DISPLAYING, EDITING AND ANALYZING SPATIAL DATA VIA INTERNET WITH A WEBGIS APPLICATION DEVELOPED USING OPEN SOURCE SOFTWARE

Dr. Tran Trong Duc

Department of Geomatics Engineering
University of Technology – Vietnam National University – Hochiminh City
268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
Email: ttduc@hcmut.edu.vn

ABSTRACT

Easily access to the internet opens up opportunities for using WebGIS to share information of geographic objects distributing on a very large geographical surface. However WebGIS applications, especially those based on open source technology, are often strong in their capability to search and display information on map while paying much less attention on users' need for editing and geoprocessing spatial data. To address this gap, a research is carried out aiming to develop a WebGIS system covering all three major functions using open source technology. The system is built by integrating OpenLayer, GeoServer, and PostgreSQL-PostGIS which run on Tomcat-Apache; and tested successfully using a sample of HCM city's drainage information. This WebGIS system enables users visualizing, inspecting, searching, carrying statistical calculations, editing drainage data, as well as performing basic Geospatial data analysis operations on drainage objects via internet.

1. INTRODUCTION

In 2009, HCMC Urban Drainage Company (HUDC) decided that GIS Technology will enable its staff to manage effectively the complicated drainage system that interweave beneath the inner city larger than 400km^2. HUDC has one headquarter and six satellite companies located at different locations and the challenge is that each of these satellites must be able to access, use, and update information belonging to its responsible part of the drainage network, whereas the headquarter must be able to manage the whole network and database. To implement GIS citywide in a manner that would maximize the access and use of geo-referenced drainage data at minimal cost, one possible solution is to develop a WebGIS system in which GIS data and functionality are made available via the Internet. Users of this system will get spatially referenced geographical information from an interactive web viewer. Processing of GIS functionality is carried out on a server and then transferred to the users' web browser. While WebGIS is built from open source software with minimal cost, its functionality is made available through an ordinary web browser and an integrated viewer with simple, user-friendly interface. Two open source *map servers* with broad deployment and advanced functionality considered are: Minnesota MapServer, and GeoServer. Although both are reportedly compatible with OGC WMS interface, Mapserver only supports basic WFS functions, while GeoServer fully implements OGC WFS supporting additional transaction operations of GIS database including insert, update and delete. HUDC wants its satellite companies to be able to implement some fundamental editing operation on its database therefore GeoServer is chosen. For displaying map data generated from GeoServer, OpenLayers a free mapping library and an integrated component of GeoServer is utilized. In order to provide users with map data via the Internet, GeoServer is installed and run on a web server connected to the Internet. The web server chosen for this WebGIS system is Apache Tomcat Server. Thus the integration of GeoServer, OpenLayers, and Apache Tomcat server create the framework for this WebGIS system, which shall be described in the next part of the paper. In our experiment, drainage data are stored in PostgreSQL/PostGIS relational database and the WebGIS system gets access to the data through GeoTools PostGIS datastores. HUDC aims to use this system to provide a web-based platform for data sharing and managing between its several offices. To access drainage data, HUDC staff just need to have a web browser and their computers connected to the internet.

2. SYSTEM ARCHITECTURE

The overall architecture of the system is depicted in figure 1 and implemented based on the documents listed in the reference section. In this system, GeoServer is an application server providing access to geographic data via WM and WFS interface and providing spatial analysis through WPS interface. GeoServer is based on Java Servlet technology as well as the Spring application server framework and runs in several servlet containers such as Apache Tomcat. One of the main building blocks of GeoServer is GeoTools.

