ISSN No. 0976-5697

Volume 1, No. 4, Nov-Dec 2010

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

# Remote Desktop Access using Mobile Phones and an Enhancement for Intra-Mobile Search

H.Mohamed Ameer Irshad
Student, School of Computing Sciences
VIT University, Vellore, India

R.Padma Priya
Student, School of Computing Sciences
VIT University, Vellore, India

V.Vijayasherly
School of Computing Sciences
VIT University, Vellore, India
vvijayasherly@vit.ac.in

Dr.N.Ch.S.N.Iyengar*
School of Computing Sciences
VIT University, Vellore, India
nchsniyr@vit.ac.in

***Abstract:*** In this era of mobility, an information retrieval system cannot be constrained to be running on one particular location alone. The system proposed in the project facilitates the user to access his personal computer's storage devices through his/her mobile phone when the user is remote. Also one can perform various operations like a desktop search, taking backup of mobile specific data and troubleshooting. The user can search for files in a remote machine for a particular keyword and download the files of his choice. Also if the mobile device is running short of memory the mobile specific data like Personal Information Manager (PIM) database (contacts, events, to-do lists), messages and file manager can be uploaded to the remote server. Remote troubleshooting deals with performing administrative tasks like removal of viruses, few file management tasks, etc. To accomplish this J2ME provides Generic Connection Framework (GCF) which is a flexible API for network connections. Also, as an enhancement to this system the user is facilitated to search one's own mobile in case of large file systems and flash memories. Thus a keyword-based search is implemented to provide a quick access to the PIM Database, messages and flash memory.

***Keywords:*** Remote Desktop, Intra-Mobile, Flash Memory, Mobile Backup, Mobile Search

## I. INTRODUCTION

Everyone is in need of portability of data so that he can have access to information anywhere, at anytime. In achieving portability, laptops and removable disks (pen drives, portable hard disks) play a great role. But assume a situation where you don't have an access to the above said portable device and you are in need of your data. In this situation, the hand-held devices like mobile phones, PDA's and pagers can be used to access our stored data.

As the user can carry his mobile phones or PDAs everywhere, these devices can be the best solutions to remotely access a desktop or a laptop. When remote access of a desktop is accomplished, one can perform various operations like searching for file, downloading them, few administrative tasks[1], trouble shooting, taking backup and similar tasks.

Though already many systems like this have come in the market, they do not solve the purpose to a great extent. Since the system use Bluetooth for connectivity, the user is obliged to stay within a coverage area. But as our system uses HTTP for connectivity, it is an anywhere, at anytime access.

Moreover the proposed subsystem intra-mobile search [3] is the need of the hour. Nowadays there are mobiles which allow storing a thousand messages and contacts and gigabytes of user data which includes multimedia. But to access one, the user has to keep navigating between screens due to lack of a proper interface. Thus intra-mobile search will facilitate the user to perform a keyword based search to have a quick access to the above said data.

## II. J2ME AND CONNECTIVITY (HTTP)

Due to many constraints like power, memory and display sizes in mobile phones, application development was complicated. Taking care of these constraints, J2ME was developed which provided a clear layer of abstraction between the application programs and the device's operating system Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

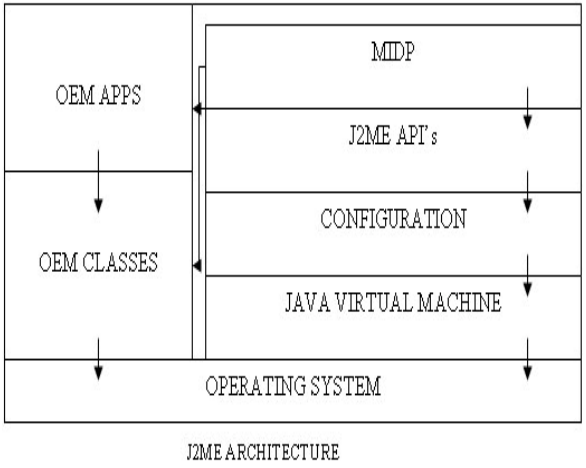


Figure 1. J2ME Architecture

Since this system deals with remote access of a computer, the mobile application should possess connectivity features. This is achieved by various API's in J2ME, one among which is Generic Connection Framework (GCF)[2]. Under GCF there are totally 7 interfaces (types of connections) that groups together classes of protocols with the same semantics.

| | |
|---|---|
| HTTP | Connector.open(Http://www.sun.com); |
| Sockets | Connector.open("socket://192.168.2.0:8081"); |
| Datagram's | Connector.open("datagram://192.168.2.0:8081"); |
| Serial ports | Connector.open("comm:0;baud rate=9600"); |
| Files | Connector.open(file://a.txt); |

Figure 2. URL Formats for Connections

In our system we use HTTP protocol to establish a connection with the remote desktop. Here we specify the IP address (and port number) or the domain name. The general syntax is given below.

