# Ulyxes automatic deformation monitoring system

Dániel Moka, Szonja Zemkó (Hungary)

*The CLGE Students' Contest 2012 – 2013*

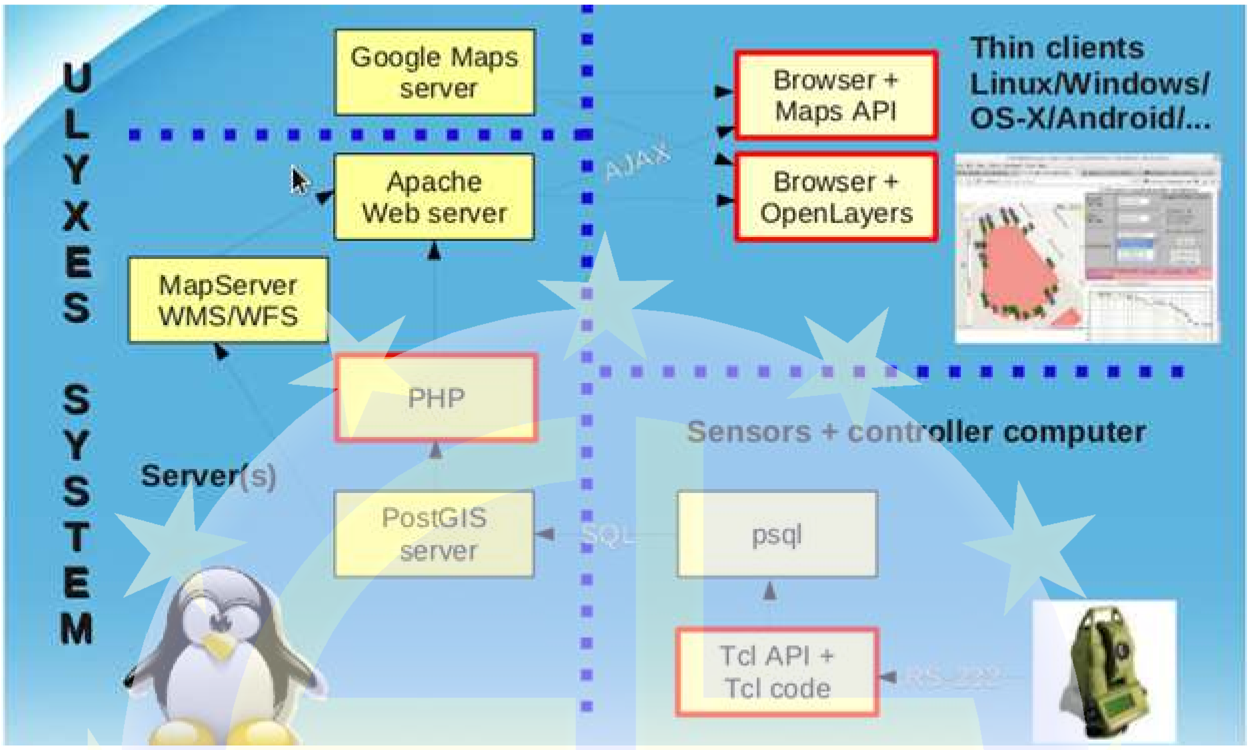
*Category: GIS and Mapping and Cadastre*

# 1. Summary

We live in a dynamic world. Buildings, bridges, dams, and all facilities are in permanent movement, that we must detect in order to preserve the objects from incidental damages. On the other hand, we want to prevent the loss of large amounts of money and last but not least we have to care about people's life who are around the construction. Nowadays, Robotic Total Stations (RTS) can be used to solve this problem. As a matter of fact, a RTS is not enough alone, we must build up an Automatic Deformation Monitoring System (ADMS) using RTSs. Beyond the automation of surveying systems our main goal is to obtain fast data processing and smart visualization. To realize it we exploit the benefits of information technology, thus raw data can be easily transferred to multiple places at once, not to mention the countless products that can help with modeling and analysis. At present the development of informatics makes it possible to create reliable applications that meet users' needs in any field. Many software components and procedures have become available lately, so the real challenge is not the realization but the ability to find the most efficient solution. Usually both the professional and the everyday user prefer the visual and user-friendly solutions, therefore we put an extra effort on these aspects.

