An Interactive Website for the River Eurajoki

Ari Jolma, Anne-Mari Ventelä, Marjo Tarvainen, and Teija Kirkkala

Pyhäjärvi Institute, Eura, Finland
{ari.jolma,anne-mari.ventela,marjo.tarvainen,
teija.kirkkala}@pji.fi

Abstract. The project "Our Common River Eurajoki" aims for a long-term sustainable and collaborative platform for improving and managing the river water quality. We describe a website that supports the project by linking social media to environmental monitoring and web mapping. The website development was based on five-tiered software architecture and on several systems working together. Coding was minimized by re-using existing software components. We discuss the workflows supported by the website and the role of information system standards in environmental monitoring.

Keywords: Environmental management, Web, Web services, Geospatial, Sensor observation.

1 Introduction

River Eurajoki in South West Finland has for decades been heavily impacted by human activities. For example the waste water treatment plant (WWTP) of the Eura municipality and two companies discharged raw sewage into the river for a long time due to insufficient capacity of the plant. The plant management kept the bypass secret for many years. Also, for example in May 2011 a paper mill leaked 50 kg of 2,2'-diallyl-4,4'-sulfonyldifenol (a chemical used in coating paper and on the Nordic Council of Ministers' list of chemicals that are dangerous for the environment) into its sewer and most of that probably went trough the WWTP into the river. Similar and even worse leaks have happened before. Microbiological studies made this year (2014) from the river water have often revealed concentrations of enterococci and E. coli that are over the limits for water used, e.g., for vegetable irrigation. The use of the river water was forbidden for swimming and irrigation from September 2012 to the summer of 2013 and again in spring 2014. The poor condition has led to citizen activism and much concern among municipalities and environmental authorities [1].

"Our Common River Eurajoki"[1] is a joint project carried out by the Pyhäjärvi Institute, University of Turku, and JVP-Eura (the company running the WWTP in Eura municipality). It is coordinated by the Pyhäjärvi Institute and runs from March 2013 to April 2015. The project aims for a long-term sustainable and collaborative platform

[1] http://www.pyhajarvi-instituutti.fi/english/image/pdf-tiedostot/yhtejoki_leaflet_brief.pdf

R. Denzer et al. (Eds.): ISESS 2015, IFIP AICT 448, pp. 81–90, 2015.

for improving and managing the river and its catchment by linking citizen activism and environmental monitoring to water quality planning and management. The goal is to increase awareness, facts, and knowledge about the environmental situation while at the same time acknowledging the existing data sources and social media. The work plan of the project comprises development of a website, real-time water quality monitoring, and a sociological study about the relationship between the river and the public. The website is planned to support and link existing activities and to provide access to existing and new data about the river. In this paper we focus on the website the project is developing. The website is online as eurajoki.info.

Using the web for collaboration rather than just as a media for publishing content is a change that has happened in the last ten years [2]. The term "Web 2.0" has been used to describe the development. According to Wikipedia, a Web 2.0 site may allow users to interact and collaborate with each other in social media dialogue as creators of user-generated content in a virtual community. Collaborative websites can be used for environmental management. Geospatial information and mapping is often a core element of the reported attempts. For example [3] discusses large-scale citizen-generated content in the web and its role in environmental monitoring and crisis management. A collaborative environmental decision support system that uses standards based web services is demonstrated in [4].

2 Materials and Methods

2.1 River Eurajoki and its Observation

Eurajoki is a 52.9 km long river in Southwestern Finland with a catchment area of 1336 km^2. It originates at Lake Säkylän Pyhäjärvi, whose ecological status is good, but there are several point sources polluting the river, and many smaller streams with possibly poorer water quality enter the main branch. Also non-point source pollution from agriculture, forestry, peat production, and rural areas may be significant at times. The average discharge of the river at its mouth is 9.6 m^3/s. Eurajoki flows into a bay in the Bothnian Sea, which is a part of the Baltic Sea.

