

# Online Internet Monitoring System of Sea Regions

Michał Piotrowski and Henryk Krawczyk

Department of Computer Architecture, Gdańsk University of Technology  
bastian@eti.pg.gda.pl, hkrawk@pg.gda.pl

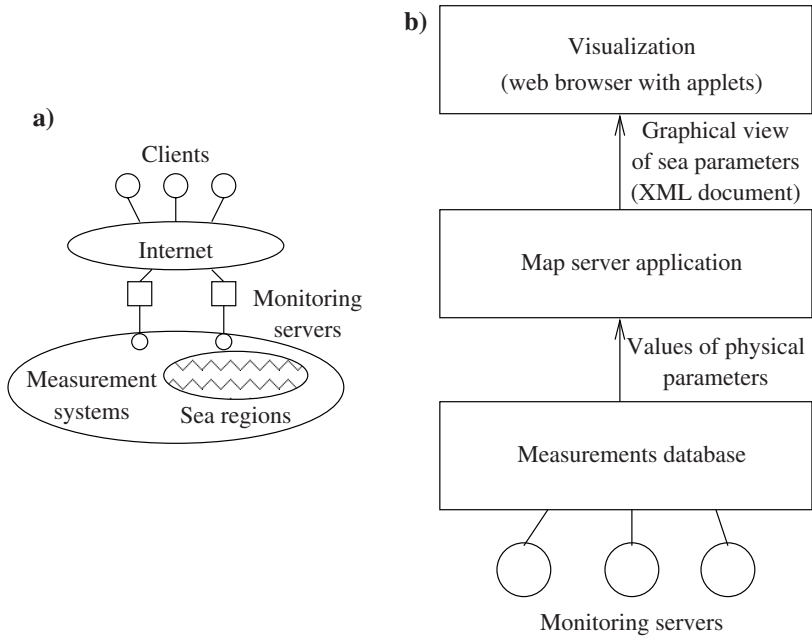
**Abstract.** The paper describes design and implementation problems of on-line Web monitoring and visualization systems. The three-layer architecture is proposed and example of oceanographic map documents (XML) server is presented. A graphical user interface representing user functionality is also given.

## 1 Introduction

Digital maps are becoming an integral part of many monitoring systems. The representative example is a measurement system, which determines different parameters such as: water temperature, drift speed and direction referring to a given geographic region. The system also can simulate different emergency events, such as oil spill at the sea and its dissipation in time. The general architecture is given in Figure 1 a). It is an on-line Web oriented application consisted of three basic components: communication media (Internet), monitoring servers which gather data from measurement systems and clients which display measurement data in a way convenient to end users [1]. In case of high time consuming simulation parallel processing can be used.

Our main goal was to create visualization part of our system as elastic as possible. Classical applications use bitmaps [2], but their big disadvantage is that for every user action like panning, zooming map and toggling layers, a server needs to generate the new map image. Besides, a bitmap could not be effectively utilized in other user applications. Using a vector format more processing is oriented on client, because it stores a whole map and it can zoom, pan and toggle layers without extra connection to server. However there is still one problem: logical structure of the map document is not available, so we cannot effectively process them after generating map images. Using a subset of XML called Scalable Vector Graphics (SVG) we could preserve logical structure of a map document. Eventually we decided to create and use XML Map Documents.

In the paper we present architecture of the designed system and describe its main layers. Next we concentrate on XML Map Documents and its server implementation.



**Fig. 1.** Idea of Monitoring System (a), Layered System Architecture (b)

## 2 Monitoring System Architecture

Figure 1 b) shows more details of the proposed architecture suitable for on-line monitoring system of a sea area. The system consists of three standard tiers: business logic (measurements database with map server application), presentation logic (map server application) and visualization tier.

Measurement data are originally inserted into the database as points. Each point represents a geographic place (it has defined latitude and longitude) and suitable sea or atmospheric parameter like water or salty levels. This data can be obtained from oceanographic models like HIROMB or ICM. Points of measurement data are converted into various map objects, what is done by special scripts or corresponding developed code. It needs many calculations and resources so it is often impossible to make calculations in real time, while serving map document. Therefore we preprocess measurement data and cache them in a database.

In the presentation logic tier, the map and preprocessed measurement data are converted in SVG (XML) document. For implementing such functions we decided to use PHP. The map server takes a map data from the database and convert them to the SVG document.