# 2. About Ulyxes

Ulyxes is and open source automatic deformation monitoring system to drive robotic total stations and other sensors and publish observation results on web based maps (http://www.agt.bme.hu/ulyxes/index.html). The project is created by Dr. Zoltán Siki at the Budapest University of Technology and Economics Department of Geodesy and Surveying. Its primary aim is to create a framework with which we are able to control different types of RTSs and to publish measured data and detected information via the Internet. Several open source applications contribute to the development of the system, which are key elements of Ulyxes. The greatest advantage of our system is its inexpensiveness thanks to the exclusive use of free services and open source software. The Ulyxes project consists of three basic parts: Instrument programming interface, Web publishing interface, Servers.



2-1 The Ulyxes System Design

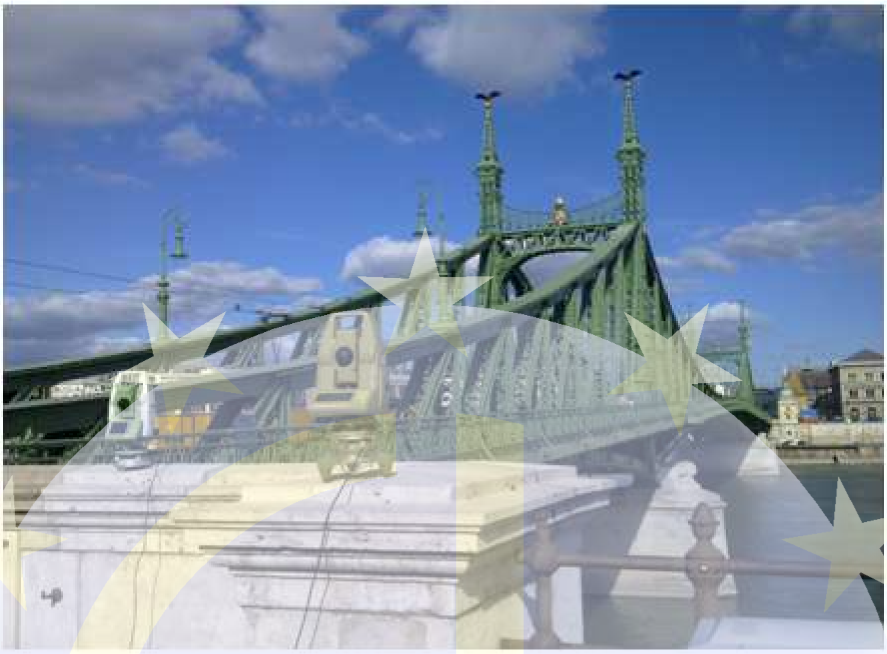
# 3. Instrument programming interface

Theodolites used by surveyors have been developed for more than half a century. The first innovative electronic instruments appeared in the '70s, which replaced the traditional optical instruments. There was a time when the theodolite and the telemeter were independent devices, and the data was recorded manually. A new product called Total Station has appeared on the market in the '80s. These instruments already had an Electronic Distance Meter (EDM), which can serve fast and accurate measurements in order to detect small displacements on objects. Subsequently, in 1990 the Swedes have presented the Robotic Total Station equipped with different features such as Automatic Target Recognition (ATR) and tracking mode. The development of RTSs has not stopped. Many new products are released each year, providing engineers more features to make their lives easier.

Nowadays, it is safe to say that RTSs are the most versatile products on the land surveying market. These equipments are very useful when surveyors and companies are dealing with small displacements and deformations. Following the development of RTSs, more and more concepts have appeared in the world of surveying, which are closely related to one-person surveying tasks. Thanks to ATR the user do not have to aim the prism manually, since the accurate adjustments are done by the instrument itself. As for the Lock Mode, we are able to determine the exact position of a prism without the interruption of tracking. In addition to these aspects, RTSs have a lot more features, i.e. Images Station, which can contribute to the operation of ADMS.

