

Visualization River Water Level Using Internet Technologies

Rafał Kłoda¹✉, Jan Piwiński¹, Roman Szewczyk¹,
Anna Ostaszewska-Lizewska², and Kamil Duchna²

¹ Industrial Research Institute for Automation and Measurements,
Al. Jerozolimskie 202, 02-486 Warsaw, Poland
rkłoda@piap.pl

² Institute of Metrology and Biomedical Engineering, Warsaw University of Technology,
sw. A. Boboli 8, 02-525 Warsaw, Poland

Abstract. In this paper we present the web-service development for water level visualization in WebGL technology, which was one of the solution proposed to be used for flood management in C2-SENSE project. The paper investigates the various technologies for visualization in different environments (groundwater, surface water, different water flow regimes and climatic zones) and their implementations in real application. Furthermore we focus on particular example taking into account whole cycle of engineering project, starting from domain analysis, survey on similar solutions and applications, project development, solution evaluation, conclusions and chart the further development plan.

Keywords: WebGL · Web services

1 Introduction

The aim of the study is to develop a system design for the visualization of river level measurement results and its implementation using Internet technologies. This system was intended to be used to visualize the results of measurements of the Fortore River in southeastern Italy in the Apulian region (in Puglia) during the first integration phase of the C2-SENSE project. This river is characterized by a fairly large drop, which in case of precipitation turns into a fierce river, causing frequent floods. The C2-SENSE research project includes the development of a reliable communications platform that utilizes current standards and network services to expose command and control (C2) functionality, sensors systems and other early warning systems, thus ensuring their interoperability for disaster management.

One of the goal of this project is to design the pilot application, to verify the actual usefulness in the field of pluviometric emergency management. For this reason it has been defined a pilot scenario that will be implemented in the territory of the Puglia Region in Italy.

The C2-SENSE was built to enable the exchange of data using prominent wired and wireless communication mediums in the emergency domain as well as will provide interoperability with the several different wireless transmission technologies and enhance the connectivity between various networks and devices.

The basic scenario during the pilot will be based on following of the Integration action of different radio networks (e.g. GSM, ZigBee) and acquisition the data from these networks to one local server to send these data directly to C2-Sense system. In this case, SQL server in the central site in Bari, have the connection with C2-Sense system, it is aware about our aforesaid data [1].

As part of the C2-SENSE project, all possible extensions and universal solutions have been foreseen. For this reason, Internet technologies such as HTML5, WebGL, Web Services and necessary adapters have been selected to facilitate later integration.

Figure 1 shows an overview of the early warning system operation.

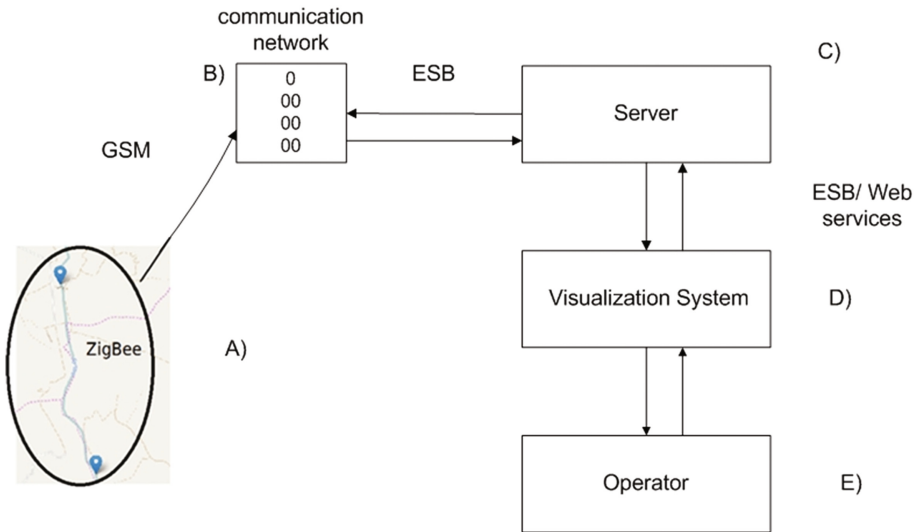


Fig. 1. The early warning system diagram: (a) river level measurement network, (b) communication network, (c) server, (d) visualization system, (e) operator

The basis of the system is a network of sensors that control the level of a river and communicate with each other and the server (Fig. 1a). ZigBee interface was used for communication between measurement modules. It specifies the wireless communication protocols described in [2, 3], therefore interface is characterized by the following features:

- Range - on average up to 100 m (maximum up to 500 m).
- Sufficient battery power is 1 year.
- Simplified transmission protocol.
- Low bit rate (depending on the bandwidth: 20 kbps to 250 kbps).

