Using Mobile Devices to Manage Traffic Infractions

Stefânia Marques
Federal University of Campina Grande, Brazil

Sabrina Souto
Federal University of Campina Grande, Brazil

Miguel Queiroga
Federal University of Campina Grande, Brazil

Hyggo Almeida
Federal University of Campina Grande, Brazil

Angelo Perkusich
Federal University of Campina Grande, Brazil

INTRODUCTION

Mobile computing is one of the recent technologies with the most impact on people's lives. Several research and industrial applications are benefiting from mobile computing, supporting various human daily activities. Transit law enforcement officials can benefit from the availability of powerful mobile devices, such as smart phones and PDAs, to help them to execute their daily tasks. In such a scenario, an official can verify a driver's data record and issue tickets online.

In this article we describe the SM-FIT system that makes it possible for transit law enforcement officials to perform online queries about potential infractions of a driver of a vehicle by using a mobile device. Queries are performed based on a unique identifier: the driver's license number.

The system is implemented based on a client-server paradigm, where mobile devices are clients and servers are base stations. Clients must have a local database to store each result of a query, when needed. Each registry stored has the following attributes: a unique identifier, the number of the vehicle's plate, the date and time that the officer registered the infraction, and the status of the infraction. Besides, photographs can be stored, digitally signed, and transmitted to a database for future prosecution.

The remainder of this article is organized as follows. We first outline some background concepts related to the system's development. We then present the proposed system architecture and functioning, and discuss some trends related to future research in this area. We close with some final remarks.

BACKGROUND

This section describes briefly some concepts related to J2ME and, more specifically, MIDlets. Such technologies have been used for developing the proposed system.

J2ME

J2ME is a development platform based on Java Technology for developing mobile and embedded applications. It focuses on two types of devices:

- **High-End Consumer Devices:**
  - CDC (connected device configuration);
  - interactive TVs, videophones, wireless devices;
  - a large variety of user interfaces;
  - typical memory of 2 to 4 Mb; and
  - persistent connection, generally TCP/IP.
- **Low-End Consumer Devices:**
  - CLDC (connected limited device configuration);
  - cell phones, bidirectional pagers, PDAs, and so forth;
  - limited processors (8 to 32 MHz);
  - limited memory;
  - lazy connection, intermittent (9600bps) and generally not based on TCP/IP; and
  - powered by batteries.

The J2ME platform includes flexible user interfaces, a robust security model, a broad range of built-in network protocols, and extensive support for networked and off-line

applications. Besides this, applications based on J2ME specifications are written once for a wide range of devices.

MIDlets

Java applications running on MIDP devices are known as MIDlets, which consist of at least one Java class and have to be derived from the abstract class *javax.microedition.midlet.MIDlet*. These MIDlets use an execution environment within the Java Virtual Machine to control the application's lifecycle through a set of methods implemented by this MIDlet.

MIDlets can also use methods to obtain services from the environment. A group of related MIDlets can be put together in a MIDlet suite, which is packaged and installed in (or removed from) a device as a unique entity. MIDlets in a suite share all static and dynamic resources in their environment:

- Execution data can be shared by MIDlets, and the usual Java conventions of synchronization can be used to control data access.
- Persistent data can also be accessed by all MIDlets in a suite.

All files in a MIDlet suite must be within a JAR package. These packages contain the classes of the MIDlet and other resources, like images, and a manifest file. This manifest file contains a list of attributes and definitions to be used by application managers to install the JAR files in the device.

Security in MIDlets

The JAVA security model in its standard edition (J2SE) is too expensive in terms of costs for memory allocation, and it requires configuration knowledge that is not present in users of mobile devices. Thus, neither CLDC nor MIDP include these functionalities.

Cryptography of public key and certifiers are not available as default, so it is necessary to pay attention when installing MIDlets and, preferentially, only accept software from trustable fonts. MIDP 2.0 included the https protocol that helps to diminish these problems.

