source and no destination indicated). Many-to-one messaging did not allow taking advantage of synergies between individual applications. Version 2 of the Project54 software [8] provides one-to-one communication between applications along with the one-to-many communication that was implemented in Version 1. This allows applications to cooperate and share information, thereby enhancing the system functionality.

### 3. The Project54 system – high level overview

The Project54 system integrates the hardware, and the software controlling the hardware, of electronic devices in police cruisers. The outline of the hardware devices of the Project54 system is shown in Figure 1.

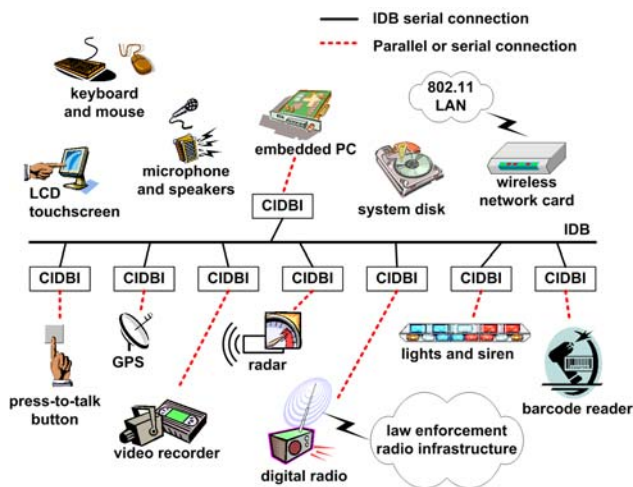


Figure 1 Hardware devices

At the center of the system is the embedded PC. The bottom part of the figure shows the devices that connect to the PC through the Intelligent Transportation Systems Data Bus (IDB): the lights and siren, the radar, the radio, the video recorder, the GPS unit, the barcode scanner, and the push-to-talk button (used to signal that speech recognition should be performed). The Common IDB Interface (CIDBI) is used to connect all of these devices

to the IDB [9]. The top part of Figure 1 shows devices that connect directly to the PC: the system disk, the keyboard and mouse, the microphone and speakers, the LCD touch screen, and the wireless network card. The 802.11 network card provides wireless connectivity to a local area network.

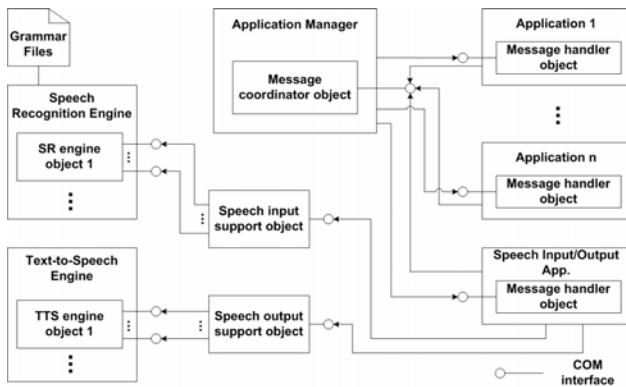
Figure 2 shows the Project54 system hardware installed in a New Hampshire State Police (NHSP) cruiser. The vehicle is equipped with an LCD touch screen. The touch screen can be used as a second input and output modality in case the speech interface is not available. There is also a keyboard, a directional microphone on the visor, as well as a push-to-talk button on the steering wheel. The keyboard can be used for tasks that are performed while the cruiser is parked. The directional microphone reduces the influence of sounds that are not coming from the driver of the vehicle – this improves SR engine performance.



Figure 2 The Project54 system in the cabin

### 4. Software support for speech interaction

The block diagram of the Project54 system software architecture [8] is shown in Figure 3. At the center of the system is the Application Manager. The Application Manager implements the message coordinator object which provides a means for receiving messages from individual applications. The applications implement a message handler object which provides a means for receiving messages from the Application Manager. The Application Manager handles messages sent between applications by receiving messages via the message coordinator object and sending messages via the message handler objects, as shown in Figure 3.



