DISTRIBUTED COMPONENTS FOR RETREIVAL OF DRIVER’S LICENSE DATA USING A HANDHELD COMPUTER

Kriste Krstovski
ECE Department
University of New Hampshire
Durham, NH 03824 USA
kristek@unh.edu

Andrew L. Kun
ECE Department
University of New Hampshire
Durham, NH 03824 USA
andrew.kun@unh.edu

W. Thomas Miller III
ECE Department
University of New Hampshire
Durham, NH 03824 USA
tom.miller@unh.edu

ABSTRACT
The Project54 system allows the integration of electronic devices in police cruisers. Both the software and the hardware system architecture are modular. The software is composed of components running on an embedded computer and distributed components deployed on a handheld computer. The hardware architecture utilizes an in-car serial bus to connect various in-car electronic devices and a wireless connection to allow interaction between the handheld computer and the in-car computer. This general framework was used to integrate a two-dimensional barcode scanner into the system. The barcode scanner is physically connected to the handheld computer. Once integrated into the system the barcode scanner is used synergistically with the in-car digital radio to allow officers to efficiently and safely perform records queries.

Keywords:
Modular system software, distributed components, handheld computer

INTRODUCTION
Project54 is a collaborative research and development effort between the Consolidated Advanced Technologies Laboratory (CATLab) at the University of New Hampshire and the New Hampshire Department of Safety (NHDS). The Project54 effort aims to incorporate embedded mobile computing equipment and wireless networking into New Hampshire State Police cruisers. The Project54 system integrates in-car based electronic systems, software and user interfaces to offer advanced support for NH State Troopers [1]. The system solves the problem of the law enforcement officer interacting with multiple user interfaces created by a large number of electronic devices found in today’s police cruisers (lights, sirens, digital radios, scanners, radars, mobile computers, GPS units, barcode readers, video recorders, etc). In the Project54 system, these devices are connected through the in-car Intelligent Transportation Systems Data Bus (IDB) [2]. The devices are controlled by integrated software components running on an embedded computer. Inside the cruiser, the officer interacts with the system through a touch screen, a speech user interface [3], a keyboard and a mouse. The preferred means of interaction is through spoken commands and synthesized speech feedback. The system creates a working environment for officers in which they do not have to take their eyes off the road and hands off the wheel.

Due to the nature of their job, officers often find themselves outside their cruisers performing driver’s license checks, various investigations, etc. In these types of situations, the police officers’ safety and efficiency could be improved if they were connected with the in-car electronic devices. For that purpose, we have developed a prototype system that provides this type of connectivity by deploying distributed components on a handheld computer that interact with the in-car components [4]. The officer stays connected with the in-car devices through the Project54 system running on a Windows Mobile based Personal Digital Assistant (PDA). For example, officers can control the lights and siren remotely simply by pressing buttons on the PDA screen.

In our effort to further improve the efficiency of the police officers outside the cruiser, we have developed software that runs on the PDA and interacts with a two-dimensional (2-D) bar code reader. Using this 2-D barcode reader the software can retrieve driver’s license data from barcodes (in PDF417 format [5]) printed on many states’ licenses (see Figure 1). The software can then use this data to perform driver’s license records queries. All an officer has to do is to request that the driver hold his driver license with the barcode side facing the officer, scan the barcode, and with a single click of a button a driver’s records query will be made and the result will be displayed on the PDA. The software architecture, and the available hardware implemented for this purpose, enables the officer to do the operation with a single hand.

Figure 1. – Sample New Hampshire driver’s license with 2-D barcode printed on the back (rectangle added for emphasis).
BACKGROUND
Ever since the introduction of handheld computers, work has been done on using them in law enforcement applications. For example, in [6] Kwasowsky et al define the problems of retrieving data at the scene of an accident using a PDA and integrating that data with an existing database. Landay and Kaufmann [7] describe issues they found developing an interface for handheld devices which are wirelessly connected to a workstation. Each application on the handheld devices has its own proxy representative on the workstation. In Project54 system applications send text messages to applications on another device using a proxy application on both sides. Myers [8] worked with applications used by individuals to interact with various handheld computers. The applications are connected and they share information synchronously. Similarly, in the Project54 system handheld and in-car applications have to be synchronized. In their paper [9] Newcomb et al discuss the process of designing a PDA application that helps shoppers in a grocery store do their shopping by having the PDA exchange information with a store database system. They address the issue of user interaction with the PDA application in situations where at least one hand of the user has to be free. Similarly, officers operating the handheld computer in the Project54 system need to have one hand free to be able to deal with physical threats to their safety.