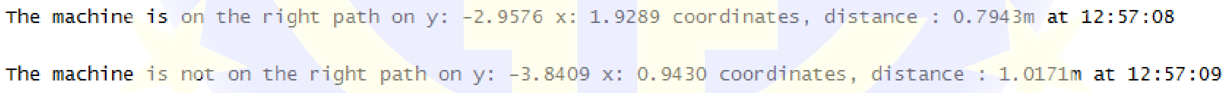
These days there are more complicated structures that are accompanied with complex planning, construction and deformation monitoring. Instead of the traditional manual methods, RTSs can be programmed by engineers to work automatically in order to solve the occurring problems. There are different ways of programming instruments, depending on the device and the aim of the task. In the Ulyxes project we work with computer-controlled measurements. During this method, a prewritten software on the computer controls the operation of the instruments. For greater flexibility and efficiency it is highly recommended to create an application programming interface (API) which can simplify the programmed controlling of devices. In our project we did so and named it TclAPI. The TclAPI is a high-level application programming interface made in Tcl programming language. With its help, users are able to manage different controlling operations and automated measurements with relatively low programming experience. On the other hand, the interface hides the differences between stations so it gives great flexibility to the user. To clearly understand the operation of the TclAPI, we have made a web-based user documentation written in HTML (HypertText Markup Language). This document can be found under the following link: http://www.agt.bme.hu/ulyxes/doc/ and it explains how the API works and describes the usage of functions and routines to perform automated measurements. The documentation is automatically generated from the comments inserted into the TclAPI source code.

The TclAPI has already been successfully tested on different objects within the framework of Dániel Moka's diploma work. Its final chapter covers four practical applications, demonstrating the efficiency of the TclAPI and robotic total stations. The main project was the Deformation Monitoring of Liberty Bridge due to Tram Traffic where the deflections of each side of the bridge were easily determined by RTSs driven by a Tcl code using API.



3-1 Monitoring work at Liberty Bridge

Furthermore the API can be used for monitoring the work of construction equipment with RTSs. A monitoring program has been made so users can observe the path of machines in horizontal direction. The software creates a log file containing the following information:



```
The machine is on the right path on y: -2.9576 x: 1.9289 coordinates, distance : 0.7943m at 12:57:08

The machine is not on the right path on y: -3.8409 x: 0.9430 coordinates, distance : 1.0171m at 12:57:09
```

3-2Two consecutive line in a log file

At present the API is still being expanded and further developed in Python programming language in order to exploit the advantages of object-oriented programming. We are currently rewriting the API in Python and the new one will be called PyAPI. With this development we are enabling many new features, i.e. driving RTSs via Bluetooth.

# 4. Web publishing interface

## 4.1. Design

The problem is a simple geodetic task, we have target points with known 3D coordinates to be marked on a map and pre-processed measurement results to be analyzed by executing queries on them. To do so we chose the implementation of a web-based service using only free software components. Browser-based applications are platform-independent software, they can be run under different operating systems and all we need on the client's side is an up-to-date browser.

In order to create a web-based service we needed to design a client-server architecture. We would like to use a thin client model. Our application requires the following server side components:

- web server
- map server
- database server

The user interface was designed to be made up of a map window, an HTML data entry form and a section for displaying results in a table or plotting them on a graph. First we had to find an adequate mapping tool, a data storage solution and the proper method of publishing our data to be accessible in an interactive manner via a simple web browser.

The greatest advantage of digital maps is that they have a structured data system which means the presence of multiple layers in one file. It is also possible to assign properties (attributes) to the displayed map objects and execute queries so digital maps can be adapted to special problems. Due to the digital technology several processes (cartographic generalization, analysis, etc.) can be automated. Instead of using a custom map we decided to use public map services. Nowadays many web map services are offered on the internet, the most popular among them is Google Maps. To allow developers to integrate Google Maps into their websites, Google launched the Google Maps API.

The use of databases can help to deal with large amount of data more effectively. Besides data storage computer database allows data management and data retrieval. The most common way to describe the structure of data is the relational model which is the basis of the relational database management system. A special, improved version of the latter is the object-relational DBMS. To accomplish our project we chose the best open source database management system available called PostgreSQL which is a general purpose software, although it can be adjusted to special needs and plug-ins can also be added. We used the PostGIS plug-in to store GIS data in the database.

Digital and web-based maps are mainly used for positioning and route planning, however by adding statistical data these applications can help with research and sampling. The visualization of analytical aspects is carried out with different geometric objects, colors, patterns, etc., but having large amount of information can make this type of process cumbersome. Spatial databases offer a more effective way and a greater performance for GIS analysis. PostGIS is an advanced support that spatially enables the PostgreSQL database. Supporting several geometric objects and more than 3000 spatial reference systems, PostGIS allows to store, analyse and query spatial information.

