

# **Device Independent Mobile Application Controller For Remote Administration Of A Server Over A GPRS Link Using a J2ME Cellular Phone**

Mr. S. Kulkarni  
[sameerkulk@rediffmail.com](mailto:sameerkulk@rediffmail.com)

Miss S. Diwan  
[surabhi\\_diwan@yahoo.com](mailto:surabhi_diwan@yahoo.com)

Prof. N.K. Bansode  
[nkb\\_1669@yahoo.com](mailto:nkb_1669@yahoo.com)

Department of Computer Engineering  
Army Institute of Technology,  
Dighi Hills, Pune.  
Ph. No. (020) 27157534 Ext 129.

## **Abstract**

Unhindered connectivity to a computer is the need of the hour. The ability to connect to a remote machine and control it will have infinite advantages. J2ME (JAVA 2 Micro Edition) is a platform created to enhance application development for mobile phones. Using this platform we have created an application which can control any desktop computer or a server by sending the required commands over an HTTP Connection.

## **1 Objective**

The main aim of the application is to remotely access and control different applications on a static IP PC by connecting to it over a GPRS link from a J2ME enabled cell phone. The possible applications are:

- i) Launching Winamp , WordPad, games or other similar applications.
- ii) Shutting down and restarting or logging off from your machine.
- iii) Formatting hard drives.
- iv) Running Internet Explorer with the required URL.
- v) Remote Database access including Query Processing and viewing the results on the cell phone.
- vi) Remote Desktop Connection from the Cell Phone.

In the present paper we propose a solution to control a computer by creating an all time functional link between the system and its administrator.

Section 2 lists the need for such an application. Section 3 gives the advantages of a remote application controller. Section 4 describes the targeted enterprises. Section 5 gives the comparison with other technologies that can be used for application development. Section 6 lists the advantages of choosing J2ME. Section 7 gives the prerequisites for running the application. Sections 8,9,and 10 give the complete s/w requirements and the procedure for developing the proposed application followed by the result and the conclusion.

## **2 Motivation**

Many applications like web hosting services, network servers, automated systems need to be monitored continuously. And to monitor them 24/7 by being physically present at the location is not viable. Therefore we propose to control such applications remotely by J2ME enabled mobile devices.

Adding a mobile remote control to an existing network can have many advantages as mentioned below:

## **3 Advantages Of A Remote Application Controller**

- Faster, decentralized decision making.
- Increased responsiveness to customers.
- Increased sensitivity to market changes.
- Lowered commuting costs/time for staff.
- Increased productivity.

#### **4 Target Enterprises (Commercial Viability Of the Idea)**

Newer applications like

- i) Online railway ticket booking,
- ii) Status checking
- iii) Credit card transactions
- iv) Information delivery systems (Internet)

are aimed at providing services to costumers and their availability must be an absolute certainty at all times for them to meet their end [4].Therefore our application would equip the administrators of these applications to have unhindered access to their systems.

#### **5. Current Work in the Field and Comparison with Contemporary Technologies**

i) The traditional implementation of a Remote Control for Electronic devices is using the Infra Red(IrDa) technology ; the major drawback of which is the requirement of Line Of Sight.

ii) A Mobile Remote Control has been implemented using a software called MoReCo [5].

iii)WAP (Wireless Application Protocol) which allows Internet Access but doesn't provide HTTP support this makes invoking server side programs cumbersome [1].

#### **6. Justification For Choosing J2ME**

Referring to the pros and cons of other technologies we realize that Java provides the best deal for the development of such an application with universal repercussions because

- 1) Java is an older platform , therefore it is robust and secure.(It's bugs have been dealt with over time).
- 2) Java is platform independent. The same application can be deployed on a whole plethora of J2ME compliant

phones without making any changes to the code.

3) J2ME is a toned down version of J2SE, J2EE, hence application development is on the same lines.

4) Java programs can deploy on both the wireless device and the application server, independent of host hardware and operation systems. Client-side and server-side Java applications can integrate seamlessly under consistent API designs

The present mobile phones can connect to the internet using mini browsers over GPRS or GSM links. Our solution is based on a program running on a mobile phone which opens an http connection to pass required information and commands to the server which in turn executes them and receive a response from the server, thus acting like a mobile remote control.

Hence looking at the large number of hardware profiles, operating systems, design and resource limitations of mobile phone models , platform independence is one need that has to be answered. This is the need answered by Java 2 Micro Edition.

#### **7. Prerequisites for the Remote Controller to Run**

There are only 2 deciding factors for the application to run

- 1) Requirement of a J2ME enabled cell phone.
- 2) A network which offers GPRS service (and not to forget network coverage..)
- 3) The system which we propose to control must be on the Internet.

In the present context the Jakarta tomcat server is running on the same system which we propose to control but we can expand this model by having a controller application running on the this machine which controls other applications running on other machine which may not be on the internet but are on the same network by using RMI/RPC in any way suitable to the system administrator.