- Low cost of manufacturing or attaching a communication module to manufactured equipment.
- Maximum number of devices - over 65000.

It is used to create a mesh topology, i.e. where each node is connected to each other network node (full mesh) or different but still large number of internal network connections (partial mesh). This will ensure high reliability and will reduce the impact of single device each node will contain a module for GSM communication with the server. However, it is anticipated that prior to communication with the server, there will be internal network communication to collect results from the entire network. The result report will then be sent to the server from one module. This will reduce power consumption, extend module life and reduce operating costs while maintaining the highest possible reliability.

The server is responsible for archiving and analysing the results and predicting possible hazards and warning in situations of health and life threat (Fig. 1c). It is also planned to develop an expert system with a decision-making system for operators. The server also serves as a central place to coordinate all actions taken to eliminate or reduce the damaging effects of river activity, such as flooding and local flooding.

Due to the fact that man is assimilating and analysing most visual stimuli, an attempt has been made to develop a coherent interface for visualizing the results and positioning of the entire sensor network. Thanks to the use of Internet technology, after the necessary security and authorization, it will be able to display the measurement results anywhere with Internet access. In order to display and retrieve collected current and past measurement results, the visualization system will communicate through the Internet services (Fig. 1d, e)

2 Project Assumptions

2.1 System Technical Requirements

Due to the fact that the application will operate under a larger project, a certain framework for communication between the various elements of the system has been imposed. The exchange of data from the archiving server with the visualization server is done through web services using HTTP and XML technology. There are no imposed requirements that take into account the hardware and operating systems used, so their existence will only be a derivative and result of the chosen technology implementation. The project assumptions are presented below, broken into client and server. The client component is a visualization system that is designed to be easy-to-use and easily available on most modern electronic devices, such as PCs, tablets, smartphones, etc. In this situation, one of two solutions can be used to share data with the device. The first is the creation of a single server with a fixed communication protocol, but with different end-user implementations depending on the platform. This allows for the best matching to the capabilities and ergonomics of the device, but requires the creation of separate programs for each architecture. The second approach is to use web services, so that it is possible to get as close to the functionality and only create two implementation (desktop and mobile

version). The whole interpretation is controlled by the web browser. Due to standardization most of the behaviour is identical to popular browsers such as Firefox, Chrome, Opera, Safari, Internet Explorer/Windows Edge. Because of greater versatility, the second approach has been chosen, i.e. HTTP and WWW based communication protocols and with browser support. It has also been found that the most innovative will be achieved using HTML5 and WebGL technology. Other possibilities were Flash technology and server-side image generation. Due to low mobility, Flash has been discarded. It is not supported on Apple mobile devices (iPhone, iPad) and Android. On the other hand, generating images and transmitting them from the server side creates many server-side performance issues (when handling multiple requests over a small period of time) and the speed of transmission between the client and the server.

The server part is responsible for communicating with the main server storing and analysing data. Information will be retrieved through Web Services, and the server of the visualization system will send requests for data. It was decided to write a server component in Java using Tomcat as an application container. The additional software used will be the HTTP server - Apache. This kit was chosen for the sake of Java's advantages:

- Portability - no application server dependency on the installed operating system.
- Ease and speed of writing code - There are many integrated environments for writing code (e.g. Eclipse, NetBeans, IntelliJ).
- Popularity - Java has become the dominant programming language in recent years. An additional factor influencing the choice of language was the professional practice of the author of the work in the aforementioned technologies.