Only logically related data can be stored in the same table, therefore we had to create separate tables for the following objects:

- measurement results (point name, date/time, relative Z coordinate)
- target points (point name, geometry)
- observed building/object polygons (ID, building name, geometry)

The geometry data is handled by PostGIS and stored in well-known binary (WKB) representation, which contains the location, the type (point, line, polygon, etc.) and the spatial reference system of the object in a hexadecimal string form.

The building polygon table is independent of the other two, there is no relationship between their respective attributes. To ensure clarity and unambiguity we had to define a column (or combination of columns) as primary key in each table.

The client-side interface is an HTML page with embedded JavaScript code. This source code is public and entirely available on the client-side. On the contrary, server-side documents can only be accessed and managed by a server-side script, in this case written in PHP. We would like to have a flexible client-server interaction, therefore we used AJAX, an asynchronous request / response model. The PHP script can create a connection to PostgreSQL, moreover, embedded SQL statements can retrieve user requested data. To further improve efficiency, JavaScript receives queried data in JavaScript-ready JSON interchange format.

## 4.2. Realization

Since the web server and its components are not part of the operation system by default, we needed to install them manually. Our choice is the most popular open source web server: Apache. There are special software packages created for easy map server setup, i.e. ms4w, which installs a pre-configured web server environment on Windows platforms, including Apache, PHP, MapServer, several supports and utilities. Testing with Google Maps in the first round, the map server components are not utilized, but saved for further development of the application.

The implementation of the Ulyxes web publishing interface consists of three steps: 1. Database creation. 2. Server-side scripting. 3. Client-side scripting.

1. Database creation

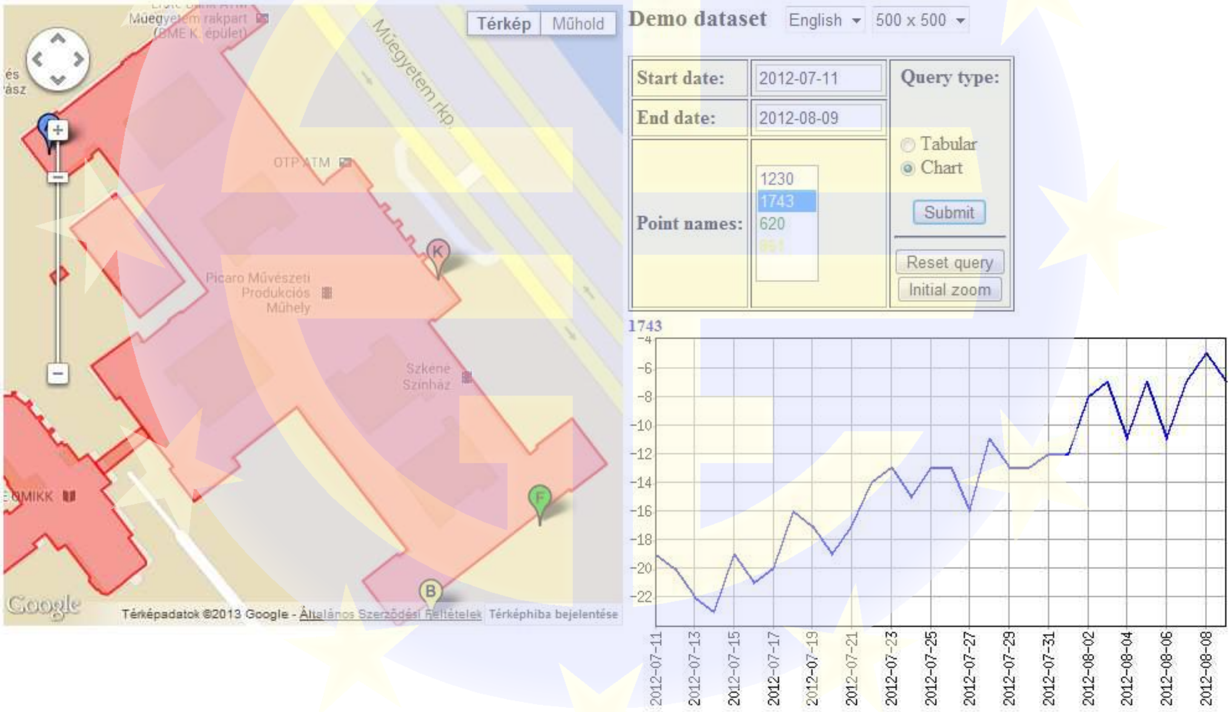
   Building polygons and target points can be copied to the database from text or shape (.shp) files. The geometry of these objects is generated from given coordinates. PostGIS has a special function that converts WKT (well-known text) format to WKB. Shape file contents can be imported to PostgreSQL from a GIS software that has support for the database management system, i.e. QGIS, a popular open source system. During observation measurement results should be loaded to the database in real-time, but we tested the application by preloading existing data sets from text files.
2. Server-side scripting