PROJECT54 SYSTEM
Software System Architecture
The Project54 system software is completely modular and is based on Microsoft’s Component Object Model (COM) [10]. As shown in Figure 2, the software system comprises the software running on the embedded computer and software running on the handheld device.

Figure 2. - Project54 System Software architecture.

The architecture of the software running on the embedded computer and on the handheld computer is the same. In both cases, the Application Manager is at the center of the system. The Application Manager instantiates applications that control electronic devices (Application 1 through n in Figure 2) and other system applications, such as the Logger Application, the Speech Input/Output Application (SIOA), etc. It also handles message dispatching between the applications. The applications can communicate with each other via the Application Manager using text messages. The Proxy Application (both on the embedded and the handheld side) allows communication between applications on the embedded computer and applications on the handheld.

Inter-Application Messaging
The software developed for the embedded PC and handheld computer have the same COM objects and interfaces. Two COM objects were implemented in order to provide the two-way communication between applications: the message coordinator and the message handler objects. The Application Manager implements the message coordinator COM object, which provides a means for receiving messages from individual applications. The applications implement a message handler COM object which provides a means for receiving messages from the Application Manager. The Application Manager handles receiving messages via the message coordinator object and handles sending messages via the message handler objects, as shown in Figure 3.

Figure 3. - The Application Manager handles message dispatching.

All Project54 system inter-application messages have a source application name, a destination application name, a message identifier string, and a message body string. A simple code model is Message(source, destination, identifier, messagebody). All four elements of the message are null-terminated binary sequences. All objects in our current system exchange text messages – that is null terminated strings. This simplifies debugging in a system that contains a large number of objects written by many developers. However, the elements of the message can be any null-terminated binary sequence.

The Proxy Application
The handheld and the embedded PC use the 802.11b standard for wireless communication. The Proxy Application runs on both computers. It has the role of a network server on the embedded side and network client on the handheld side. The in-car Project54 Proxy Application supports Project54 system messaging with remote applications via UDP/IP network packets. The Proxy Application listens continuously for incoming UDP/IP packets at a specific network address. Incoming UDP/IP packets from remote applications are converted by the Proxy Application into Project54 system messages.
which are then submitted to the Application Manager. Outgoing messages delivered to the Proxy Application by the Application Manager are converted by the Proxy Application into UDP/IP packets and sent to the appropriate remote applications. The Proxy Application on the handheld side has the role of a network client and does the same message conversion and forwarding as the in-car Proxy.

**DRIVER’S RECORDS QUERIES**

The Records Application is the portion of the Project54 system software running on the in-car computer that enables police officers to do driver’s records queries inside the cruiser. The Records Application is used as the user interface that allows the officer to connect to the State Police records database through the in-car digital radio. The digital radio is integrated into the Project54 System Software through the IDB [2] as shown in Figure 4. The Project54 software includes the IDB component that allows each application to interact with the IDB. The connection between the Project54 software and the digital radio is accomplished through the Records Query Application component. This component is in charge of controlling the network connection with the state police headquarters. Features of the Records Query component include monitoring the radio connection for errors, maintaining persistent reliable communication over the radio, control and handling of delays, and other packet switching network related issues. The officer can submit a query through the Records Application. The officer provides the data necessary for the query by either typing in the data or by using the speech user interface. This query is forwarded to the Records Query Application which in turn sends it over the digital radio to a server. Data from the server arrives through the digital radio and the Records Query Application sends it to the Records Application for display.

Another way of doing records checks is through the Records Application that is running on the handheld computer (Figure 4). This Records Application also serves as a user interface that allows the officer to submit queries and to receive feedback. The information exchange between the Records Application running on the handheld computer and the Records Query Application on the embedded computer is accomplished through the proxy applications and the wireless link.

When performing the remote records query procedure, officers have to enter data from a driver’s license into a form displayed on the handheld computer. Without a barcode scanner this would require them to look at the Records Application GUI and tap on a miniscule keyboard. However, in many cases officers need to keep their eyes on their environment in order to be able to act in potentially dangerous situations. They also need to keep at least one hand free in order to be able to respond to any physical attack. Thus, for officers, using one hand to hold the handheld computer, looking at a GUI, and using the other hand to tap on the screen with a stylus is often impractical.