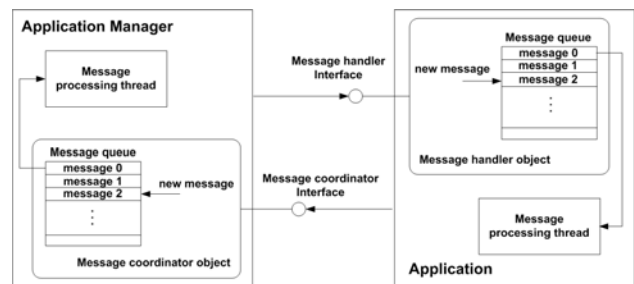
**Figure 3** Using SR and TTS engines in the Project54 system software

In the Project54 software system most of the applications control electronic devices. Most of these applications are speech enabled. These applications do not interact with speech recognition (SR) and text-to-speech (TTS) engines directly. Instead they utilize a special application, the Speech Input/Output Application (SIOA). The SIOA receives recognition results from the SR engine via the Speech input support object and forwards these results to the appropriate applications in the form of text messages. Likewise, applications send text messages to the SIOA to request speech output. The SIOA forwards these requests to the TTS engine for output via the Speech output support object. Each support object exports a set of interfaces. These interfaces are standardized for the Project54 system. Consequently, in order to use a new SR or TTS engine only the appropriate support COM object needs to change. The applications (including the SIOA), as well as the Application Manager, need not change.

The two-way messaging between applications plays a central role in supporting speech interaction in the Project54 system. A message queuing process is implemented in order to guarantee that all messages are going to be received. The message coordinator object puts incoming messages on the Application Manager's message queue. This is illustrated in Figure 4. The Application Manager also implements a message processing thread. This thread is called whenever there is a new message on the message queue. The message processing thread gets the message from the top of the queue and forwards the message to the destination application. The Application Manager may also be the destination of incoming messages – the message processing thread deals with these messages too. Since the message coordinator object only places incoming messages on a queue, its interaction with the application that wants to send the message is very short. The

potentially time-consuming message delivery or processing tasks are left to the message processing thread. This improves system responsiveness and guarantees that all messages sent to the Application Manager are received.

The Application Manager's message processing thread delivers messages to a destination application via the application's message handler interface. As illustrated in Figure 4, individual applications also implement a message queue and a message thread in order to handle received messages. Messages sent to an individual application are placed on the application's queue. When a new message is placed on the queue the application's message processing thread is called. This thread processes messages on the queue one at a time. Thus, message reception and message processing are decoupled and receiving a message takes very little time, even when processing the message may take significant time.



**Figure 4** Message queue and message processing thread

## 5. Speech interaction

An example of an officer's interaction with an application in the Project54 system using speech recognition and speech output is shown in Figure 5. In our example the officer wants to interact with the Records Application. This application provides access to databases such as a state's vehicle database or drivers' records database. If the officer wishes to query a database for records on a vehicle he/she needs to tell the application the license plate number of the vehicle. Let us assume that the police officer already told the application the license number of interest. Once this is done he/she needs to issue the "check records" command for the application to execute the query. To accomplish this, the officer presses the push-to-talk button, causing the Push-To-talk Application (PTTA) to send a message to the SIOA. The push-to-talk button is used to inform the system that it should perform speech recognition. The PTTA monitors the push-to-talk button and sends a message to the SIOA whenever it is pushed or released. All inter-application messages are

sent via the Application Manager. When the SIOA receives the message that the push-to-talk button is pressed it starts the SR engine. Once the button is pressed the officer says “check records” in order to request the data from the police database. When the push-to-talk button is released, the PTTA sends a message to the SIOA to stop the speech recognition. Following that, the recognized “check records” command is sent from the SIOA to the Records Application. In response to the “check records” command the Records Application retrieves the requested information from the state database and sends the information to the SIOA. The SIOA outputs the requested information using the text-to-speech engine.

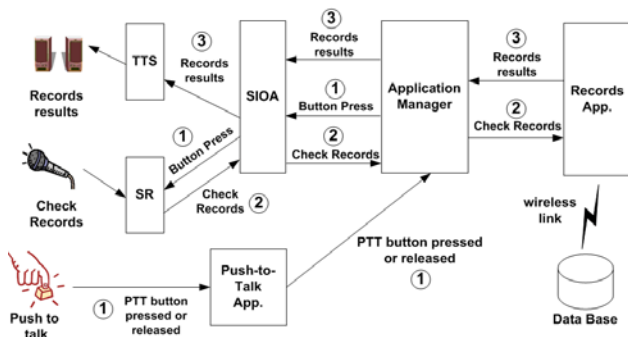


Figure 5 Project54 speech interaction

In applications that can process speech input the speech interaction is highly structured. Stages of interaction are predefined and have to be followed by the police officer. For example, in the Records Application the officer has to announce to the system what information will follow and then say the information. Thus, to provide the system with a vehicle’s license number the officer first has to say “license number,” a command which will be echoed by the system for confirmation. After the echo the officer can proceed to say the letters and numbers making up the license number.