   PHP has to have its PostgreSQL extensions enabled in order to communicate with the database management system. The essence of our PHP script are those three cases that query and return the required data when they are called. The first two cases obtain geometry data for displaying objects (building polygons and target points) on a map, while the third case can get the desired measurement data. In the latter case the query requires user-set parameters (point name, time interval) that should be entered and submitted in the HTML form. However, all three actions must be called by a JavaScript function and they consist of an SQL statement and a JSON data generator. The map server cannot handle the WKB format, therefore stored data must be converted back to discrete coordinate values.
3. Client-side scripting

   In order to embed Google Maps API into our web page, we needed to include the API script in our source code and initialize the map object (zoom, center, map type). We would like to get the stored geometry and measurement data from the server, therefore the client has to activate the above-mentioned PHP commands on the server's side. Due to asynchronous communication (AJAX) this process can be made flexible. Having three different PHP cases, the HTML document must contain three different AJAX calls respectively. Google Maps API provides marker symbols for points and enables the use of polygons. These objects are not related to user-commands, they appear automatically when the map is loaded.

In addition, we needed to create a form in the user interface that requests query parameters. The form is made up of static elements (radio buttons, options, text fields, buttons) written in HTML and manipulated by JavaScript.

For displaying results, a dynamic table has been created that changes its size according to the number of points and the amount of data. As a graphical solution, we have chosen an open source server-side application called OWT Chart. This program can produce different types of graphs from appropriate inputs and send it back to the client in GIF image format. It was immediately evident that this application was not fitting our needs, because it cannot handle time intervals and the scaling is equal on the horizontal axis, hence data gaps are not represented.



Start date: 2012-07-11
End date: 2012-08-09
Point names: 1230, 1743, 620
Query type: ○ Tabular ● Chart

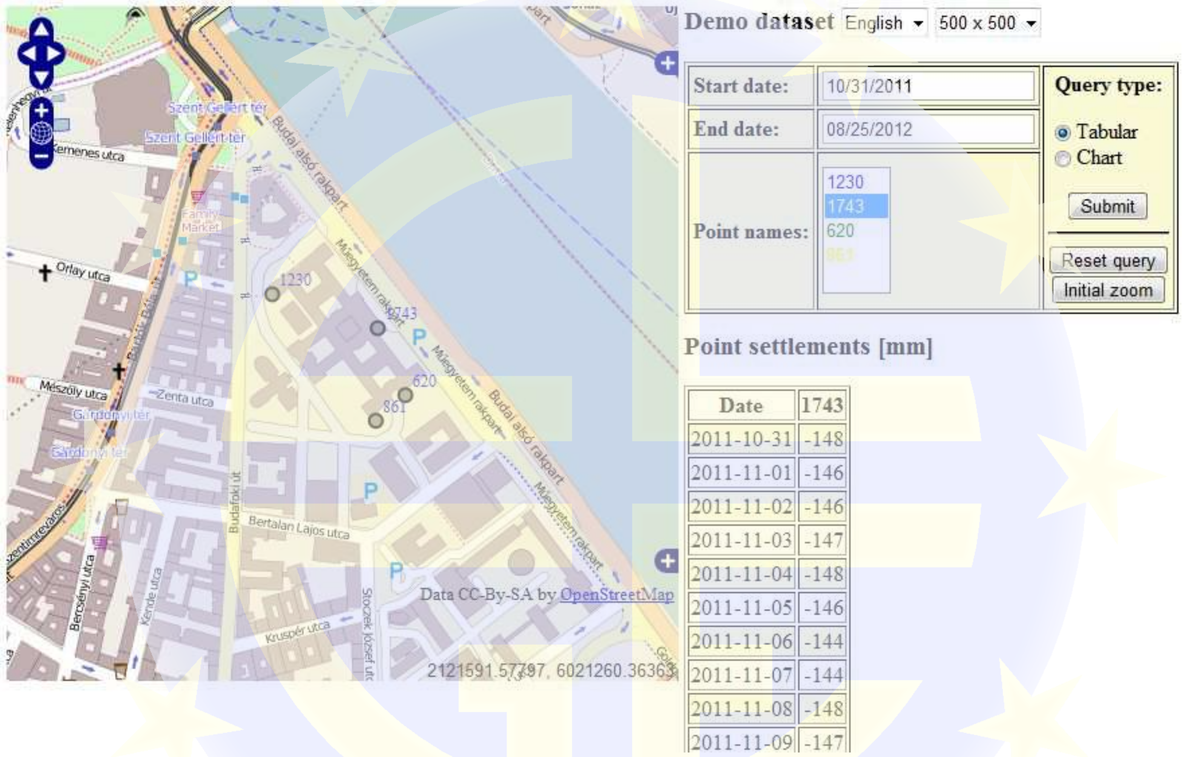
4-1 The web publishing interface Google Maps API version (http://www.agt.bme.hu/maps/ulyxes/)