The input data of the program will be data downloaded from the server. The basic input is a tuple: sensor name (identifier), location (latitude and longitude), measurement result (depth), and date of measurement. The output of the program will be a map with a detailed river bed along with marked measurement results. In conclusion, the following technologies and solutions have been proposed:

- Client side support via browser.
- Internet technologies: HTML5, WebGL.
- Implementing server parts in Java using Apache and Tomcat.
- Communication between servers via Web Services.

3 Implementation of an Application

3.1 Selected Technologies

The main premise for the choice of technology for systemic implantation was to base it on ready-made, technologically stable solutions. This applies to both the selected programming language and the integrated development environment (IDE). Below is a description of the tools used to perform the job:

- Programming Language: Java - is an object-oriented programming language. Source programs are compiled into bytecode, that is, the form performed by the virtual

machine. It fulfils several paradigms: object-oriented, structural, and imperative. It was chosen for its portability, its author's experience in it, great integration with web services, and a number of documentation supporting work. Java is available under the GNU General Public License.

- **Developer Platform: Vaadin** - is a Java application framework that offers a range of Rich Internet Application (RIA) applications. A dynamic one-screen interface is envisioned. Using Vaadin eliminates the hassle of standard HTML technology. It offers server-side architecture and automatically supports AJAX technology. The code is written once, and at compilation it is separated into server and client (Widgets). Free software license: Apache License 2.0.
- **Mapping library: LeafletJS** - was selected as the most transparent, with an intuitive map library support. In addition, it has provided support in the form of an extension in the Vaadin platform via the official add-on. It uses HTML5 and CSS3 technology. The first edition took place in 2011. Free Software License: BSD-2-Clause.
- **3D visualization library: Cesium JS** - one of two found libraries supporting WebGL technology. No official stable addition to the Vaadin platform. Library support will be provided through the JavaScript API. License to Apache License 2.0.
- **Application container: Apache Tomcat** - has been selected as an application/server container to run Web applications in Java Servlets and Java Server Pages (JSP) technologies. It is available under a license: Apache License 2.0.
- **IDE: Eclipse** - Eclipse was written for writing applications. It provides a comfortable and stable development environment. In addition, it has integration with Maven - a tool that automates compiling and building software on the Java platform. License to Apache License 2.0.

The developed system will use directly or indirectly through libraries the following programming languages and technologies: HTML5, JavaScript, CSS3, SCSS, Java, AJAX, Web Sockets. In addition, if the program was made available on the Internet, the HTTP server should also be used. The proposed option is Apache HTTP Server. Available under license: Apache License 2.0. The selection of the development environment included both the convenience of software development, its stability, development opportunities and licensing issues.

3.2 Application Description

The application was implemented using the technologies indicated in Sect. 3.1.

A dynamic, one-screen graphical user interface (GUI) has been implemented. Intuitive map control and a clear menu are used. In relation to the original concept, the main menu was moved from the right to the left. This greatly increased the ergonomics of working with the program. The top menu was also dropped. Options moved to one menu on the right. This expanded the map area and raised aesthetic qualities. Figure 2 shows a cutting from the main program window.