There are 18 observation and measurement sites along the river and in the database that have been used for hydrological (water quantity and quality) measurements. Some sites have been used for continuous measurements; some have been sampled at regular or irregular intervals, some both. Observations and measurements have been done and/or commissioned by private organizations (Pyhäjärvi Institute is one of them) or companies, and by municipal or governmental authorities. Private citizens also sometimes share observations in social media. The observation/measurement workflow depends on who makes the measurement but it is often rather complex involving more than one party and many types of data cleansing and processing especially in the case of continuous measurements and averaged and derived values.

2.2 Requirements

Requirements engineering is according to [5] the process of discovering the purpose of the software by identifying the stakeholders and their needs, and documenting these for analysis, communication, and implementation.

The eurajoki.info website was initiated with several goals in mind: 1) to create a hub for existing sites and social media, 2) to aid in formulating the future water quality management work, 3) to display the existing and new water quality data, 4) to aid in collecting and displaying cultural and experiential data and stories about the river, 5) to display map datasets of the river and its catchment, and 6) to allow access to existing and new reports about the river and its environmental state. It was understood that a website is only one product for water quality data and that there is a need for improved information management in general. Especially support for collaboration and data exchange among individuals and organizations, and publication of data and results was seen important. This strengthened the interest in modern web technologies and distributed systems that are based in standard protocols.

Web mapping and presentation of water quality data were core requirements. It was seen important to be able to visualize the data and spatial information together. The measurement and observation locations, measured quantities (variables), and data itself and their attributes should be browsable. Annotating data was seen important. Public, after adequate identification, should be allowed to add stories to the map and participate in discussions. “A story” was defined as text and pictures that are attached to a point location.

New requirements appear as experience is obtained. For example the state of a web page should be attainable simply with a single URL. The reason for this is how information is often shared in social media with links.

2.3 Architecture

Five-tiered software architecture, such as the one presented by [6], was selected as the basis for the development. The tiers are visualization, presentation, business, data access, and data.

We define the visualization tier as the website as presented by the web browser. A website consists of web pages, which may be static or dynamic. Web pages are written in HTML (or one of its versions), which allows embedding of objects such as images, interactive forms, program code for interactive and dynamic content, etc.

We define the presentation tier as the data formats and the communication between the client (user) application and the servers. For a website http and HTML are naturally the most important communication standards. When it comes to geospatial and observation data, the OGC (Open Geospatial Consortium) offers at least two standards, namely WFS (Web Feature Service) and SOS (Sensor Observation Service) for publishing data [7]. SOS is a part of the SWE (Sensor Web Enablement) framework of OGC; the other parts are standards for observation and measurement data, sensor and observation processing descriptions, and sensor planning service. The OGC discussion

paper 'CUAHSI WaterML' defines another family of web services, called WaterOneFlow, that was developed for communicating hydrologic data. WaterOneFlow was created by the CUAHSI (Consortium of Universities for the Advancement of Hydrologic Science, Inc.) community of US universities.

We define the business tier as the code which implements and enables the workflows that the website supports. The business tier may be implemented on the server side or on the client side. Server side code is executed on demand (when a client requests a web page), possibly taking parameters or other input from the client, and it may use any resources it has available to produce a web page or other content for the client.

The data access tier is the code which connects to the database and prepares data for the presentation or business tier, or stores data coming from those tiers to the database.

The data tier is the database and database management system, which accepts connections to the database, and typically interprets the data access and management commands. The database schema is an important element of the database.

2.4 Software

There are several web mapping and other interactive data visualization technologies for the web. Common ones include Adobe Flash, Microsoft Silverlight, Apache Flex, HTML5, and JavaScript. Each alternative has its pros and cons. We chose JavaScript because it is well supported by web browsers and operating systems, and because of the availability of good free and open source libraries that we could use.

There are several options for developing web pages using JavaScript. These include JavaScript libraries and server side APIs, especially for web mapping, which to use. Our first try was to use the Ext JS library for time series visualization. Ext JS is a large framework for rich interactive and visual web content. GeoExt project develops a library for integrating the mapping library OpenLayers with Ext JS. However, we later found out about the Flot library for time series visualization, which in our opinion supports better time series visualization. For example it can handle missing data, which is common in environmental data sets. Flot is based on jQuery, which was thus selected as the core library instead of Ext JS.