User-friendliness was an important consideration during development, thus we tried to add more and more functions that enhance the quality of the application. To facilitate time interval setting an interactive JavaScript calendar was joined to the form. Besides mapping the site we would like to share some images of the survey in a tasteful, unobtrusive way. Our choice was a popular JavaScript library for image viewing called Lightbox 2.

## 4.3. Testing the first version and further development

The software fulfils the basic requirements of a deformation monitoring system user interface. However, some alternative add-ons have been tried out to eliminate weaknesses.

Google Maps is free to use according to its terms of service, but not open source. Using own map server would stop the dependence on other companies, and with custom maps or satellite images we can have an up-to-date service of higher resolution. The Ulyxes project has been improved with an OpenLayers & OpenStreetMap map service. OpenLayers offers a wider set of functions than Google Maps and OSM can be freely used in multiple mapping environments.



Demo dataset English ▾ 500 x 500 ▾

| | | |
|---|---|---|
| Start date: | 10/31/2011 | Query type:<br>● Tabular<br>○ Chart<br>Submit<br>Reset query<br>Initial zoom |
| End date: | 08/25/2012 | |
| Point names: | 1230<br>1743<br>620<br>861 | |

Point settlements [mm]

| Date | 1743 |
|---|---|
| 2011-10-31 | -148 |
| 2011-11-01 | -146 |
| 2011-11-02 | -146 |
| 2011-11-03 | -147 |
| 2011-11-04 | -148 |
| 2011-11-05 | -146 |
| 2011-11-06 | -144 |
| 2011-11-07 | -144 |
| 2011-11-08 | -148 |
| 2011-11-09 | -147 |

4-2 The web publishing interface OpenLayers & OSM version (http://enfo.hu/ulyxes/ol.php)

The client-side script has been rewritten in jQuery so the JavaScript code has become easier to interpret and develop, smaller in size and the support offers many pre-written plug-ins.

We are planning to further improve the Ulyxes user interface. We tested our program with 1D settlement measurement results only, however deformation analysis requires 3D data. In order to obtain a precise display OWT Chart has to be substituted by a more complex data plotting tool. More detailed queries could be executed on the measurement data, which represented special information (deformations larger than a specific value, relative positions of points, etc.). Isoline plotting and mathematical operations (cross correlation calculation, etc.) would allow complex analysis. Last but not least, due to the extreme popularity of smartphones we would like to create the mobile version of the site developed in jQuery Mobile.

# 5. Conclusion

The two parts of the Ulyxes, the TclAPI and the user interface work properly in their current development stage. However, they have not been tried in cooperation yet. The user interface is not ready for real-time data reception, therefore the server-side script should be improved to handle incoming data. Deformation measurement and construction management will be more and more computer-controlled, therefore these type of engineering systems will be used in smaller projects too. The Ulyxes system will be offering a low-cost universal solution for any type of monitoring tasks that could be customized and extended to fit special needs.