The drawback of such structured interaction is that it requires attention to be focused on the mechanics of the interaction. However, the advantage is that we can define a separate grammar for each stage of the interaction. Grammars define valid phrases for the user to utter at a given time. Given a grammar the speech recognizer will try to match any speech input to what it knows to be one of the valid phrases. By making the number of valid phrases small we can improve speech recognition accuracy. Given the accuracy of today’s commercial speech recognizers this is of primary importance.

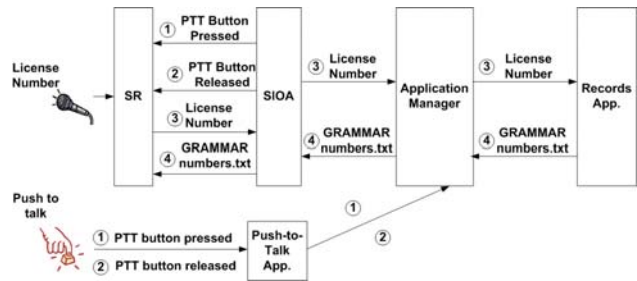


Figure 6 Grammar file change

An example of using multiple grammar files in the course of a structured interaction is shown in Figure 6. In this example the police officer is using the Records Application. The Records Application is waiting for the officer to initiate an interaction. When it entered this waiting state the Records Application instructed the SR engine to load a grammar that lists phrases that will allow the officer to start the interaction. One of these phrases is “license number” which tells the Records Application that the officer intends to give it a vehicle’s license number. Suppose that the officer does indeed want to tell the Record Application the license plate number of a vehicle. The officer presses the push-to-talk button and the Push-To-talk Application (PTTA) sends a message to the SIOA. The SIOA responds by making the SR engine listen for speech input. The officer says “license number” in order to tell the Records Application that he/she intends to give it a vehicle’s license number. When the push-to-talk button is released, the PTTA sends a command to the SIOA to stop the speech recognition. The recognized “license number” command is sent to the Records Application. In response to this command the Records Application sends the “GRAMMAR numbers.txt” command to the SIOA. The SIOA instructs the SR engine to load the grammar found in the “numbers.txt” file (using the correct path). This grammar tells the SR engine that phrases containing alphanumeric characters (the 26 letters and the numbers zero through nine) in any order are valid. Now, the SR engine and the Records Application are waiting for the license number.

## 6. System testing

A variety of testing was performed to ensure the system operated correctly. The system was first tested in the laboratory using the Project54 Labcar, shown in Figure 7. The Labcar is the middle section of a police cruiser equipped with a lightbar and (muted) siren, radio, radar, GPS unit, as well as the Project54 system. The Labcar was used to functionally test the Project54 system as well as to test the speech recognition accuracy of various

commercial SR engines when used with the Project54 system [10].



**Figure 7 Labcar**

Testing in the Labcar was followed up by off-road testing by Project54 team members, and ultimately on-road live testing. On-road testing was performed by officers of the New Hampshire State Police (NHSP) over a period of approximately two years, between January 2001 and the spring of 2003. Qualitative on-road testing results indicated that both the system hardware and software function reliably.

We are currently performing quantitative on-road testing in the form of a large-scale data collection effort. The system is currently used in over 100 cruisers of the NHSP and other departments in the state of New Hampshire. Officers using the system are asked for permission to record their speech interactions with the system. Both the raw audio input and the SR engine's classification are saved for each speech input event. This data is processed in order to find out if the system is being actively used by the officers, if the speech recognition engine performs well and how the speech user interface design can be improved. This final question is very important since it promises to be the quickest way to improve system performance. We are complementing our data collection effort by spending time with officers on the road, observing how they perform their tasks and how they interact with the Project54 system.

As of March 2004 we have about 20000 officer utterances on file. Processing these utterances we found that the recognition rate of the system is 86%. We also found that recognition is imperfect primarily because of user errors. Approximately one third of the system's errors are due to

speech recognizer errors, while about two thirds are due to user error. We have identified three types of user errors. The most common error (54% of all user errors) was issuing a command that is not valid in any of the grammars. For example officers sometimes said "A Alpha," however the grammar defining the correct phrase for letters and numbers lists the NHSP official designation for the letter A, "A Adam." Officers also sometimes issued a command that is valid in some grammars but not in the current grammar (35% of all user errors). For example, officers sometimes wanted to turn the lights and sirens on but forgot that they cannot do so without first changing to the lights and siren control application. Finally, officers sometimes pressed the push-to-talk button too late and/or released it too early (11% of all user errors). When this happens, the SR engine receives a fragment of the utterance and it often fails to recognize the utterance. We were somewhat surprised that using the push-to-talk button was a problem since officers are used to operating the push-to-talk button of the in-car radio. But we found out that officers in the field sometimes make the mistake of pressing the radio's push-to-talk button too late and/or releasing it too soon. The resulting fragments are often difficult to understand even for dispatchers (human listeners).