**Connector.open("<protocol>:<address>:<parameter>");**

Serverside:

```
    doGet(HttpServeltRequest rq, HttpServletResponse rs)
  {
            doPost(rq,rs);
  }
  doPost(HttpServeltRequest rq, HttpServletResponse rs)
  {
           // code is done here
           String s = rq.getParameter("xxxx");
          //operations using s
  }
```

Based on this connection framework, three subsystems can be developed. Various operations which involve remote desktop access can be performed like performing a keyword based search, troubleshooting and administrative tasks, taking back-up of mobile specific data.

**Client-Side**:

```
String servurl="http://192.168.0.2:8084/xxxx/aaaa";
String param="?parameter="+xyz.toString();
String url=servurl+param;
conn=(HttpConnection)Connector.open(url);
//to obtain the results
is=conn.openInputStream();
while(filename=is.read())!=-1)
String.append((char)c);
```

## III.REMOTE DESKTOP SEARCH

Using the framework as in Figure 3 we can perform a keyword based search inside the files present in the remote machine. To do this search algorithm is defined inside the servlet. The servlet gets the keyword from the mobile phone as a parameter amd transfers the control to the search algorithm. None all the files in preferred locations are searched recursively in every sub-folder.

### STEPS INVOLVED IN DESKTOP SEARCH

```
drive="e:\\";
search (f);
{
f=contents (drive);
if (f.isDirectory())
search (drive+"\\"+f);
else
{
Check file extension;
open (f);
read (f);
if (read(f).contains(keyword))
Path=f.getAbsolutePath();
write the path in the browser;
}
}
```

### CALCULATING THE COST OF RECURSION

For a normal algorithm to calculate the factorial of a given integer the time complexity is given as:

$$T(n) = \sum_{i=1}^{n} T(n-1)$$

For this search algorithm: Let 'n' be the maximum number of iterations.Let 'k' be the possible number of Iterations, for all **k € 1≤k≤n.**

Let T(n)=expected running time of the whole algorithm for 'n' number of files. Assuming that all files are distinct.

If i<k is the sequence of n-i files.

Then the time Complexity

$$= 1/n\,[ \sum_{i=1}^{k-1} T(n-1) + \sum\nolimits_{i=k+1}^{n} T(n-1) ]$$

$$= 1/n\,[ \sum_{i=n-k+1}^{n-1} T(i) + \sum\nolimits_{i=k}^{n-1} T(i) ]$$

Thus for the number of files greater or equal to 2:-

$$T(n) \le C_n + Max\left\{ \frac{1}{n} \left[ \sum_{i=n-k_1}^{n-1} T(i) + \sum_{i=k}^{n-1} T(i) \right] \right\}$$

Splitting it as $1 \ge k-1 \ge k+1 \ge n$ .The results are written in to the browser which is read by the midlet using the Input Stream Connection. Again using the class File Reader the preferred file can be opened and accessed using the mobile itself.

### B. TROUBLESHOOTING

System administrators or even normal people when required to perform administrative tasks they need not use their laptops or desktops. Instead hand-held devices can be used to access the DOS command shell. Thus the user has a complete access over his machine.

Various commands like net, netsh, task scheduling, user management, file management, network and system administrations tasks can be performed.

The servlet should be able to run the dos commands after obtaining than from the mobile phone.

The general syntax looks like this:

**Process p=Runtime.getRuntime().exec("cmd/c" +command, null);**

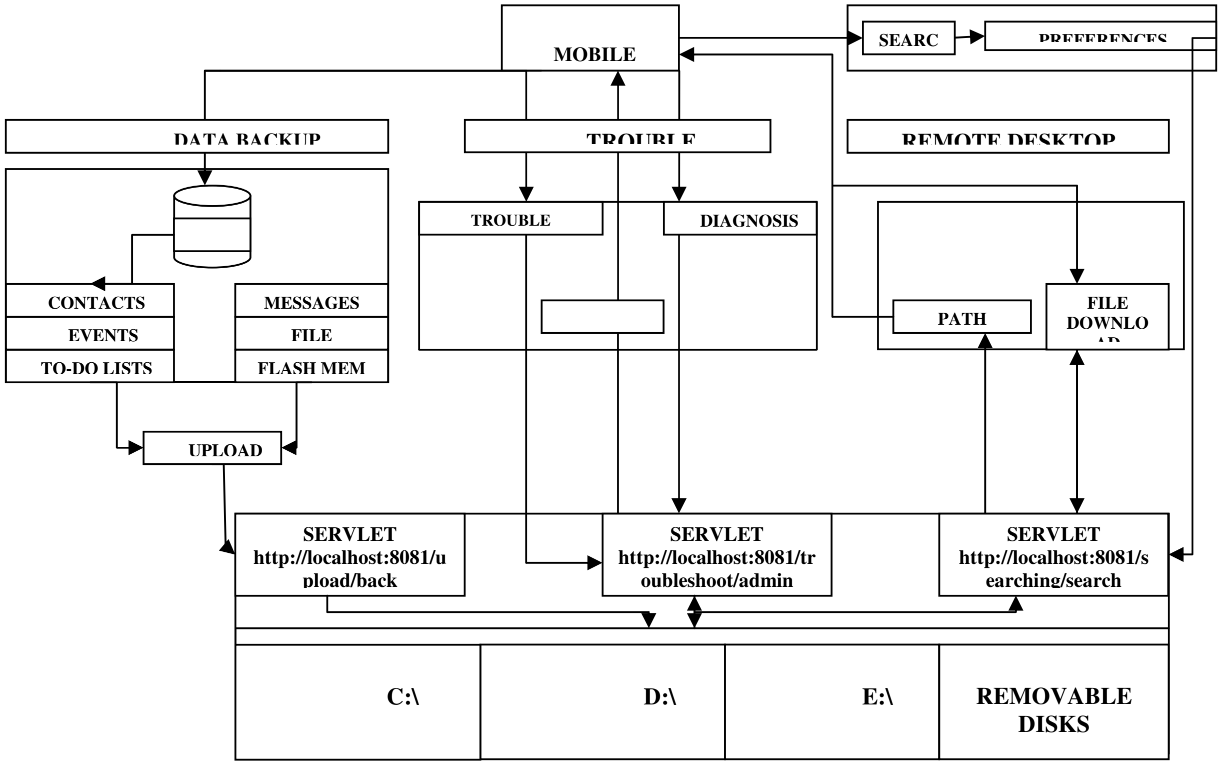


Figure 3. System Architecture of Remote Desktop Access Using Mobile Phones

## IV.TAKING BACKUP OF MOBILE SPECIFIC DATA

When the user is running short of memory in his mobile device, he need not delete existing data to add new ones. Instead a backup of his data can be taken on a remote machine and restore whenever he is interested. This system deals with data present in the following locations

a. PIM databases
   - ➢ Contacts
   - ➢ Events
   - ➢ To-do list

b. File manager
c. Flash memory
d. Messages

These data can be accessed in the midlet and sent to the servlet by appending to the URL. The servlet gets the parameters and stores the data in the hard drive in its native format. A proper interface is designed which will help the user to restore the data in the mobile phone when needed.

## V. INTRA MOBILE SEARCH

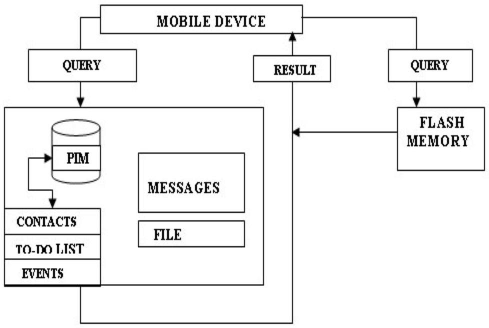


Figure 4. System Architecture

Unlike the previous modules, this system is a stand-alone Application independent of any servlet. This system deals with searching inside the mobile storage,.i.e.,PIM database, file manager,flash memery and messages.

The syntax to open the file manager is:
Fc= (FileConnection) Connector.Open
(file:///c:/pictures/,Connector.READ);

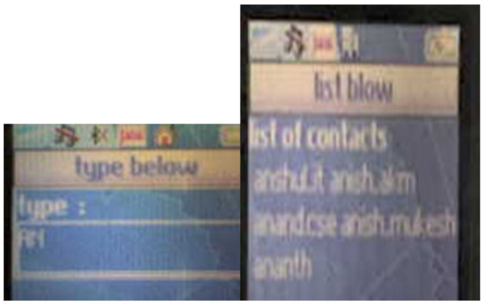
In case of Sony Ericsson mobiles the file manager is organized as follows:
For JP6 Models:
file:///c:/pictures/
file:///c:/sounds/
file:///c:/videos/
file:///c:/other/

For JP7 and JP8 Models:
File:///e:/ (For extendible memory slots)

## VI.EXPERIMENTAL RESULTS

Currently this project is under development and many of the above discussed modules have been successfully completed. Coding is being done in Netbeans5.5 with Mobility Pack. The built applications were tested in real-time by deploying them in various models of Sony Ericsson like K320i (figure 5.), k550i, k750i and W700i.Also in Nokia mobiles like N73 and the results were significant.