The Scanner Application solves this problem by having the Scanner Application utilize the barcode scanning of the 2-D barcode found on the driver’s license. Having a barcode scanner allows the officers to keep their eyes on the environment and it also allows them to keep at least one hand free. The officer can request that the driver show the driver’s license with the PDF417 barcode facing the officer. Next the officer needs to trigger the scanning of the barcode. This can be done using the Records Application. The Records Application will send a message to the Scanner Application to retrieve driver’s data encoded on the driver’s license barcode. The driver’s data obtained from scanning will be sent to the Records Application and the driver’s records query process can be executed using the Records Query Application. Shown in Figure 5 is a New Hampshire State Trooper scanning the barcode on a driver’s license using one hand.

**SCANNER APPLICATION**

The Scanner Application controls the operation of a 2-D barcode scanner, extracts driver’s information fields from the data obtained from the scanner, and presents the data to the user. In order to create an algorithm that will reliably extract driver’s information fields from the
encoded data, the application development involved analyzing the PDF417 symbology [5] and the American Association of Motor Vehicle Administrators (AAMVA) standard for encoding driver license information [11]. The structure of the Scanner Application as part of the Project54 Software System is shown in Figure 6.

The Scanner Application provides an interface between a commercial 2-D barcode scanner with a Compact Flash (CF) interface (Socket’s In Hand Scan Card Reader/Imager) [12], and the rest of the Project54 System software. Using a 2-D barcode scanner with a CF interface that runs in Windows Mobile environment we can utilize the CF slot found on almost every Windows Mobile based handheld device. Hence, the scanner is attached to the handheld computer and the user can control it using one hand. The child window created as part of the Scanner Application creates the needed interface between a 2-D barcode scanner and the rest of the Project54 System software. It hosts the Application Programming Interface (API) of the 2-D barcode scanner. The interface between the scanner’s API and the rest of the Scanner Application is done through Windows messaging. The Scanner Application controls the barcode scanner by posting system messages to the child window. The child window processes these messages through its WindowProc function. There are two types of system messages that the Scanner Application sends to the child window. These messages with their description are listed in Table 1.

<table>
<thead>
<tr>
<th>System Message</th>
<th>What it is used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIGGER.Scanner</td>
<td>Triggers the barcode scanner</td>
</tr>
<tr>
<td>CLOSE.Scanner</td>
<td>Deinitializes the barcode scanner</td>
</tr>
</tbody>
</table>

Table 1 - Messages being sent to the Scanner Application child window.

The “TRIGGER.Scanner” message is posted in response to the “SCAN” button being pressed or a “TRIGGER” message being received from another application through the Project54 system messaging facility. “CLOSE.Scanner” deinitializes the barcode scanner and is posted as part of the Scanner Application closing procedure.

Scanner Application Messaging

Project54 system software applications can utilize the functionality of the 2-D barcode scanner using the Scanner Application. Messaging between the Scanner Application and the rest of the Mobile Project54 Software System applications is done through the Project54 application messaging facility. This is depicted in Figure 6 where the Scanner Application is connected with the Application Manager. This section describes the way messaging between the Scanner Application and the Records Application is done.

The Records Application enables officers to do driver’s records queries through two of its windows, shown in Figure 7. The “Oper.OLN” window performs a driver’s records query using the operator’s license number. When doing this type of records query the officer has to enter the license number in the “OLN/” field and the license issuing state in the “LIS/” field.

Queries by operator name are done through the “Oper.Name” records application window. In this case the officer is required to submit driver’s name (“First/”, “Middle/” and “Last/” fields), driver’s date of birth (“DOB/” field), driver’s gender (“SEX/” field) and license issuing state (“LIS/” field). Both windows have the “SCAN” button created as part of their GUI.

When the “SCAN” button is pressed, the Records Application sends a “TRIGGER” command to the Scanner Application. Shown in Table 2 are the elements of the message sent by the “Oper.OLN” window.

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Message ID</th>
<th>Message Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>records</td>
<td>scanner</td>
<td>OLN</td>
<td>TRIGGER</td>
</tr>
</tbody>
</table>

Table 2 - Format of the Message being sent by the “Oper.OLN” window.

Included in the message is the name of the Records Application window that requested a scan event. When utilizing the barcode reader, the application can use the ID field as a tag to specify what part of the application requested the scan event. The Scanner Application makes use of this information to send the appropriate driver’s
For example, when triggered by the “Oper.OLN” window the Scanner Application responds by sending two messages. Both of these messages use the ID field of the message to tag the destination part of the application that requested the scan event and the type of driver license data being sent back. The returned message ID is constructed by taking the ID field of the received message and appending the appropriate Bar Code ID of the driver license data specified in the American Association for Motor Vehicle Administrators (AAMVA) National Standard for Driver License Identification cards. Shown in Table 3 are the elements of the messages sent by the Scanner Application in response to a “TRIGGER” message received by the “Oper.OLN” Records Application window.

