Design and Development of Application Software Systems for the Acquisition and Storage of Local and Remote Data for Mobile Applications using TCP/IP protocols and GPS/GPRS services.

Mauricio Daniel Gomez Torres, Jose Antonio Miranda Solis, Jesus Gutierrez Gomez
Research Center for Automotive Mechatronics
ITESM Campus Toluca, Mexico

Eduardo Monroy Cardenas no. 2000. Zip50110
Toluca, Edo. de Mexico, Mexico
Tel. 01722-2793118
Fax: 01722-2741178
Mauricio <A00743525@itesm.mx>
Jose Antonio <A00927650@itesm.mx>
Jesus Gutierrez <jesus.gutierrez@itesaz.mx>

Keywords
Internet, TCP, UDP, Databases, Mobile technology.

Abstract
The objective of this work is the design and implementation of a computer software application for the acquisition and storage of local and remote data for mobile applications using TCP/IP and GPS/GPRS services.

It is very important for enterprises to manage and control mobile technology systems in production and assembly lines. Universities have the need to maximize the learning experience of the students by bringing theory and practice to the classroom and the laboratory using a remote electronic or mechanical device as a CNC machine.

In both cases when a vehicle or a product is in movement and/or being assembled in a production or assembly line, variables as temperature, acceleration, deformation and position have to be sensed, collected, transmitted and stored in a remote database and visualized at a control central office in a company or a classroom/laboratory in educational institutions, on a PDA device or in a computer system.

In this work, electronic wireless sensors are used to obtain the data from the mobile element. The acquired data is processed using an Internet UDP server, which transmits the data using GPRS service from the GSM cellular phone network towards an Internet TCP server in which data is stored in a database. The stored data in the TCP server is then read with a Java Servlet or a PHP program and then visualized in graphic mode using a browser with Flash on a PDA or computer system.

This paper summarizes the design and implementation of each software module to allow the visualization of the remote variables sensed from the mobile element using the Internet TCP/UDP/IP communication protocols and GPS and GPRS services.

1. Introduction
Technological advances in factory automation, production and assembly lines, have established challenges to the design and engineering departments of companies and universities, as there is an increased need for controlling and monitoring processes and test anytime and anywhere. Temperature, acceleration, deformation, position and visualization have to be sensed, collected, transmitted and stored in a remote database and visualized at a control central office in a company, at a classroom or laboratory in educational institutions, or even at a PDA device or laptop computer system.

The basic goals for controlling these variables are, analysis of the performance of the assembly line or the element produced, and to modify and get better...
productive processes using statistical information, in graphical or numerical mode, produced by the remote variables sensed.

As a response to these needs, a working prototype system has been designed and developed at the Research Center for Automotive Mechatronics (CIMA) at ITESM Campus Toluca. This system allows the remote control and monitoring of processes independently of geographical position.

The general scheme of this system is shown in Figure 1. Using as an example a vehicle in motion, the car is equipped with wireless sensors to measure temperature, acceleration, and deformation.

![Diagram of the system](image)

**Figure 1.** Prototype system on a virtual laboratory

The data sensed are received by a Reception Base (RB) device with an Ethernet NIC included [7]. The RB device sends the data to a UDP server. The UDP server packs a data packet with the data received and uses an API communication interface to transmit the data to the remote TCP protocol server [1][3]. The transmission of data between the UDP server [1][3] and the TCP server is achieved with a special application program interface (API) designed and developed in Java. The API interface establishes a serial communication with the Télit GM-862 [6] device which uses a cell phone SIM card and allows the communication to the Internet using the GPRS service from GSM cellular phone network. The GPRS service delivers data to the TCP server using the PPP communication serial protocol [1].

Data stored on the TCP server database is now read by mean of a Java Servlet [5] or PHP script [4], which is part of the Graphic Visualization System. The servlet or PHP program processes the data and sends them to the remote office or laboratory equipped with a PDA or computer client with a browser and Flash [8] software.

2. Design and implementation of the system

The application software system was designed using the UML and Object COAD Methodology and implemented with the Java language. The system is formed with four main software modules: the UDP protocol based server, the API communication interface using Cellular Phone Technology [9], the TCP protocol based server, and the Graphic Visualization System. Each one of the software modules is described in the next sections.

3. The UDP protocol based server

As mentioned earlier the instrumented vehicle uses a Reception Base (RB) device whose basic functions are to receive sensed data and transmit them via Internet. A UDP protocol server [1][3] was designed and developed for acquiring the data from the RB device.

At the beginning, a TCP protocol server had been considered for receiving data from the RB device, but an ACK (ACKnowledge) signal is waited for this kind of server, and this ACK signal is not sent by the RB device causing a conflict in the reception of data. Then it was decided to design and implement the server using the UDP protocol. The Internet server under UDP protocol was able to read correctly the RB device data.