The last tier – visualization is located on users computer. It is a client application which displays SVG map, allows user to zoom, pan map and select displayed

map's layers etc. Apart from displaying maps it manages map documents and it has JavaScript interface for dynamic visualization of simulations, like oil diffusion in the sea. To create client application we use Java applet technology and Batik library which is a part of The Apache XML Project [3].

### 3 XML Map Documents

We use SVG format to create map documents. Graphical objects in SVG can be grouped into layers. There is possibility to define a user coordinate system and use scripting languages for processing events (e.g. mouse click on a graphical object). SVG allows to insert private application data into file and to create new tags and attributes which defines special shape types. Besides, SVG images can be imported into popular vector image editing applications developed by Adobe, Corel and many more. These applications will ignore a private application data and they will display a graphical content of the file.

One of the challenges was to use SVG in such way, that graphical applications could display as much of the map as possible and we maintain logical structure of the map. Bearing this in mind we designed coding of semantic data referring to layers, map's legend etc. The part of data which corresponds to logical structure of map is ignored by graphical applications but it is used by our applet. Additionally each graphical object corresponds to one map object and is labeled with measured value. This label will be interpreted by our applet and by graphical applications.

### 4 Map Server Implementation and Testing

Map documents generator we have implemented in PHP. Server code performs calculations connected to changing of coordinate system into screen coordinates, generates appropriate map layers and adds XML (SVG) headers.

For developing the client application we have used Batik library which allows to create extensions. We used the extension mechanism to implement a class used to display special map objects (special XML tag added to SVG maps). This possibility is used because, we have encountered some problems with implementation of displaying map symbols, which have special behavior: they must not change their size while zooming and they must not intersect with other symbols on the same layer. Creating our own extension allowed us to optimize symbols rendering speed by more than 5 times.

Figure 2 illustrates clients user interface. On the left side there is a list of available layers, and we can choose layers to display. On the top, there is a toolbar for choosing date of measures, zooming and panning. At the bottom of this applet there is a status bar with cursor's current coordinates and messages. The main part of applet's window displays the map. We can see wind measurements (speed and direction) showed as colored regions and symbols. There is also a small window with some information about selected symbol (list of some other measurements).

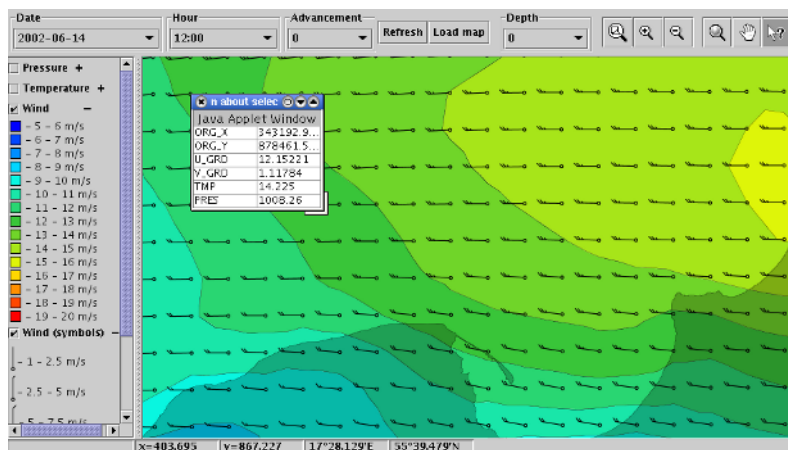


Fig. 2. Applet's graphical user interface with map of wind's speed and direction

## 5 Remarks

The client-server application creates measurement maps without losing any significant data and delivers always present values of measured parameters. Big advantage of our solution is that it can be used as a base for visualizing different simulation. Map displaying applet has a suitable JavaScript interface, which allows viewing of dynamic changes of environment and simulated phenomenon.

The basic features of the proposed solution are as follows:

- flexibility – it can be used in various environments, because a map is created as a XML document, its logical structure can be interpreted. Besides, it can be used as visualization of sea simulations results;
- interchangeability – XML map documents can be used not only with clients (map viewer) but also with many other graphical manipulation applications;
- usability – all maps are on-line updated, so users have always access to latest measurement data;

The obtained Web application is a good example of utilization of Web technologies for creating modern scientific applications.

## References

1. Michał Piotrowski, *MSc. Thesis: Graphical Web Browser Interface To Oceanographic Database*, Gdańsk University of Technology, Department of Computer Architecture, 2002
2. Plewe Brandon, *GIS Online: Information Retrieval, Mapping, and the Internet*, Santa Fe, 1997
3. Apache Software Foundation, *The Apache XML Project*, <http://xml.apache.org>, 2001