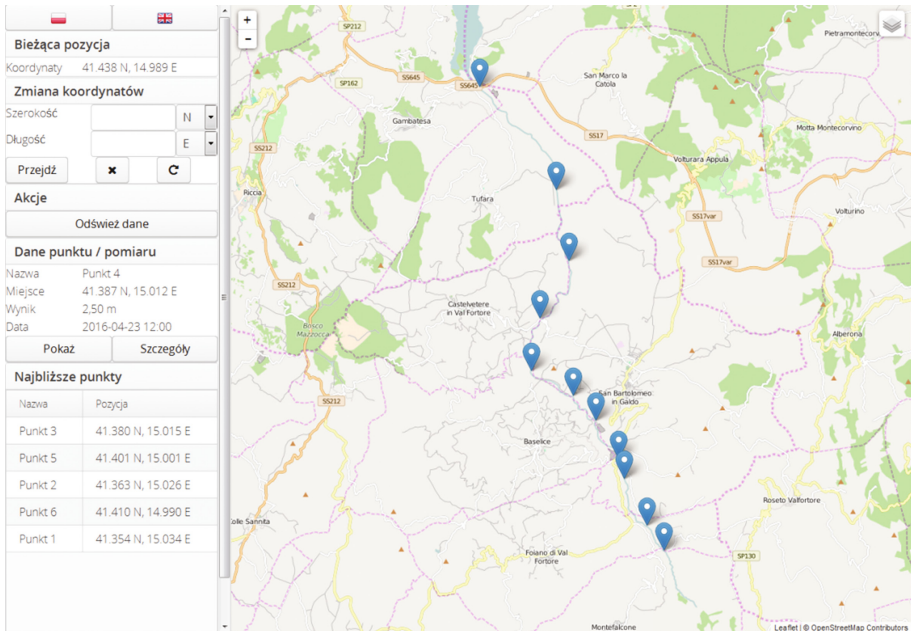


Fig. 2. Cutting from the main window of the implemented application

The program occupies the entire available surface of the page both in width and height. In the central part there is a map. The left side occupies the menu. The breadth of menu widths to maps is 1 to 3. This provides the ergonomics possible with the map and provides clear measurement information. Information coupling and actions performed on the map menu are guaranteed. For example, clicking on the selected element of the list of neighbours moves the map view to a given point. The following functionalities are displayed in the menu:

- Change of application language (Polish and English).
- Current position.
- Change of coordinates.
- Actions (refresh data).
- Measurement point data.
- List of neighbours.

The current position displays the coordinates of the center of the map window. GPS data displayed are rounded to 3 decimal places, therefore this allows to find ones position in case of too quick movement on the map. Changing coordinates is used to change the position on the map if coordinates are known. They are set separately for longitude and latitude. The default settings are aligned to the C2-SENSE project, that is, the location of the Fortore River. After entering the desired values, click the “Go” button. There are also 2 other buttons with “X” and a rounded arrow icon. The first action cleans to force the program to download the latest data and measurement data displays current results and information for a given location. It is possible to change a point by clicking on the

selected marker on the map or by selecting a point from the neighbourhood list. Displayed data is: name, location and last measurement data: score and performance. There are also two other actions available: show and details. The first is to go on the map to the point. It does not change the zoom level, only the camera setting. The second action enable to open the details on the window, as is showed in the Fig. 3, below. The nearest panel shows list of five closest neighbours, as showed in Fig. 2 in left corner. These distances are measured in a vector manner between the points. This list allows to quickly move to the next measurement points, by choosing a neighbour by right-clicking on a row in the table and it causes the map to move to the selected point and to display its basic data in the measurement panel.

