Using QR Codes as a Model for Vending Machine Control

Rebecca Fischer
Hiram College

Professor Margaret Martonosi
Princeton University

ABSTRACT

Quick Response (QR) codes are gaining popularity as a medium of information storage. Their potential has yet to be explored fully. This project aims to use QR codes as a means of controlling a vending machine. This will be accomplished by utilizing a QR decoder to decode the messaged and vend the selected item at the price indicated by the QR code.

KEYWORDS

Quick Response Codes
Vending Machines
Bluetooth
Multi Drop Bus
PC2MDB
Cell Phone

I INTRODUCTION

The aim of this project was to create a system that allowed users to purchase vending machine items by taking a picture with their cell phone. This would allow a secure means of cashless purchasing for most people. A lot of people carry cell phones, and a lot of those cell phones are equipped with camera's. Using these camera's for cashless purchasing would be convenient for many people who don't carry cash.

In this paper we present a system for implementing a QR code based cell phone purchasing system. Many elements of the implementation showed great promise, however there were many elements that were not able to be implemented in the alloted time.

II RELATED WORK

A group of researchers from University of Washington recently developed a system using mobile phones and QR codes in for the use of assisting with micro finance in rural India[2]. In rural India, there is not much technology, so QR codes are a relatively safe way of transferring money. Some of their early concerns with image warping plagued this project as well. There was a fundamental flaw in their work, however. QR codes are not very secure. It only takes basic Internet access and a printer to create QR codes. This leads to a large chance of fraud even in rural India where the technology is not highly available.

III BACKGROUND INFORMATION

Quick Response(QR) codes were developed by Denso-Wave in 1994. QR codes have been used primarily in the shipping industry as a means of describing what a shipment is and it's destination. They have also been used in Japan as a means of expediting data entry into cells phones. This technology is very useful for encoding data as character recognition is not very reliable.

Vending Machines have been around for decades and are a popular means of purchasing items without human involvement. They are also immensely popular in Japan.

IV OVERVIEW

Figure 1(next page) is a chart representing an overview of how the proposed system would function. The chart is color coded as to the type of item each entity is.

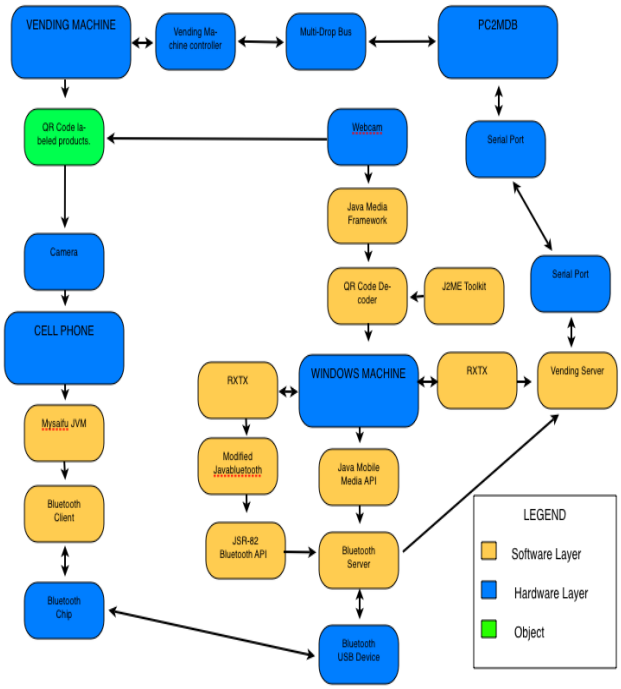

Figure 1

V WINDOWS MACHINE

V.1 QR DECODER

V.1.a Decoder

A QR decoder uses the features of the QR codes along with error correction to properly view the QR code. The decoder chosen for this project was an open source one provided by Open Source QR Code Library[4]. It was written in Java. The decoder read the code in byte by byte. The decoder first looked for the finder pattern. These were three double boxes in the corners of the QR code that allows the reader to determine the correct orientation of the QR Code and a bit of the code connecting them. After the positioning had been determined the textual encoding was read. The decoder used Read/Solomon error correction to determine accuracy.

