WEB Model for Remote Data Acquisition Based on Google Maps GIS

Siniša Ranđić[1], Uroš Pešović[2], Željko Jovanović[3], Vlade Maksimović[4]

Abstract – **In this paper, we presented the Web model which can be used for storage and representation of data acquired from remote objects. These objects, which can be either mobile or stationary, send acquired data to Web server using GSM/GPRS networks. Data is stored in database, from where it's displayed to user in various forms. Standard table view is used to presents data history, while live data with position of remote station is shown on geographic map which is based on Google Maps API, global Geographic Informational System (GIS).**

Keywords – **Remote object, data acquisition, Google Map API, Web application.**

I. Introduction

This article is striving to show how is possible to remotely monitor measured values for different locations. Remote stations are used for acquisition of various environmental values, such as metrological, hydrological, air and water pollution and many other applications. For exact representation of some environmental property on observed area, wide network of these stations is necessary. Usually, only acceptable mean of connecting these remote stations into network is by some kind of wireless network, for example cellular GSM network.

Data acquisition system presented in this paper is based on GSM/GPRS modules, which can be used both for localized and distributed data acquisition systems. These kind of modules use cellular GSM networks which are widespread in every country around the world and cheap and easy to use.

Web application of our remote data acquisition system is based on Java platform. For geographic maps, we use Google Maps API. This global geographic informational system Geographical Information System (GIS) offers precise worldwide coverage which is continuously updated.

II. Data acquisition

Data acquisition system based on Telit GM862-GPS module which has GSM/GPRS capabilities. This device besides its GSM/GPRS functions, has built in GPS receiver with high accuracy of acquired position. Module also has one analog input and thirteen digital multipurpose I/O pins. Digital I/O pins beside its primary functions can be use as interface for digital sensors, which are connect to device with SPI or I^{2}C bus. Telit module itself can be use for localized data acquisition where sensors are not too far away from module, as shown in Figure 1.

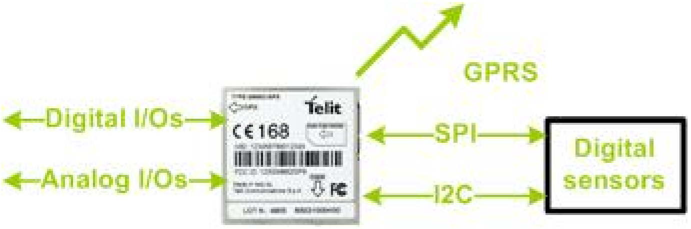


Figure 1. Localized data acquisition system

In case for dislocated sensors, acquisition system must be base on remote I/O modules, which are connect together with some kind of industrial network, such as RS485 or Ethernet. These modules are capable to measure various analog signals such as voltage, current, temperature, etc., acquire, and update values of discrete signals. This configuration of measuring system has many advantages over centralized measuring systems. Analog signals could not travel long distances because voltage-drops on cables and noise which can be easily picked-up in noisy industrial environment. Also for every sensor, separate cable should be used which could be quite costly if the cables are long. Drawback of this configuration is low speed, which is limited by speed of the network which is used for communication between remote I/O.

Module used in our application is ADAM-4017, which is 8-channel A/D converter, with various input ranges for voltage and current signals. It is capable to communicate over RS485 network bus, using ASCII character protocol. Telit module is connected to this RS485 network by its serial port using RS232/RS485 converter module ADAM 4520. Layout of distributed data acquisition system used in our application is shown on Figure 2.

[1] Siniša Ranđić is with Technical Faculty in Čačak at University of Kragujevac, Sv. Save 65, Čačak 32000, Serbia, E-mail: rasin@tfc.kg.ac.rs

[2] Uroš Pešović is with Technical Faculty in Čačak at University of Kragujevac, Sv. Save 65, Čačak 32000, Serbia, E-mail: pesovic@yahoo.com

[3] Željko Jovanović is with Technical Faculty in Čačak at University of Kragujevac, Sv. Save 65, Čačak 32000, Serbia, E-mail: zjovanovici@gmail.com

[4] Vlade Maksimović is with Technical Faculty in Čačak at University of Kragujevac, Sv. Save 65, Čačak 32000, Serbia, E-mail: vlade.maksimovic@gmail.com