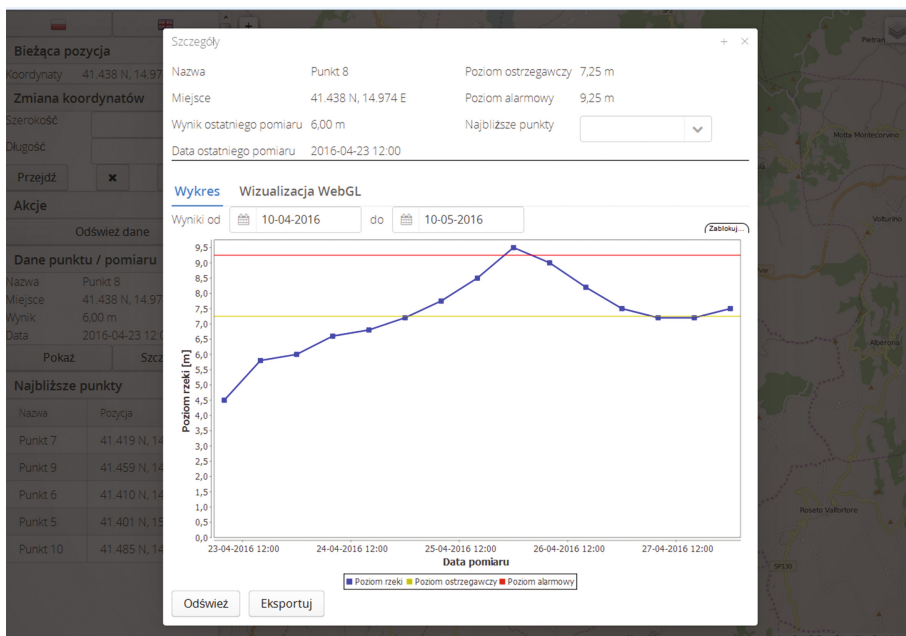


Fig. 3. Detail window of application

The map contains a zoom panel in the top left corner. The mouse wheel may be used to change the zoom level. In the top right corner there is a possibility to change the maps provided. Currently, maps are provided by OpenStreetMap (OSM), Esri WorldImagery, Mtb Map, and OpenTopo. However, there is an easy possibility to introduce others according to ones needs. In the lower right corner there is information on the use of the Leaflet and the map provider in someone's project. On the map there are markers indicating the position of the measuring point. Operator can click on them, to move the map position so that the point is in the middle. In addition, a balloon with point information (name, result and place) and the ability to open the details window are displayed.

The details window is invoked by clicking on the details button on the point data panel or by running from the balloon located at the points on the map. The window opens

in the center and in modal mode. This mode greets data outside this window and prevents interaction with them other than by closing the window. Once closed, the application returns to the base state where data and maps are available. The details window is divided into 2 parts: point data and analysis. The presented data includes:

- Name.
- Location (GPS coordinates).
- The result of the last measurement.
- Date of last measurement.
- Value of the warning level.
- Alarm level value.
- Nearest points.

The list of nearest points allows to quickly move to neighbouring measurement points. The analysis section contains tabs with graphs and visualizations made using WebGL technology. The range of presented measurement dates and their results is determined by the date filters. Dates cannot cross, i.e. the start date cannot be longer than the end date. In the event of an intersection, the system automatically sets the dates to equal. Below the chart are presented possible actions to perform. These are refreshing and exporting. Refresh analogy to actions on the map, retrieving the latest data from the server. Export generates a PDF file containing the displayed point and graph data. Charts are executed in the JfreeChart library, export to PDF is done using the iText library, and visualization in the Cesium JS library.

4 Conclusions

As part of the implementation work, the core and backbone of the system were developed. It is a good base for developing it depending on further requirements and needs. Here are briefly implemented functionalities:

- Layers: background (map) and measurement results.
- Intuitive ability to move around the map.
- Clear user interface.
- Ability to quickly move to measurement point (menu list).
- Import measurement results.
- Generating graphs of measurement results due to user-specified time interval.
- Multilingualism (Polish and English).
- Basic 3D visualization using the Cesium JS library in the tab detail.

Thanks to technology used in our developed application, navigating the map is intuitive and ergonomic. Furthermore, it allows Operator to add more visual layers, for example the current location of the fire department. Common idea about described application, in this paper, is based on open architecture, which consists universal adapters, structural design pattern, therefore it enables the connection to any sensor data source. Overall, this solution has the primary goal to design a cooperation platform that promotes spatial data interchange involving the end user, and provides support for the

assessment and decision on the territory to determine how to deal with in operating mode and systematic different types of events from the knowledge of the location, by the compliance Geographic, from environmental data and historical information.

This goal represents an improvement from a basic approach focused on damage management, to a culture of prevention and prediction, spread out at various levels, based on the identification of risk conditions and the adoption of measures aimed at minimizing the impact of events.

References

1. Bączyk, A., Piwiński, J., Kłoda, R., Grygoruk, M.: Survey on river water level measuring technologies: case study for flood management purposes of the C2-SENSE project. *Advances in Intelligent Systems and Computing*, vol. 543, pp. 610–623, Springer (2017)
2. Franceschini, F., Galetto, M., Maisano, D., Mastrogiacomo, L., Pralio, B.: *Distributed Large-Scale Dimensional Metrology*. Springer, London (2011)
3. Nawrocki, W.: *Measurement Systems and Sensors*. Artech House, Norwood (2005)

Recent Trends in Control and Sensor Systems in
Emergency Management

Szewczyk, R.; Havlik, D. (Eds.)

2018, VIII, 103 p. 50 illus., Softcover

ISBN: 978-3-319-70451-7