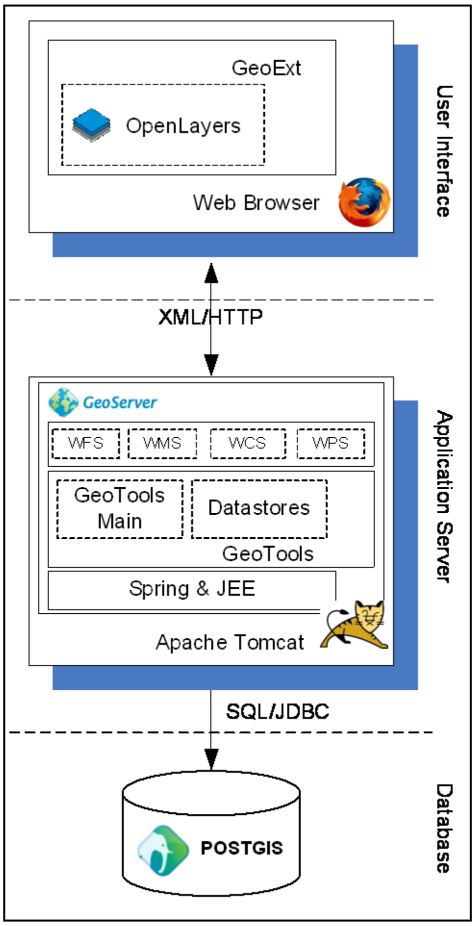

Figure 1. System Architecture

GeoTools is a free software (LGPL) Java GIS toolkit used for developing standard compliant solutions. It provides an implementation of Open Geospatial Consortium (OGC) simple feature model. GeoTools provides the concept of datastores, which allows GeoServer to access PostGIS spatial databases and middleware components in a unified way. GeoServer and its datastores can be easily configured through a browser-based user interface like Mozilla Firefox. OpenLayers is a pure JavaScript library used for displaying map data in most modern web browsers, with no server-side dependencies. GeoExt is a JavaScript library that combines the web mapping library OpenLayers with Extjs, “a cross-browser JavaScript library for building rich internet applications.” GeoExt provides a suite of customizable widgets and data handling support that makes it easy to build applications for viewing, editing, and styling geospatial data.

In this system, users access the server via a Web browser and acquire an HTML form necessary for sending a geospatial data request. Next, users click commands relevant to their request, or enter data into the form and send request with needed parameters to the server. Requests are executed and results are sent back to browser either in form of a map, or a collection of spatial features, text reports or any combination of these items.

3. CASE-STUDY: WEBGIS FOR THE DRAINAGE SYSTEM OF HOCHIMINH CITY

This section describes data-sources, WebGIS implementation and shows some result of the WebGIS designed for the HUDC.

3.1 Data Sources

Data of the drainage system is organized into layers. Each layer contains a collection of related features, such as layer of manholes, of pipelines. Layers are stored in PostgreSQL/PostGIS relational database

3.2 Preparing data to publish on Web

Once the datasets are prepared, the next step is to include them as layers in GeoServer. Geospatial data has no intrinsic visual component, hence to get them visualized, styling must be done. Styling specifies color, thickness, and other visible attributes used to render data on a map. Filters could be used to select a data subset that needs to be rendered in a specific way, for instance use of different colors for different particular administrative units in administration map. Label could be assigned to each geometry feature. Figure 2 illustrates a layer with assigned styles and labels.

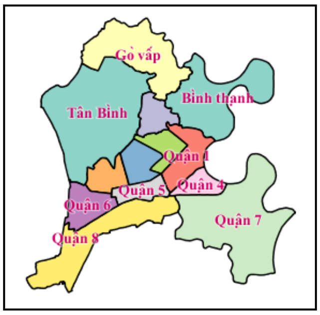

Figure 2: layer with style

3.3 Design of Map Web Page

Next step is to design a user-friendly web map interface. The layout of the default webpage is show in Figure 3.