OpenLayers was chosen over other alternatives because of its good functionality, and support for standard and common geospatial data services[2].

The 52°North Sensor Web Community has developed a SOS server, for which various clients exist, including Javascript ones[3]. For WaterOneFlow there is server software but currently no JavaScript client library exists to our knowledge. Despite of these existing solutions it was decided to develop a lightweight ad hoc solution as a

[2] JavaScript web mapping libraries are compared for example on page `http://tinyurl.com/o8k3b4k` (URL made tiny and tested 2014-10-04)

[3] A demonstration website which uses 52°North SOS and a JavaScript client has been developed by British Antarctic Survey (`http://tinyurl.com/my2erpv`). Their JavaScript client uses the Flot library, which was the inspiration for us to use Flot. CSIRO has implemented a Google Maps based client for 52°North SOS (`http://tinyurl.com/nxgae7k`). (URLs made tiny and tested 2014-10-04)

simple initial attempt and as a learning experience. Both the server and the client side of the ad hoc solution are described below.

On the server side we used Perl CGI programs for WFS, for story editor/server, and the ad hoc server. The overlay raster maps were developed into map tiles and set up as tile service. The WFS we used is a part of the Geoinformatica suite and built on top of GDAL and its Perl API.

PostgreSQL relational database management system (RDBMS) with PostGIS extension was used as the main data storage (image and pdf files are stored in file system). phpPgAdmin and Perl CGI programs were used for administrative data management.

3 Results and Discussion

The architecture of the developed system is shown in figure 1. The system is implemented on four subsystems: client web browser, website (eurajoki.info), dedicated services (ajolma.net), and common services (Google map tiles, OpenStreetMap map tiles, and National Land Survey of Finland map tiles). The five tiers of the conceptual architecture are divided between these subsystems. Visualization is defined mostly by the content downloaded from the website and tiled maps downloaded from services. Presentation is based on standards except for time series. Business logic is implemented partly in JavaScript that runs on the client and partly in dedicated services. Data access is based in SQL within dedicated services. Data is in ten tables in the RDBMS.

The web pages of eurajoki.info comprise several types of files. PHP is used to compose the HTML pages and to create certain pieces of JavaScript into the HTML (see below). The website architecture is flat and consists of seven web pages with uniform style. The pages are front page, management plan, water quality mapping, story map, atlas, reports, and about page. Each page has its own Facebook discussion thread. The front page has also the Facebook feed from the citizen activists' Facebook account. The management plan page is currently a stub and it will be developed more at a later stage. The water quality mapping page (figure 2.) has an OpenLayers map panel and a Flot time series panel. The story map and the atlas have a map panel for browsing and editing (story map). The overlays (historical maps and aerial photos) are tiled maps so they load fast and are easy to pan. The reports page contains PDF reports about the river environment. The about page contains information about the site, its developers, data, etc.

The JavaScript code developed for the website is divided into ten files and totals 973 lines[4].

3.1 Water Quality Mapping

The water quality mapping page (figure 2) contains panels for a map, site information, control form, and time series graph. The JavaScript code associated with the page

[4] The code is available at `http://tinyurl.com/leg4yp9` (URL made tiny and tested 2014-10-04)

obtains data from services, and displays it on the panels. The measurement and observation sites are obtained as a WFS layer object. The time series data is obtained from the service as JSON (JavaScript Object Notation) and handed over to the Flot library for visualization. Both map and time series panels have interactive functionality that is mostly managed by the two libraries.