The server was developed using the API Sockets [1] using the Internet UDP protocol and implemented in Java language. The activities that server performs are:

2.1.1 Data reception from the Reception Base device

2.1.2 Global position coordinates reception from GPS service, using a Java API interface designed and developed for acquiring data from the GPS service

2.1.3 Packaging data sensed and position coordinates, in a data packet protocol

2.1.4 Transmission of the information to the TCP server using a Java API interface which establishes a serial communication with the GM-862 device for sending the data to the remote TCP server using the GPRS service from GSM cellular phone network.

The datagram socket server is initialized over a port by executing the following code instruction:

```java
socket=newDatagramSocket(puerto);
```

The data reception from the Reception Base device is achieved by next method:

```java
paquete = recibirDatos();
```

This method is defined as DatagramPacket

```java
public DatagramPacket recibirDatos();
```
An array of bytes is initialized to store all the information:
```java
byte data[] = new byte[100];
```
Next is the designed protocol for receiving the data from the RB device:
- **F1**:
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F1 -- 10 bytes
  ```
- **F2**:  
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F2 -- 10 bytes
  ```
- **F3**:  
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F3 -- 10 bytes
  ```
- **F4**:  
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F4 -- 10 bytes
  ```
- **F5**:  
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F5 -- 10 bytes
  ```
- **F6**:  
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F6 -- 10 bytes
  ```
- **F7**:  
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F7 -- 10 bytes
  ```
- **F8**:  
  ```java
  S1 S2 S3 S4 S5 S6 S7 S8 F8 -- 10 bytes
  ```
- **FF**: means end of the message

A datagram package is initialized:
```java
paquete=new DatagramPacket(data, data.length);
```
Finally the data is received from the RB device:
```java
socket.receive(paquete);
```
The data is now packaged and transmitted to the remote TCP server using a Java API interface which establishes a serial communication with the GM-862 device [6] and the GPRS service from GSM cellular phone network [11].

4. The remote TCP protocol based server

A remote TCP server was designed using the API Sockets [1] and developed in Java language. The server receives data delivered from the GPRS service, and stores the data on a data base server implemented in a DBMS (MySQL).

MySQL database server was chosen because of its consistent fast performance (about 0.013 ms per consult), high reliability, ease of use and configure, and great flexibility because MySQL server runs on more than 20 platforms including Linux, Windows, OS/X, HP-UX, AIX, Netware.

Next section explains how the TCP server is initialized.

The TCP server is initialized:
```java
server=new ServerSocket(puerto, 0);
```
The Server accepts one request from a client:
```java
sfd=server.accept();
```
```java
+ sfd.getInetAddress().getAddress();
taDatos.append("\nCliente número "+ noCliente);
```
The server attends the request of the client using the next class:
```java
new serverRequestClient( sfd, noCliente, taDatos);
```
The `serverRequestClient()` class has defined the methods for receiving and sending data:
```java
public void serverRequestClient(Socket s, int noCliente, TextArea datos) {
  salidaDatos = new PrintStream(sfd.getOutputStream());
  entradaDatos = new DataInputStream(sfd.getInputStream());
  salidaDatos.writeBytes(sfd.getInetAddress().getAddress());
  salidaDatos.writeBytes("\nConexión del Host\n");
  entradaDatos.readFully();
  datos.append("\nCliente número "+ noCliente);
  datos.append("\nConexión del Host\n");
}
```
The data received is now stored in the database for being processed by the graphical visualization system.

5. The graphical visualization system

The design of the visualization system considers five main aspects: System design methodology, HTTP server, Data reading from the database and Graphical Visualization module. This section explains each one of the modules.

5.1. System design methodology

The COAD methodology [2] was used to design the visualization system. This methodology defines four types of classes and/or objects for designing an application. The system is implemented with two type of COAD's objects: 1) the Data Administration Object and 2) the Graphic User Interface Object. Figure 2 shows the object's diagram of the system.
5.2. HTTP server

When a remote client with a browser requires the sensed data visualization, it is attended by an Http Apache server, which calls a Java Servlet [5] or a PHP script [4]. Any of these programs access the database, process the data and generate a graphic that is sent to the apache server and the server sends the reply to the client, which visualize the data as shown in Figure 3.

5.3. Database data reading module

A PHP program is responsible for reading the data from the database (MySQL) [4]. First, the PHP program establishes a connection with the database, by executing the following code instructions:

```php
$conexion=mysql_connect($host, $user,$pass);
mysql_select_db($dbase,$conexion);
```

Once the program has established the connection with the database, it receives a low limit and a high limit. Both limits indicate the number of registers in the database that it must read for executing the data consult. Such limits are provided by the graphical visualization module (a Flash program). The data consult is accomplished by executing the following code instructions:

```php
$consult = "SELECT * FROM sensors LIMIT $lim1, $limh";
$result = mysql_query($consult);
```

After the data consult is done, the PHP program save the retrieved data into local variables created sequentially at the moment of the values assign.

Finally, the PHP program sends all data storage in the $num variable to the Graphical visualization Module (a Flash program) by using the echo function. This task is accomplished by executing the following code instructions:

```php
$i=0;
While($val=mysql_fetch_array($res))
{
    echo "$num$i = $val[valu1]";
    $i++;
}
```

The data reading module was also developed by using Java Servlet technology.

5.4. Graphical visualization module

The graphical visualization module was implemented using Flash as graphical software. This module is responsible for creating a Chart from the obtained data by the Reading module described earlier.

Firstable, the Flash program sends a low limit and a high limit to the Reading module by using the LoadVars() function; both limits indicate the number of registers in the database to be read and perform data consult.

Once the data consult is done, the Flash program receives all the read data from the Reading module by using the LoadVars() function, then it obtains verification of successful data loading. If result of verifications is true, it saves all the data into an array with the help of the eval() function and a temporal variable, else the Flash program will wait for the data result.

After the verification was successful and all the data were saved into the array, the Flash program calls the create_graphic() function, and finally increment in one the value of vliml and vlinh in order to send them to the Reading module in the next data consult. These
tasks are accomplished by executing the following instructions:

```javascript
var vliml = 0;
var vlimh = 100;
var a = new LoadVars();
a.liml = vliml;
a.limh = vlimh;
a.sendAndLoad("http://10.25.131.94/LDatos.php", a, "GET");
a.onLoad = function() {
    for (i=0; i<100; i++) {
        tmp = eval("this.uno"+ i);
        _root.arr[i]=tmp;
    }
    i=0;
    create_graphic();
vliml+=1;
vlimh+=1;
};
```

The `create_graphic()` function is executed and takes all the data in the array, then applies three mathematic processes: 1) to differentiate between the positive numbers and the negative ones, 2) to draw the chart with the selected characteristics by the user such as line style and 3) to know the coordinates and the exact position of data on the chart.

The Graphical Visualization System has the capability to draw a Chart from any value in the set of the real numbers in Real Time with a little delay (the chart is refreshed every second). The result chart is shown in Figures 3 and 4.

6. Results

Four software modules were designed and developed for the mobile application software system: the UDP protocol based server, the API communication interface using Cellular Phone Technology [11], the TCP protocol based server, and the Graphic Visualization System. The software modules interacts one each other and work as a true system. The tests performed with a vehicle in movement have widely succeed. The variables sensed have been perfectly seen on a remote PDA or a computer.

The devices used with the APIs developed allow any software to be used in a completely mobile context. However, the use of the cellular network technology diminishes the transfer speed of the data since it is traveling through the cells of the cellular network and routed through internet to reach the destination server, but the delay is just for a few seconds.

The developed applications are effective, low-cost solution, which can be implemented in almost every working environment. Because of the architecture used and the placement of the application-sets, it is possible to change the communication media without having to make changes to the applications. Due to the selected development technology the developed applications can be implemented without licensing problems, which makes it ideal for educational and enterprise purposes.

This work can be used and applied in many other areas inside a company for achieving the basic goals for controlling the sensed variables: to analyze the performance of the assembly line or the element produced, and to modify and get better the productive processes using statistical information, in graphic or numerical mode, produced by the remote variables sensed.

The software modules were developed using Java technology. The graphical visualization module was implemented with both, Java Servlets [5] and PHP scripts, and Flash as the graphical software. The final result of the mobile variables sensed are shown in Figure 5.

![Figure 4. Data visualization chart](image-url)
7. Conclusions

The application software system is fully functional but the system can be improved. More tests are needed in order to determine the rates of delays, data loss and some other issues that might cause a malfunction of the application system.

The design of the communications APIs allows their use in several different applications. The use of an API provides several advantages since the programmer of the application that uses it does not have to worry about serial communication or specifics of the Gm862-PCS device commands or the GPS device commands.

This implementation allows all the software developed to be able to communicate almost everywhere, as long as there is a GSM network available.

8. Acknowledgements

This work would not have been possible without the help of the following Computer System Engineering students at Campus Toluca: Garem Gaspar Anaya, Miriam Colin, Eduardo Mendoza, Antonio Garcia, Alberto Perez, and Raymundo Jimenez. Also the guidance and advice of M.Sc. student Omar Jara and professors Gerardo Alducin and Jose Carlos Miranda is gratefully acknowledged. This work was kindly supported by HP through the Mobile Computing Technology Grant 2004.

9. References


10. Acronyms

API Application Program Interface
GPS Global Position System
GPRS General Packet Radio Service
GSM Global System for Mobile Communications
HTTP Hyper Text Transfer Protocol
IP Internet Protocol
PDA Personal Digital Assistant
TCP Transmission Control Protocol
UDP User Datagram Protocol
UML Unified Modeling Language