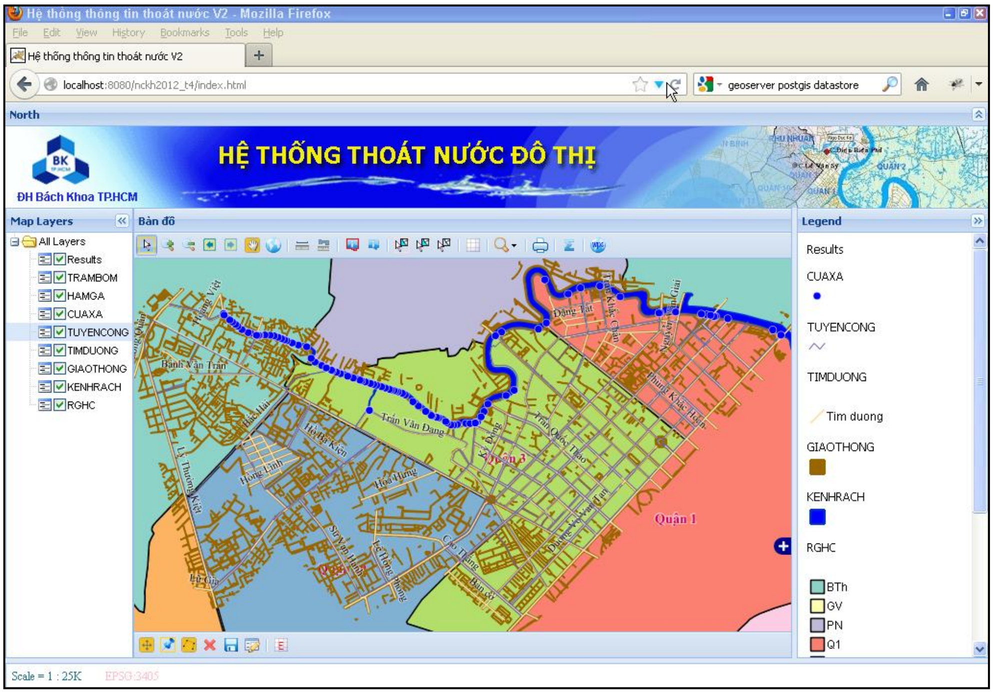

Figure 3. Layout of the map webpage

The top toolbar panel contains tools in the following orders:

- View functions:
  - Navigation tool, *Zoom In* /*Zoom Out* tool ; *Previous Map* /*Next Map* tool: to go back to the previous map or to go to the next map
  - *Zoom to Full Extents* tool : to zoom to the map's full extents.
- Distance functions:
  - *Measure Distance* /*Measure Area* tool: to measure distance or area
- Query functions:
  - *WMS* query / *WFS* query tool: to interactively query information of objects on WMS/WFS Layer;
- Select functions:
  - *WMS Select* /WFS *Select* tool: to select a particular object on the WMS/WFS active layer. WMS selected object is copied and displayed on temporary vector layer named 'results".
  - *WFS Un-Select* tool : to unselect all selected features on WFS layers.
- Attribute Table Function : to open attribute table of selected WFS or WMS Layer.
- Search Functions :
  - *Street Search*: to find any street name entered by user.
  - *Manhole/Pipeline Search: help to find manholes or pipelines satisfied search conditions.*
- Statistics functions : to make some statistics, for example summarize length of pipelines by its level.
- Print functions : to print map
- Geospatial data analysis functions : to do some Geospatial data analysis.

The bottom toolbar panel contains tools applied to WFS layers in the following orders:

- : to move objects; : to modify objects; : to draw objects; : to delete objects; to edit object attributes; to save edited objects on the active layer.
- : to set up edit options

3.4 System implementation

3.4.1 *Codes for the execution*

Codes for the execution are written by using Hypertext Markup Language (HTML) and Javascript.

3.4.2 *Map display*

A Web Map is generated by using "*OpenLayers.Map*" constructor. Layers are created by using Web Map Services or Web Feature Services and then added to the map by using function "*map.addLayer*". Web Map Service (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application.

Web Feature Services (WFS) allows direct access to the features stored on a server. Use a WFS when you want to perform actions such as: query a dataset and retrieve the features, find the feature definition (feature’s property names and types), add features to dataset, delete feature from a dataset, update feature in a dataset. Vector layers can be produced by manually creating Geometries and Features, but it is usually easier to load a data file. To load a vector data file, users can use *OpenLayers.Layer.Vector* constructor with properties like *protocol* and *strategies*. Protocol is used to determine how the data is to be retrieved (HTTP or WFS). Strategies property is used to coordinate feature management for a layer.

3.4.3 *User Interaction*

Controls are OpenLayer's way of interacting with the users. There are over 30 controls available. Controls can be used by adding control to the map through function “*map.addControl*”. Some of them are used in our WebGIS system, such as *Button, DeleteFeature, DragFeature, DragPan, DrawFeature, GetFeature, Measure, ModifyFeature , Navigation, NavigationHistory, OverviewMap, Scale, SelectFeature, ZoomBox, WMSGetFeatureInfo, …*

3.4.4 *Data editing*

To allow a vector layer to be edited, besides the standard way of layer setup, users have to add property *Save strategy* which permits changes on the vector layer to be saved. To edit a vector layer, a control for each edit function is used, in particular, Control “*DrawFeature*” to draw new features, control “*ModifyFeature*” to edit existed feature, control “*DragFeature*” to move features and control “*DeleteFeature*” to delete features. When a user loads the html page, and makes the changes to features with the above controls; each change sets feature state to 'INSERT', 'UPDATE', or 'DELETE'. Once finished, the user presses the  button. This calls *Save* on the strategy, which in turn calls *Commit* on the protocol. This goes through each feature, and for each one where state is set the appropriate HTTP POST/PUT/DELETE transaction is sent to the server.

Beside fundamental editing operations as described above, in this research splitting features is implemented by using control “*Split*” which splits lines into different segments and snapping between features during editing process is done by using control “*Snapping*”.