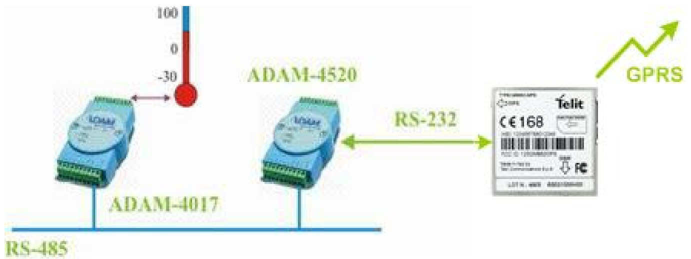


Figure 2. Distributed data acquisition system

Telit module can work in two operating modes, as a microcontroller and as modem. When working as microcontroller is control by program written in, Python script language, which enables full autonomy of the module. If module works as modem, it needs to be control by external microcontroller or computer using AT commands.

Basic principle of both data acquisition systems is based on continuous polling of analog, digital or remote I/Os and sending of acquired data to web server using GPRS protocol. Data transfer between module and server is carried by HTTP protocol, using GET request. One cycle of data acquisition and communication with server is shown in following code listing.

```
Reading ADAM module channels
Response: >+7.2111+7.2567+7.3125+7.1000
+7.4712+7.2555+7.1234+7.5678

Send data to: 91.187.132.16:8080
Socket: CONNECT
GET
CSLGPS/GISRazvojnoOkruzenje/NewPoint?Telit_id=3
&lon=02021.1222E&lat=4352.9047N&parm1=+7.2111&p
arm2=+7.2567&parm3=+7.3125&parm4=+7.1000&Status
=01 HTTP/1.0

Response: HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 0
Date: Fri, 23 Jan 2009 19:11:19 GMT
Connection: close

Response: NO CARRIER
```

III. WEB APPLICATION

As regards of the practical part of this project, its main goal is to represents the data in a way, which was explain, in the previous chapter. Normally, it is possible to pass parameters and values in some other way only if the specified URL http request made. The entire software solution has been realized using the open source J2EE [2] technology. For the realization of this solution, it is possible to use any programming language like PHP, but because of the large number of functions that provides JDK (Java Development Kit) the Java programming language be selected. One of the reasons why the J2EE technology was select is the possibility of extension to mention the system used on mobile phones in a J2ME applications. Completed software solution can be dividing roughly into two parts:

- Part that exchange data with hardware
- Part that presents results to the client

Part of the software used to exchange data with hardware, is the interface with the hardware support realized in described way. This part of the software has no visual interpretation and is execute only when the http request is pass to the Servlet by a device that forwards data. It has been develop as a Java Servlet, which call by hardware is using the HTTP GET request. To the appropriate data Servlet that performs the same processing and saves them in the MySQL database. The server confirm a successful reception of data and it is able to send the parameters which correct the way that device work. Since the Servlet can be accessed with the using HTTP requests from anywhere in the world, this way of communication gives this project one global level. There is ability of usage in different cities, countries and even continents. The preference listed in the realization of communication in addition to the great advantages is also a defect that openness a project to the attacks and simulation of some other persons who would like to emulate values of passed parameters. Therefore, it is very important to pay extra attention to the protection and checking the validity of the device that sends data. One of the first ways to protect validity of data is to assign to the device a unique identification number to be verifying on each received data. In this field, there are more possibilities for the improvement of security using certain cryptographic methods, which would preserve “the privacy” of data.

The J2EE platform is not able to handle all necessary requirements so implementation of the JavaScript language used in combination with AJAX [3] technology was required. The reason for this is that the dynamics of reading separate data from the database is required. Data in the database changes after a certain time, i.e. after each sending data from the hardware. His is the only way that allows tracking the desired settings for the desired station without any user action. There was no need to use some additional framework like hibernate, struts-and others. From the platform of the work, Servlets are use to process requests and JSP (Java Server Pages) for presentation of processed request. Some JSP pages are implementing using the Expression Language, which provides very easy access to the data stored in JavaBeans components.