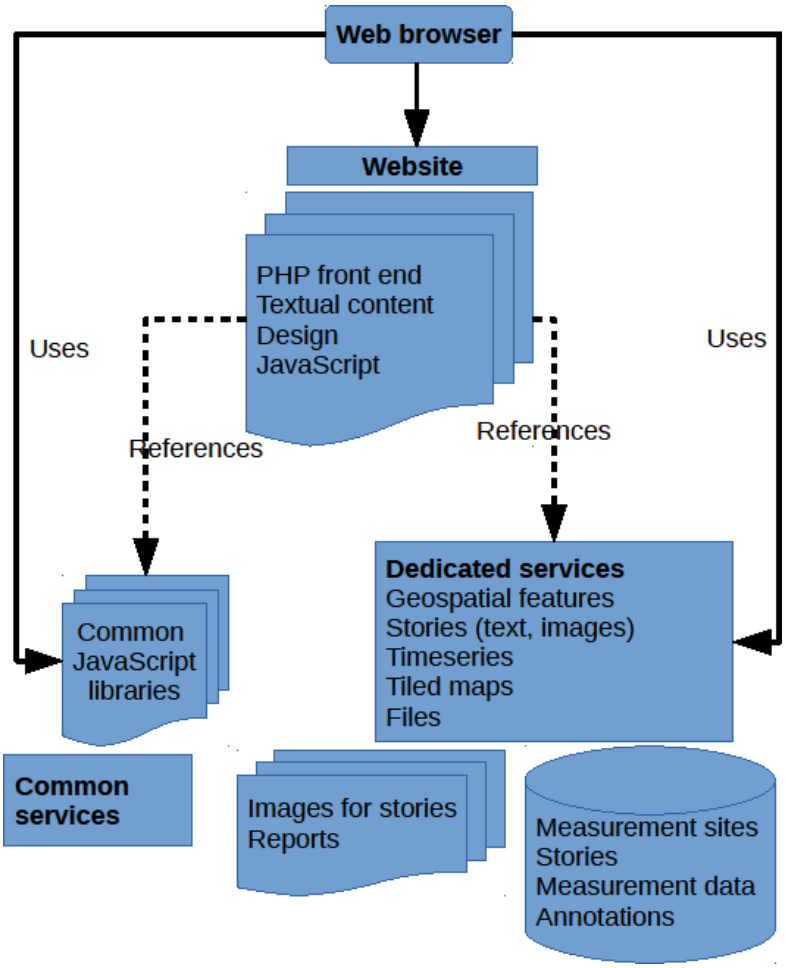

Fig. 1. Architecture of the developed system

The control panel contain standard HTML form widgets for the locations and variables (select multiple list boxes), begin and end dates (text entries with associated jQuery calendar widgets), and a button for updating the time series graph. The locations and variables lists are updated with data obtained from the time series data server (retrieved also as JSON). The locations are linked to WFS features in JavaScript through a key attribute. Thus the selection in the list widget can be synchronized with selected features on the map. The locations are also linked to variables from which