SYSTEM DESCRIPTION

The SM-FIT is a system that makes it possible for transit law enforcement officials to register transit irregularities in a local database. Each record of this database consists of the number of the vehicle's plate, a code of the infraction, date and time that the infraction occurred, and the photography of the vehicle involved in the corresponding infraction.

The system is implemented based on a client-server paradigm, where mobile devices are clients and servers are base stations. At the end of the day, transit law enforcement officials send the registers of their mobile devices to the server in order to make available more space in their devices for local databases.

Consider Figure 1 to understand the system functioning. Through the starting screen of the system (1), it is possible to access the system main menu (2). This menu of options includes: *Add a record, Send records,* and *View records*.

If the user chooses the first option (2a), to add a record, it is shown a screen with the following data to be filled: the vehicle's plate involved in the infraction, the code of the infraction, date and time of the infraction (this information is taken automatically from the mobile device), and the photography of the vehicle involved in the infraction.

If the user wants to send all the records stored in the mobile device to the server (2b), it is shown a screen that requests the IP address of the server. Then, the process of sending the information to the database is initiated.

Finally, if the user chooses to visualize the records of the mobile device (2c), it is shown a screen with a list of records contained in the local database. This visualization is taken in a way that each record is shown individually on the screen.

FUTURE TRENDS

Future research and development of new technologies may make possible the transferring of streams from mobile devices to the server. In this way, the system described here could store videos of infractions or even the voice of the transit law enforcement official describing how the infraction occurred. For this kind of improvement, it would be necessary to consider efficient lower battery consumption mechanisms.

Another point to be considered is cryptography in mobile devices, which would increase security during the transference of records from the mobile devices to the server. Considering that it is an industrial application, security is an essential requirement.

In the context of the driver, a module for drivers to consult the situation of their licenses or cars could be made available. This module would make available all the information contained in the server database.

CONCLUSION

The usage of mobile devices to manage traffic infractions can bring some benefits, such as saving time on queries made to check if drivers have any previous infractions. Another benefit is that using a mobile device that can take pictures, when the transit law enforcement official registers a new infraction, he/she can also take a picture of the vehicle involved in the

infraction to prove in the future that it really occurred.

Regardless of application domain, there is a widespread use of mobile devices and an increasing need for real-time answers wherever a person is. This makes it necessary to use a technology like the SM-FIT system. It allows for transit law enforcement officials to get real-time answers for their queries about the drivers' situations, and also about their vehicles. But this system can also be adapted to be used in many other application scenarios, such as to make restaurant reservations, access credit card accounts, and make payments.

REFERENCES

Alves, D. (2004). *Introdução ao J2ME.* Retrieved November 30, 2004, from http://www.conexaojava.com.br/conexaojava04/download/minicursos/Java2.Micro.Edition-Conexao.Java.2004.pdf

Borges, R. L. (2004). *J2ME na prática.* Retrieved November 30, 2004, from http://www.ucb.br/java/JavaDays/J2ME_RosfranBorges.pdf

Easy Process. (n.d.). *A software development process.* Retrieved from http://dsc.ufcg.edu.br/~yp

Gomes, H. M. (2004). *Arcabouços de software para desenvolvimento de aplicações embarcadas.* Retrieved November 30, 2004, from http://www.dsc.ufcg.edu.br/~hmg/disciplinas/nokia/ASDAE.pdf

XP1. (n.d.). *A software development process.* Retrieved November 30, 2004, from http://www.dsc.ufcg.edu.br/~jacques/cursos/2002.2/projii/xp1/xp1.html

KEY TERMS

Base Station: A centralized repository for the storage and management of information, organized for a particular area.

CDC: Connected device configuration.

CLDC: Connected limited device configuration.

IP Address: An Internet protocol address attributed to a client or a server in the client-server paradigm.

J2ME: Java Second Micro Edition.

MIDP: Mobile information device profile.

Personal Digital Assistant (PDA): A handheld device that combines computing, telephone, Internet, and networking features.