The decoder came with several sample programs. It required use of the J2ME wireless toolkit version 2.2 in order to compile these demos. One was a line based decoder in which a file is specified in the actual code. This was the demonstration that was attempted first. This program allowed the use of either a file or a url in order to decode. The URL aspect of the program was successful immediately. The file aspect was more difficult as the file specification of Java is rather particular about escape characters.

After the successful modification of the command line demonstration, the GUI version was attempted. The GUI version was poorly designed. It was difficult to determine the appropriate place to press in order to select a file. The URL selection was straightforward and worked on the sample QR code. Once the file selection was found, that it was proven to work as well. The GUI was modified by changing the instruction text and creating an actual button for the file browsing portion.

V.1.b Webcam Support

After the decoder was modified, it was decided that the integration of a webcam to take photographs of the QR codes was necessary. The Java Media Framework was used to detect the webcam and use it to take a photograph[3]. The resulting photograph was saved as a jpeg.

Once the camera was able to take pictures the GUI was modified into a QR decoding program. The webcam took a picture and saved it under the same name each time. As soon as the picture was taken the QR decoder was activated and decoded the image. The results were then displayed to the user.

With this version, the biggest difficulty of dealing with QR codes became apparent. In order to get the image to decode properly the picture had to be taken in a very straightforward way. There could be very little skewing of the image or the appropriate finder patterns would not be discovered.

V. BLUETOOTH

	Cell Phone time in mils	PC time in mils
18 Character QR Code	11660	25.02
94 Character QR Code	42040	41.9

250 Character QR Code	90620	80.74

Figure 2

It was decided that Bluetooth would be used to transfer the photos taken on the phone device to be decoded on the PC. This was decided due to the increase in speed in decoding on a PC vs. a cell phone (figure 2). The Bluetooth server would be running on the PC, while a client ran on the cell phone. The Bluetooth protocols would provide security for the vending machine. The Bluetooth address of the users phone would be used as a security check, alongside of a password/key system.

V.2.a JSR-82 Bluetooth API

Sun provides a standard Bluetooth API for java called the JSR -82 Bluetooth API. This provides the methods necessary to address the Bluetooth chip. This API, however, is not implemented by Sun. It was necessary to find a third party implementation.

V.2.b JavaBluetooth

JavaBluetooth is a popular implementation of the JSR-82 Bluetooth API. It is built on top of Sun's Java communication(comm) API. It uses serial ports to address the Bluetooth chip. Unfortunately, the java comm API is not implemented by Sun. JavaBluetooth had to be modified to use a different communications package.

V.2.c RXTX

RXTX is an implementation of the java communications API. It also has it's own methods for serial port communication. JavaBluetooth was modified to use this package instead of the one offered by Sun.

V.2.d Bluetooth Server

The Bluetooth server in this project was never fully functional. It attempted to open a connection to a remote Bluetooth device(the cell phone) and allow for file browsing. It was never able to communicate with the cell phone. This could have been an issue with any of the layers used to create the server, or an issue with the corresponding Bluetooth client.

V.3 VENDING

The Bluetooth sever was intended to communicate with the vending machine server and allow items to vend. The vending machine server was never developed due to issues pertaining to the PC to vending machine communication.

VI PC2MDB

PC2MDB is the name of a commercial board designed to communicate between the multi-drop bus of the vending machine and the PC. It uses a serial port to communicate with the PC. It was connected using a Molex connection and a serial port connector and was able to be powered on.

VI.1 SERIAL PORT

After the board was powered on serial port communication was attempted between the control board of the vending machine and the PC. Communication was attempted using HyperTerminal. After a self-test was run it became apparent that the PC was able to communicate with the PC, but not the vending machine.

VI.2 MULTI DROP BUS

The PC2MDB board connects to the multi-drop bus of the vending machine controller. This connection allows communication between the vending machine and the PC.

VII VENDING MACHINE

The vending machine dispenses products to the user. Items would be priced individually, allowing for more variety. The vending machine used was the TT9 Tabletop Vending Machine.

VII.1 CONTROLLER

The controller of the vending machine board is its brain. It handles all commands executed by the machine. This was were the project hit it's incurable snag. The controller of the vending machine was unable to poll a cashless device. This meant that the PC2MDB board was incompatible and no communication between the vending machine and the PC could take place.

VII.2 PRODUCTS

Products in the vending machine were labeled with QR codes. This allowed them to be priced individually. The codes were as large as possible to make taking a viable picture of the QR code possible.