## 8. Structure Of the Application

The architecture can be broadly divide into 3 layers as can be seen in Fig 1.1

- 1) The J2ME client.(MIDlet running on it)
- 2) The HTTP server (Java Servlet running on it)
- 3) The Static IP machine (on the Internet) with it's set of applications.

Let us take a look at each of these in detail [6].

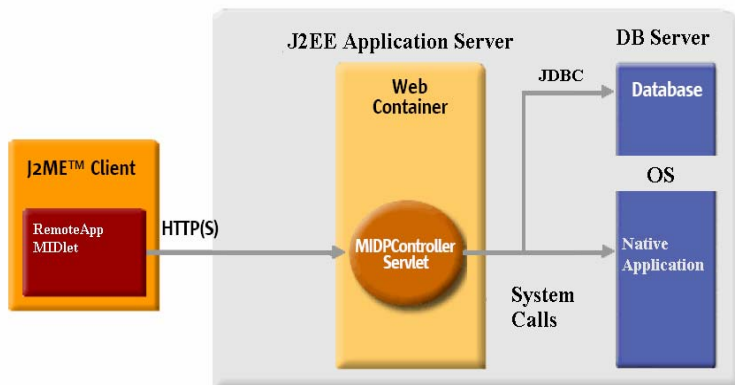


Fig. 1.1 Structure of the application.

### 8.1 HTTP Server Specifications

Any HTTP server which supports server side scripting can be used. In the present experimentation purposes we have used Jakarta Tomcat deployed with Java servlets. Such a server which needs to be controlled must be visible over the internet. It is possible to use the current application on Linux but experiments have been conducted only on Windows-XP.

## 9. Software Requirements

- i) J2SDK 1.5.0
- ii) JAVA wireless toolkit 2.2
- iii) Jakarta Tomcat Server

## 10. Building the Mobile Application Server Controller Step By Step

The MIDP application runs on the mobile device which acts as the controller. MIDlets communicate with JAVA Servlets through the HTTP protocol. The JAVA Servlet then passes the MIDlet's request to the local application controller that uses local or remote method calls to invoke the application software (on the same machine or on a different machine on the same network visible to the local application controller which was originally invoked) for the specified service eg. database server or the task manager. and prints a message on success. The steps are:

1. **Accepting User Input On The Mobile Phone.** This can be done in two ways
  - i) One way of doing this would be to get the name of the application to be controlled.
  - ii) To make a remote desktop connection and use a pointing device to select the application.

The MIDlet accepts input from the user, and the parsed parameters are appended to the URL which in this case is the string "winamp.exe" which is the name of the application which needs to be invoked by the mobile phone on the server.

2. **Connecting MIDlets to the server using HTTP** [3]. A new connection to the server machine was created with the following code:

```
HttpConnection conn =
(HttpConnection) Connector.open(url);
```

To send data over HTTP, use

```
conn.setRequestMethod(HttpConnection.POST)
```

for the POST method or

```
conn.setRequestMethod(HttpConnection.GET)
```

for the GET method.

Now we can send data to the server using the output stream from the connection object:

```
DataOutputStream os =
conn.openDataOutputStream();
os.write(StringToPost.getBytes());
```

Then, the server will execute the command and send the response data back to the MIDP client through the Internet.

## 11. Results



**Fig 1.2 The emulator used to control the various applications on the tested server**

It was found that a local Executable could be started by the servlet and once that happens then there can be any number of applications that could be controlled by this basic controller. In this case we have started Winamp (an MP3 playing software) by sending a request "winamp.exe" (Fig 1.2) from the mobile phone to the server and then the requested servlet invokes the winamp player (Fig1.3). Also another program was created that could control the mouse and send screenshots of the desktop to the cell phone With this kind of mobile control having been accomplished it will be possible to control any application remotely.



**Fig 1.3 Launched Application**

## 12. Conclusion

The ability of the J2ME enabled devices to connect to FTP, HTTP or even the use of raw sockets has given it the power to break all computational barriers in spite of the resource limitations, making small handsets synonymous (in computational power) to any high performance server.

In the present application we have managed to control an application on a remote machine using the mobile phone. But the same can be extended to control any computer on the network.

Hence with this application in mind it is not difficult to imagine controlling huge computing clusters from your fingertips anywhere, anytime .....

## References

1. "Wireless community networks" by Rob Flickenger published by O'reilly Publications.
2. December 2000 by DELL Technology on "Handheld devices: comparing the major platforms"
3. Writing Java Applications for the Communicator Getting Started with Mobile, Part Four.
4. "Mobile Computing and wireless Networking Systems Management" by Synchrologic.  
[www.mobileinfo.com/Vendor\\_Dir/management\\_sw.htm](http://www.mobileinfo.com/Vendor_Dir/management_sw.htm)
5.  
[http://moreco.moelholm.com/\\_internal/overview.html](http://moreco.moelholm.com/_internal/overview.html)
6. "Architecting Mobility" White paper by Bill Day <http://www.billday.com>