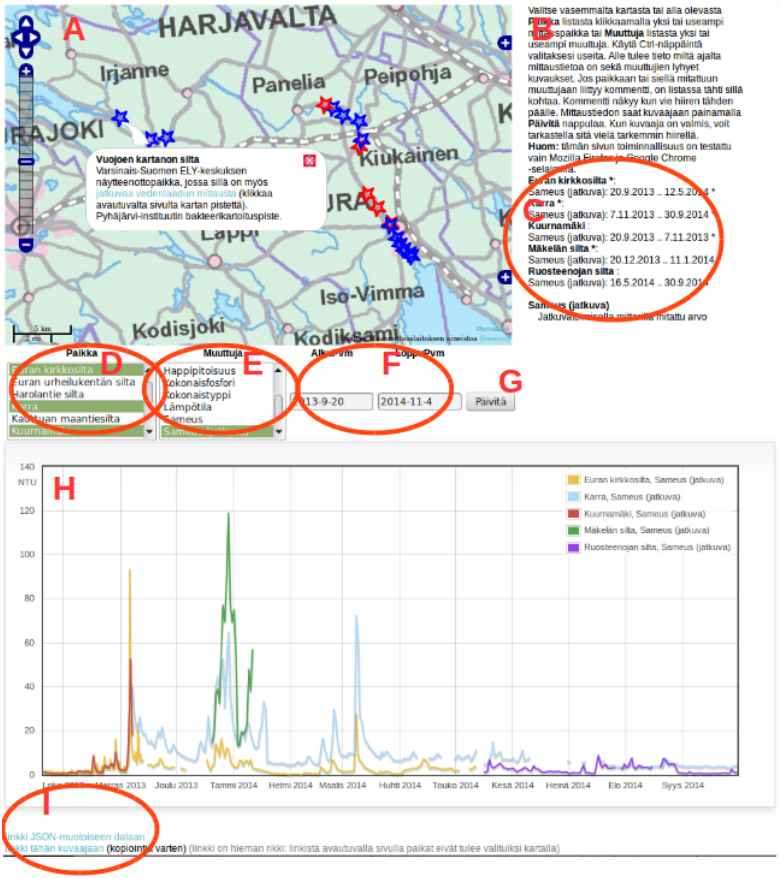

Fig. 2. A screenshot of the water quality mapping page. A is the map panel, five features/locations are selected and one that is not selected has a popup information box. B is general information to the user. C is information about the selected locations. D is the list of locations. E is the list of variables. F is the time period. G is the button to update the graph. H is the time series graph. I is links to raw data and URL of the visualization.

there is data in this location. Thus it is possible to synchronize the selected locations with the selected variables.

A solution to the state-in-a-URL problem (section 2.2), was to use PHP to create JavaScript that sets selected locations, variables and time period in initialization code. The current implementation of the solution is not complete because it does not set the selected locations in the map panel and it does not restore the possible zoom and pan of the time series graph.

3.2 Ad Hoc Observation Service

The requirements for the time series services were the following. The server must be able to list at least the locations, the measured variables for each location, and the

period from which there is data for each measured variable; the server must be able to list the measured variables; and the server must be able to provide time series from requested locations, variables, and period. The service must also provide notes and comments linked to the locations, variables, and data points. The server should return the data as JSON for easy consumption by the client JavaScript code.

The requirements were met with a small Perl CGI program (409 lines of code) that supports three commands: `GetDatasets`, `GetVariables`, and `GetDataset`. The program uses SQL to retrieve metadata and data from the RDBMS and then sends it out as JSON. The observation and measurement database consists mainly of three tables: locations, variables, and data. Complexity is added to the solution by characteristics of data, in particular data being either daily data or data with arbitrary measurement time, and data being either "checked" or "not checked". The former is further complicated by the fact that some data sets may have data points that have only an arbitrary date while some other data points have an arbitrary time stamp (i.e., also hour and minute). The checkedness of data sets is not a simple concept; some datasets can go through several checks and may change more than once due to new calibration data etc. The current solution has four data tables: two for checked data (checked data is shown by default) and two for daily data and two for data with time stamps. The fact that some data sets may have data points in two tables complicates the service code.

3.3 Story Map Editor

The story map was required to have user authentication for editing stories. This requirement was filled by adding an identification dialog before the user enters the story editor. The identification dialog asks for an email address (or name) and a password. These are later used as keys to the data the user has submitted. By default data that a user has submitted is not public. It has to be explicitly set as public by somebody having appropriate rights to the database.

The story editor (its mapping part) is based on transactional WFS (WFS-T). In order to keep things simple on the client side, the stories are always served through the same WFS layer. However, for this to work, the service had to be changed to require the email and password fields and serve features according to those fields: no values: serve public data, values given: serve private data specified by those fields. This feature is not a part of WFS specification.

3.4 Overall Assessment

The result fulfills the baseline requirements. The website uses Facebook social plugins and allows public to create content, however the content has to be published by admins. The solution uses two servers on the Internet, one for the website and one for the dedicated services. The latter has a database and custom CGI programs. The solution we developed in this study did not require extensive development as the program code line counts show.

4 One or Two Standards for Geospatial Features and Observation Data?

Several middleware programs were developed for the services. Some of them are standard, some are standard but extend the standard somehow, some could be standard, and some are specific to this purpose.