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Message ID</th>
<th>Message Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanner</td>
<td>Records</td>
<td>OLNDAQ</td>
<td>Driver’s License Number</td>
</tr>
<tr>
<td>Scanner</td>
<td>Records</td>
<td>OLNDAJ</td>
<td>Driver’s License State</td>
</tr>
</tbody>
</table>

Table 3 – List of messages being sent by the Scanner Application to the Records Application “Oper.OLN” window.

When received by the Records Application, the message ID is examined and the appropriate steps are taken by the application. In the case of the “Oper.OLN” window requesting scan event the driver license data in the first message retrieved is displayed in the “OLN/” field. The second message is to be displayed in the “LIS/” field. When triggered by the “Oper.NAME” Records Application window the Scanner Application responds with four messages which contain the appropriate driver’s data for the six fields required when performing operator by name records query. Shown in Table 4 is the list of messages sent by the Scanner Application to the “Oper.NAME” Records Application window.

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Message ID</th>
<th>Message Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanner</td>
<td>Records</td>
<td>NAMEDAA</td>
<td>Driver License Name</td>
</tr>
<tr>
<td>Scanner</td>
<td>Records</td>
<td>NAMEDBB</td>
<td>Driver’s Date of Birth</td>
</tr>
<tr>
<td>Scanner</td>
<td>Records</td>
<td>NAMEDBC</td>
<td>Driver’s Gender</td>
</tr>
<tr>
<td>Scanner</td>
<td>Records</td>
<td>NAMEDAJ</td>
<td>Driver’s License State</td>
</tr>
</tbody>
</table>

Table 4 – List of Messages being sent by the Scanner Application to the “Oper.NAME” window.

**Scanner Application GUI**

The Scanner Application GUI is used for viewing the status and to test the functionality of the barcode reader. Furthermore, the officer can use this GUI to compare the information printed using characters on a driver’s license and the information encoded in the 2-D barcode thus verifying the authenticity of the driver’s license. The Scanner GUI contains the “Main Screen”, “SCAN” and “License Check/Scanner Status” buttons. The Scanner GUI runs in two modes - “Scanner Status” and “License Check” mode. When in “Scanner Status” mode the text box window is used for displaying scanner status messages. Status messages indicate whether the scanner hardware is present, is the scanner initialized or not, what the scanning trigger source is, etc. These messages depend on the type of the barcode reader used.

The officer can switch the Scanner Application GUI in “License Check” mode whenever s/he wants to view the driver’s license data encoded and verify its authenticity. Pressing the “Main Screen” button displays the Main Screen application GUI. Triggering the barcode is done through the “SCAN” button. Triggering the scanner from the Scanner Application itself has two purposes depending on the mode of the scanner GUI running. We can test the functionality of the barcode scanner and we can view the driver’s data encoded on the PDF417 bar code. Shown in Figure 8 are snapshots of both modes of the Scanner GUI running on a Pocket PC 2003 PDA.

![Scanner Application GUI](image)

**CONCLUSION**

The modular hardware and software architecture of the Project54 system allowed us to integrate the 2-D barcode scanner into a system that includes in-car electronic devices and a handheld computer. The 2-D barcode scanner can now be controlled through the integrated system - the barcode scanner is controlled by the Scanner Application, which is a standard Project54 application that implements COM-based messaging. More importantly, the barcode scanner interacts with the system synergistically. It interacts with the in-car digital radio in performing driver’s records queries. Having both a stand-alone barcode scanner and a stand-alone digital radio in a cruiser would enable officers to perform records queries more efficiently than by reading information from a license to a dispatcher. However, with the Project54 system we were able to incorporate both devices in the system and allow them to interact. This in turn made them more powerful tools than they would be as stand-alone devices.

The functionality of the barcode scanner integrated into the Project54 system was tested on sample New Hampshire driver licenses in a laboratory environment.
We expect that by the end of 2004 our tests will involve police officers using the system for their everyday work. The Scanner Application can be useful in introducing new functionality to the Project54 system. For example it could be used in conjunction with an application that can print citations. This new Citation Application may request that the Scanner Application send it the scanned data and may then use this data to populate fields in the citation.

REFERENCES