VIII CELL PHONE

The phone used in this project was a Windows 2003 ME enabled smart phone. It had a built in camera and a 400mghtz processor.

VIII.1 MYSAIFU JVM

Mysaifu JVM was developed by freebeans[1]. It allowed the use of java on the phone.

VIII.2.a Limitations

The Mysaifu JVM had a few limitations and errors. It was limited in it's ability to display the java.swing interface items. It also had an error in it's picture class that made it impossible to read in jpg or gif files. It was, however, able to decode png files.

VIII.3 QR DECODER VS BLUETOOTH

It was decided after developing the phone version of the QR code reader, that it was simply not fast enough to practical. It would be easier to use Bluetooth to transfer the files and then decode them on the PC. Figure 2 illustrates the fact that it was more that 1000 times faster to do the decoding of the large QR code on the PC. This is enough on a gain to justify the addition few seconds of a Bluetooth transfer.

IX CONCLUSIONS

There were many successes and failures within this project(figure 3 right). The QR code image reader and webcam reader worked very well. The Bluetooth was close to being functional. The main issue with this implementation was the control board on the vending machine.

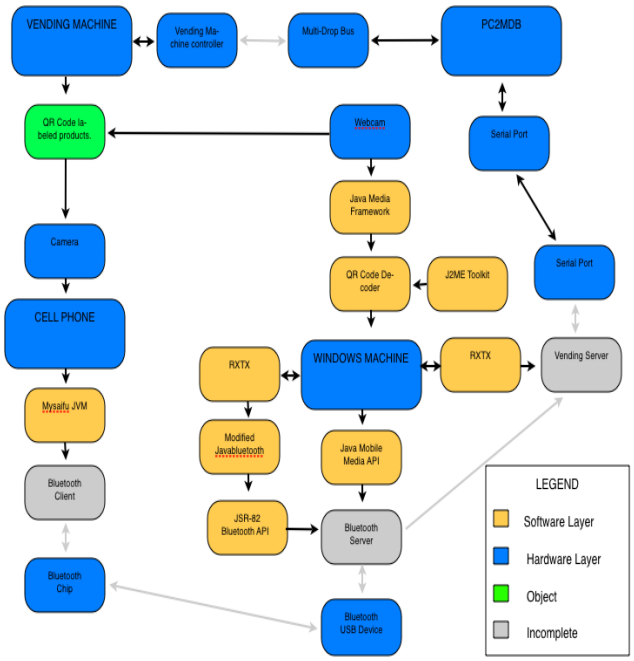

Figure 3

X FURTHER RESEARCH

X.1 SMS

Using SMS messages to send picutres for decoding, or decoded messages to the PC is one avenue for future research. It the time of this projest it was decided that it was unsuitable for small scale deployment due to the fact that messages have to be run though gateways that are signed by cell carriers. SMS would allow billing through ones cell phone company. This would require a lot of licensing by the various cell phone companies. This is a large cash investment for a small scale project.

X.2 Image Un-Warping

Image un-warping is one of the most important

areas of further research for this project. It was impossible to decode images that were even slightly warped using the current system. It was very common to experience slight warping due to using camera phones and no tripods.

X.3 C# QR Decoder

If a C# QR decoder were developed it would most likely allow for great speed increases while decoding on a windows mobile phone. This would be a great further area of research for the project.

X.4 New Vending Controller

If a new vending machine controller was purchased, it would be possible for the vending machine and the PC to communicate. If communication were possible, the project had the potential of functioning.

XI RESOURCES

[1]Mysaifu JVM. 2005. freebeans. June 2007 <http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html>.

[2]Parikh, Tapan S., Paul Javid, Sasikumar K, and Kaushik Ghosh. Mobile Phones and Paper Documents: Evaluating a New Approach for Capturing Microfinance Data in Rural India. University of Washington. 2006. 14 May 2007 <http://portal.acm.com>.

[3]wbracken. "Here is the Source Code to JMF Webcam App + Saves Jpeg." Sun Java Developers Forum. 31 Aug. 2006. May 2007 <http://forum.java.sun.com/thread.jspa?threadID=764745&messageID=4361913>.

[4]Yanbe, Yusuke, ed. Open Source QR Code Library. Vers. 0.7.2. 19 Mar. 2007. May 2007 <http://qrcode.sourceforge.jp/#sec-3>.