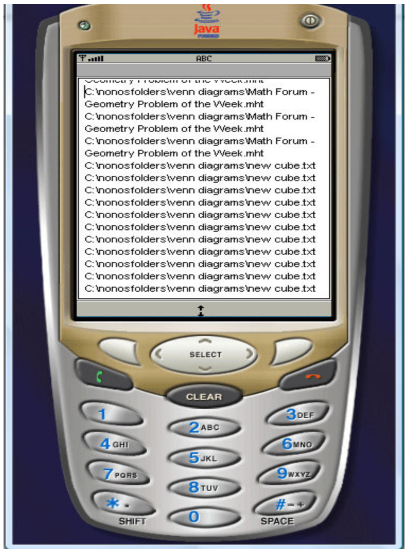
Snapshots of a few deployed applications are given below



type below
type :
An

list blow
list of contacts
anshul.t ansh.alm
anandcse ansh.mukesh
ananth

Figure 5. List of contacts starting with “an” in First name

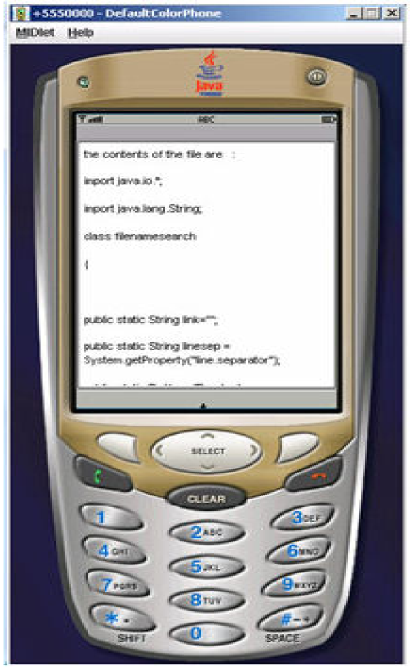


```
C:\nonosfolders\venn diagrams\Math Forum -
Geometry Problem of the Week.mht
C:\nonosfolders\venn diagrams\Math Forum -
Geometry Problem of the Week.mht
C:\nonosfolders\venn diagrams\Math Forum -
Geometry Problem of the Week.mht
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
C:\nonosfolders\venn diagrams\new cube.txt
```

Figure 6. File names containing the word “cube” in its contents

## VII. CONCLUSION

Thus a mobile user can access his personal computer from anywhere using mobile devices even when he is located remotely. Also in mobile handsets which do not have extendible memory slots the user need not over write his existing data to store a new one. Instead he can take a backup of his mobile specific data on a remote desktop machine. As there is no facility to perform a search inside the mobile device, the intra-mobile search application makes the user spend less time in searching for data.



```
the contents of the file are :

inport java.io.*;

inport java.lang.String;

class filenamesearch

{



public static String link="";

public static String linesep =
System.getProperty("line.separator");
```

Figure 7. File “filenamesearch.java” downloaded in the mobile

## VIII.REFERENCES

[1] S. Kulkami , Miss S. Diwan and N.K. Bansode , “Device Independent Mobile Application Controller for Remote Administration of a Server over a GPRS Link using a J2ME Cellular.” IEEE Lndia Annual Conference (INDICON) 2004 IIT- Kharagpur- 721302, December 20-22,2004

[2] Vincenzo Auletta, Carlo Blundo, Emiliano De Cristofaro, "A J2ME transparent middleware to support HTTP connections over Bluetooth," pp.3, Second International Conference on Systems and Networks Communications (ICSNC 2007), 2007

[3] Haklin Kimm, Sung Y. Shin, and Charlie Shim, “Two Approaches to Improve Java MIDP Record Management System in Wireless Devices," Proceedings of 2005 IEEE International Conference on Electro Information Technology, Lincoln, Nebraska, May 2005.

[4]Chang-Tien Lu, Manu Shukla, Siri H. Subramanya, Yamin Wu. Performance Evaluation of Desktop Search Engines. IRI'2007. pp.110-115.