As the container Apache Tomcat version, 6.0.16 was use. The reason for using Apache Tomcat Applications Server before the other variations of JBoss, Jrun, Glashfish Applications Servers, etc. is a very simple usage with the Eclipse development tool that was use during construction. As a geographical support in the project, the Google Maps API [6] was use. It provides great opportunities thanks to the entire globe coverage with satellite and aero-photo shots of high resolution. In addition, a very well developed mapping system that is in Western Europe and America developed to such an extent that charted the streets with their names in almost every inhabited place. Serbia is now cover only with the main motorways and roads network, but in the near future it is expected a complete GIS coverage. The principle of working with Google Maps API is that a complete GIS system is on Google's server. The user passes the coordinates and parameters for the display, to the corresponding server that replies sending the required graphic content.

The main condition that must be fulfill to be able to work with Google Maps API is that in each Web page (in our case, the JSP page) that should be displayed must be imported script tag in the HTML head tag of the (JSP) page. The key is supplied free of charge by the Google and it is required to have a Gmail account (mail). In order to get key it is needed to enter the URL of the Web server on which Google Maps will be used. Google will generate a unique key and send it to the Google mail account. It is necessary to set the key to the previously described position.

```
<script
        src=http://maps.google.com/maps?
        file=api&v=2.x&key=

        ABQIAAAA6HHnyYiT2EVEKFG_RERksh
        QqcwwaKzD9OpoV4WAgC4lSGJ_ZoxSrievm
        QkroThPpsog3YNQBW33JsQ

        type="text/javascript">
</script>
```

After that, the image with the desired coordinates could be display. In the JavaScript that will start when the page is loading, it is necessary to define the element in the HTML of the page to display the image in it:

```
if (GBrowserIsCompatible ()) (
        var map=new GMap2
        document.getElementById (
        "map_canvas"),
        (size: new GSize (570,380)));
```

Google Maps has three types of the maps:

- Map - a base geographic-information map with drawed roads and street names
- Satellite - a satellite or aero-photo image record of the desired location
- Hybrid - a combination of previous two types

During the presentation of desired location, it is necessary to forward a map type and coordinates of the locations that want to be show to the server. Shown coordinates correspond to the central position of the image.

```
        map.setMapType (G_SATELLITE_MAP);
        map.setCenter (new GLatLng (latitude,
        longitude), 13];
```

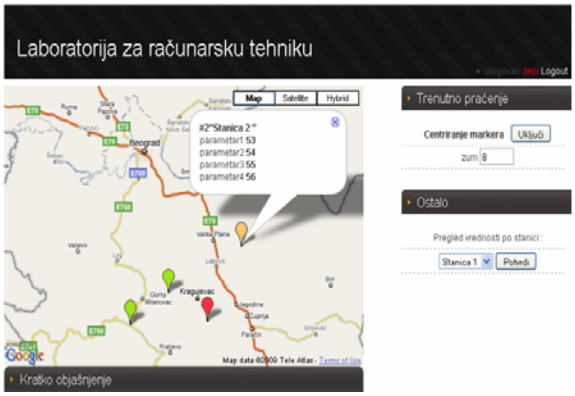
In addition, user has the possibility of post-election of the map type that would be display. Optional it is possible to add zoom control scale that is on the left side of the map. Map clarity depends on the quality of recordings. Some of the more important locations have high-quality aero-photo shots with the possibility of zoom increasing up to 17 times. If necessary, map can be zooming without additional control with double-clicking the left mouse button, and un-zoom with double-clicking the right mouse button.

```
        map.addControl(newGMapTypeControl());
        map.addControl(newGSmallMapControl ());
```

The project was implementing as a Web application [7], which is located on the server in Computer Science Laboratory of Technical Faculty Cacak. The application consists of two parts. One provides currently monitoring of more stations and displays their view in real time with information about latitude and longitude as well as the values of parameters related to each of the station, which is show on the Figure 3. Since this system has a role in monitoring, the value of parameters for easier viewing, marker, which is a station on the map, may be green, yellow or red depending on the values of parameters, i.e. whether they allowed or not allowed critical range, respectively. In addition to transferring data, hardware has a role that does the processing of data, and as a parameter sends its status. Processing is not on the server from the simple reason that for n stations which monitoring m parameters would representing an m * n processing every few seconds. In addition to status of station by the colors of the marker, it is possible to see the value of all parameters by clicking on the marker.