After saying a phrase that was not part of the current grammar (or not part of any grammar) officers sometimes tried to "explain" to the system what they meant to say by using a different set of phrases. This is a reasonable approach when dealing with a human listener but it is hopeless with an SR engine that uses grammars. Another approach officers sometimes utilized is to speak louder to, or even shout at, the system. However, the SR engine almost never failed to recognize an utterance because it was softly spoken. Therefore, just repeating the same phrase often did not help. Also, speaking too loudly causes distortions in the microphone which in turn reduces the accuracy of speech recognition.

Results from processing the speech collected in the field will be used to improve officer training. If user errors are reduced by only 50% we will improve the system recognition rate to over 90%.

## **7. Conclusion**

We created a system that integrates electronic devices in police cruisers and allows the officers to interact with all the devices through a single speech user interface. This approach removes the multitude of user interfaces usually present in police cruisers and thus helps officers to

concentrate on the task they are trying to accomplish rather than on the mechanics of accomplishing the task.

The system uses a software architecture that provides one-to-one as well as one-to-many communications between applications. Messaging is accomplished by implementing the message coordinator and message handler COM objects. Using these objects made it possible to achieve a high level of modularity. The speech recognition and speech output related functions were implemented as an application that is called the Speech Input/Output Application. The SIOA interacts using the Project54 messaging with the speech enabled applications. Two support COM components are used in order to simplify access to the SR and TTS engines and allow easy support of new engines.

Testing showed that the system's hardware and software performs reliably in the field. Ongoing field testing is aimed at quantitatively determining, and eventually improving, the performance of the speech user interface. The system is currently in use in over 100 police cruisers in the state of New Hampshire.

## Acknowledgments

The authors wish to thank the US DOJ for funding Project54 through grants 1999-DD-BX-0082, 2001-LT-BX-K010 and 2002-CK-WX-0104.

## References

- [1] M. Weiser, "The computer for the 21st century," *IEEE Pervasive Computing*, Vol.: 1, Issue: 1, pp. 19-25, 2002
- [2] A.L. Kun, W.T. Miller III, W.H. Lenharth, "Project54: standardizing electronic device integration in police cruisers," *IEEE Intelligent Systems*, Vol.: 18, Issue: 5, pp. 10-13, 2003
- [3] M.J. Hunt, "Some Experience in In-Car Speech Recognition," IEE Colloquium on Interactive Spoken Dialogue Systems for Telephony Applications, pp. 4/1 - 4/9, London, Great Britain, November 1999
- [4] P. Pollak, P. Sovka, and J. Uhlir, "Noise Suppression System for a Car," Third European Conference on Speech Communication and Technology - EUROSPEECH'93, pp. 1073-1076, Berlin, Germany, September 1993
- [5] J. Smolders, T. Claes, G. Sablon, and D. Van Compernelle, "On the Importance of the Microphone Position for Speech Recognition in the Car," IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-94, pp. 1/429 -1/432, Adelaide, Australia, April 1994
- [6] T.E. Starner, "The role of speech input in wearable computing," *IEEE Pervasive Computing*, Vol.: 1, Issue: 3, pp. 89-93, 2002
- [7] A.L. Kun, W.T. Miller, III, W.H. Lenharth, "Modular System Architecture for Electronic Device Integration in Police Cruisers," Proceedings of the 2002 IEEE Intelligent Vehicle Symposium, Versailles, France, June 18-20, 2002
- [8] A. Pelhe, A.L. Kun, W.T. Miller, III, "Project54 System Software Architecture," Proceedings of the Winter International Symposium on Information and Communication Technologies, Cancun, Mexico, January 5-8, 2004
- [9] M.E. Martin, F.C. Hludik, and W.T. Miller, III, "The Project54 common interface for the Intelligent Transportation Systems Data Bus," IEEE Spring VTC2002, Birmingham, AL, 2002
- [10] B.J. Vinciguerra, A.L. Kun, "A comparison of commercial speech recognition components for use in police cruisers," NDIA Intelligent Vehicles Symposium, Traverse City, MI, June 9-12, 2003