According to the CUAHSI WaterML Web Services page[5] WaterOneFlow has eight functions (requests), which are very close to the ad hoc time series server we implemented in this study. The OGC SWE, due to its generic nature, is rather complex and thus it may be difficult to implement considering varying database schemas, and server and client software. Noting this, [8] defined a lightweight SOS profile for facilitating its practical application. The lightweight profile is very similar to the ad hoc profile of this study but it goes beyond ours by including a transactional part for inserting observations and sensors, which in our case can be done but only via separate tools.

The application scenario of [8] is European-wide sharing of environmental data. Our case study focuses on a single river, which is a part of a river basin. The river basin falls mostly within one administrative region, but constitutes only a fraction of it. It is worth considering what types of data services are needed and useful at each level. It could be argued that at higher level the data and information needs about lower level parts would be simpler or less dimensioned (averages, loads at system boundaries, etc.) and when the focus is on a specific system, there data needs are more detailed and also of different kind.

Our system uses two services, WFS and sensor data service, at the same time and in coordination. The coordination is to a large degree ensured by the use of a single database. Although the features (observation and measurement sites) appear in different services and thus as separate instances, they are in reality the same entities in the database. The reason for two services is also practical: WFS layers are well supported by common web mapping clients while SOS is not so well supported.

From a user point of view the technology behind a website has little importance if they do not affect the user experience. However, interoperability becomes very apparent when for example one measurement data stream cannot be added to the same graph as the others.

5 Conclusion

The website has been online since December 2013. The project ends in April 2015 but the website will be kept online after that. Currently the website has a rather steady usage with 150 or more unique visitors per month. Statistics show that water quality mapping and reports are the most popular pages on the site and only a small fraction view the stories or the tiled overlay maps. Only the project team has so far used the story editor. The discussion forum on the front pages has attracted some use but

[5] http://his.cuahsi.org/wofws.html

forums on other pages only little. However, this was expected since the activist Facebook forum (which is shown on the front page) is the main forum for discussions.

Future work will focus on making the provided data and information more meaningful for the public and developing the page that is dedicated to supporting the development of the water quality management plan.

Acknowledgements. The "Our Common River Eurajoki" project is funded by the Regional Development Fund of the European Union (project code A32601).

References

1. Saavalainen, H.: Eurajoki became an open sewer - City of Rauma still using it as a water supply. Helsingin Sanomat (May 25, 2014) (in Finnish)
2. O'Reilly, T.: What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software, http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html (retrieved March 11, 2014) (2005)
3. Boulos, M.N.K., Resch, B., Crowley, D.N., Breslin, J.G., Sohn, G., Burtner, R., Pike, W.A., Jezierski, E., Chuang, K.-Y.S.: Crowdsourcing, citizen sensing and sensor web technologies for public and environmental health surveillance and crisis management: trends, OGC standards and application examples. Int. J. Health Geographics 10, 6 (2011)
4. Sikder, I.U., Gangopadhyay, A., Shampur, N.V.: Interoperability in Web-Based Geospatial Applications. Int. J. Information Technology and Web Engineering (IJITWE) 3(3), 66–88 (2008), doi:10.4018/jitwe.2008070105
5. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. ICSE 2000 Future of Software Engineering [1]. ACM Press. (2000)
6. Tafti, A.P., Janosepah, S., Modiri, N., Noudeh, A.M., Alizadeh, H.: Development of a Framework for Applying ASYCUDA System with N-Tier Application Architecture. In: Zain, J.M., Wan Mohd, W.M.b., El-Qawasmeh, E. (eds.) ICSECS 2011, Part III. CCIS, vol. 181, pp. 533–541. Springer, Heidelberg (2011)
7. Bermudez, L., Bogden, P., Bridger, E., Cook, T., Galvarino, C., Creager, G., Forrest, D., Graybeal, J.: Web Feature Service (WFS) and Sensor Observation Service (SOS) Comparison to Publish Time Series Data. In: CTS 2009. International Symposium on Collaborative Technologies and Systems, pp. 36–43. IEEE (2009)
8. Jirka, S., Bröring, A., Kjeld, P., Maidens, J., Wytzisk, A.: A Lightweight Approach for the Sensor Observation Service to Share Environmental Data across Europe. Transactions in GIS 16(3), 293–312 (2012)