In addition, there is a option to see the value for a specific station during the time interval as shown in the Figure 4. When reviewing the values shown in the table there is a visual assistance in the form of colors that represents status of certain station at a time.

It is important to note that Google Maps allows the execution of up to 500 000 requests during the day in a free license. If a higher number of requests are need for cases to monitor a large number of devices, Google Maps need to contact with the request to increase that number.



Laboratorija za računarsku tehniku

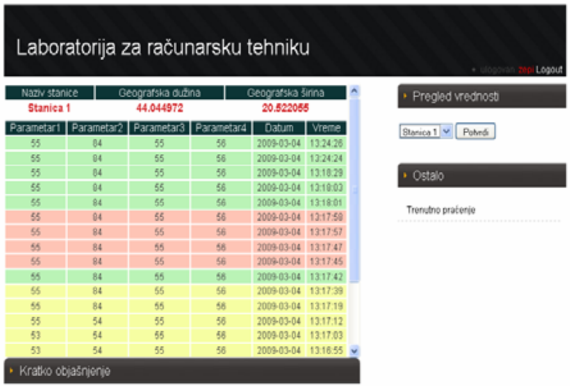
#2"Stanica 2"
parametar1 53
parametar2 54
parametar3 55
parametar4 56

Centriranje markera Uključi
zum 8

Pregled vrednosti po stanici :
Stanica 1 Potvrdi

Figure 3. View the current follow-up station

Complete realization of solutions in the form of a web service provides one more level in the field of globalization. Is possible to develop users applications on any Web platform and the necessary information for presentation are receive from the web services, and they can be use in various ways. Building the web service is also available on the J2EE platform [4] required the use of Apache AXIS framework that is responsible for the creation of WSDL and UDDI file that describes the structure of the web service and its location on which it would be find.



Naziv stanice	Geografska dužina	Geografska širina
Stanica 1	44.044972	20.522055

Parametar1	Parametar2	Parametar3	Parametar4	Datum	Vreme
55	84	55	56	2009-03-04	13:24:26
55	84	55	56	2009-03-04	13:24:24
55	84	55	56	2009-03-04	13:18:29
55	84	55	56	2009-03-04	13:18:03
55	84	55	56	2009-03-04	13:18:01
55	84	55	56	2009-03-04	13:17:58
55	84	55	56	2009-03-04	13:17:57
55	84	55	56	2009-03-04	13:17:47
55	84	55	56	2009-03-04	13:17:45
55	84	55	56	2009-03-04	13:17:42
55	84	55	56	2009-03-04	13:17:39
55	84	55	56	2009-03-04	13:17:19
55	54	55	56	2009-03-04	13:17:12
53	54	55	56	2009-03-04	13:17:03
53	54	55	56	2009-03-04	13:16:55

Figure 3. View values of parameters for the selected channel

When the structure of the service are known client application communicates with the web service using SOAP protocol which is independent of platform and represents the HTTP protocol which send the XML files.

IV. Conclusion

Since the project in the stage of development, we are facing with a lot more work on the selection of appropriate hardware, and the software, primarily to increase the reliability and security of data presentation. The plan is to develop specialized protocols for the exchange of data between devices and servers, in order to increase the rate and reduce the amount of data that is exchanged, which in the case of the HTTP protocol is charge with large overhead. It has enough space for development and implementation of Web service and possibly his connection with some of the companies which goal is to give and use a value of some parameters, of the measuring stations around the world.

References

[1] *GM862-GPS Hardware User Guide*, Telit Communications S.p.A., 1vv0300728 Rev. 8 -2007

[2] *ADAM 4000 series user manual*, Advantech Co., Ltd., Edition 10.7, May 2008

[3] Paul Perrone Venkata S.R. Tom Schwenk, J2EE Developer's Handbook**,** 2003

[4] Kris Hadlock, Ajax for Web Application Developers (Developer's Library) , 2006

[5] Steve Graham, Doug Davis, Building Web Services with Java, Sums publishing, second edition

[6] http://code.google.com/apis/maps

[7] http://csl.tfc.kg.ac.yu:8080/GISRazvojnoOkruzenje/