3.4.5 *Geospatial data analysis*

Geospatial data analysis is implemented through Web Processing Service (WPS). Server is the main component in WPS deployment. It contains a list of processes, accepts requests and retrieves data from them, passes data to the process and generates response with the result. Currently, there are several successfully developing open-source projects, that implement WPS at server side, for example, Deegree WPS , ZOO Project, GeoServer WPS, and 52° North WPS. In this research we use GeoServer WPS. To find out what an OGC server can offer, users request its *GetCapabilities* with the following input *http://localhost:8080/geoserver/ows?service=WPS&version=1.0.0&request=GetCapabilities*. By parsing response from *GetCapabilities* request, the following information can be extracted: web service title and abstract, supported WPS version and operations, list of available processes. Information on each supported process, for example *Buffer* operation, is obtained by issuing a *DescribeProcess* request such as *http://localhost:8080/geoserver/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=JTS:buffer*. The *Execute* operation makes a request to perform the process with input values and output data items as specified. The request may be made either as a GET URL, or a POST with an XML request document. Because the request has a complex structure, the POST form is more typically used. Figure 4 illustrates buffering process through WPS.

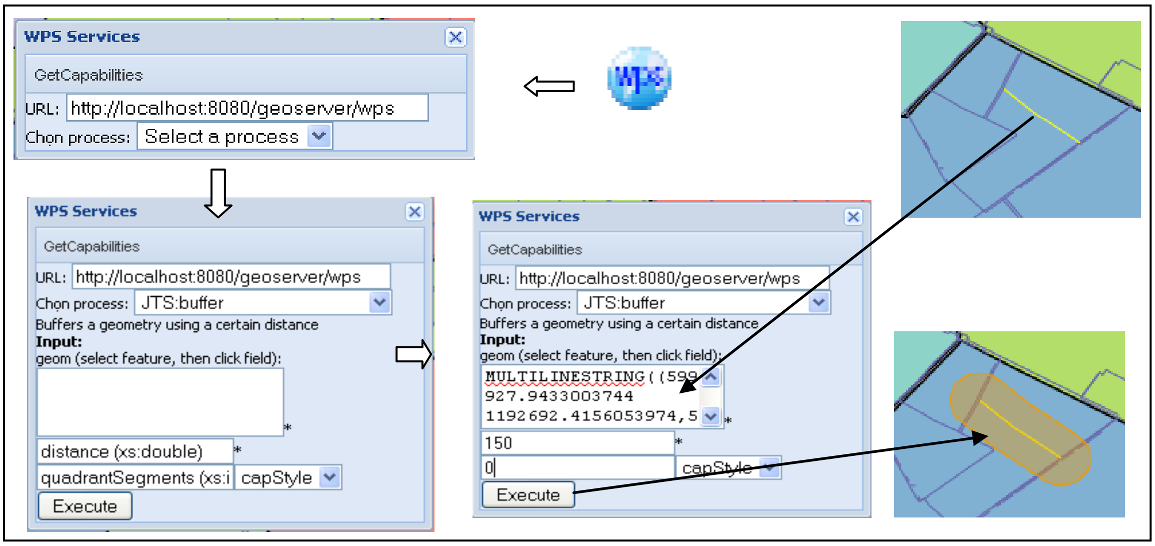

Figure 4: Buffer Processing by using WPS

4. CONCLUSIONS

Integration of GIS and Web technology has opened up a new opportunity for sharing and managing spatial information via the internet. This paper introduces how this opportunity was captured in the design of a WebGIS which is used to manage an urban drainage information system. Conscious to lower investment cost, this simple WebGIS of urban drainage information is implemented based on the framework of open source softwares including GeoServer, OpenLayers, and Apache Tomcat. The system facilitates users from different satellite companies with no special expertise in GIS to be able to access to a common database, display map information, make simple editing operations easily via user-friendly interfaces and even make fundamental Geospatial data analysis . Preliminary results from a pilot at HUDC demonstrates that the use of WebGIS for managing large and complex urban drainage information is effective.

5. ACKNOWLEDGMENT

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number B2012-20-35/HĐ-KHCN.

6. REFERENCES

GeoTools, 2010. Community website http://www.geotools.org
GeoServer,2010. Community website http://www.geoserver.org
GeoExt, 2010. Community website http://www.geoext.org/
OpenGeo, 2010. Community website http://opengeo.org/
OpenLayers, 2010. Community website http:// www.openlayers.org
Peter Robins, 2010. Website http://www.peterrobins.co.uk/it/oledit.html