Available online at www.humg.edu.vn

ESASGD 2016

GIS-IDEAS (2016)

Conference Title: International Conference on GeoInformatics for Spatial-Infrastructure Development in Earth & Allied Sciences (GIS-IDEAS)

Developing a WebGIS application to exploit drainage network data via the internet

Tran Trong Duc*

Ho Chi Minh city University of Technology, VNU – HCM, Vietnam

Abstract

To enable users to remotely access drainage network data and perform analysis via the internet, a WebGIS system has been developed based on ASP.NET framework of Microsoft, the open source products GeoServer, and OpenLayers. This webGIS allows users: i) to securely access drainage data which is stored in relational database SQL Server 2008 and organized based on Multi-user Geodatabase model, ii) to display map layers with flexibly layer style, iii) to search for features based on their attributes and then locate the founded features on map, iv) to search for features in spatial relationship with other features v) to edit drainage network data, vi) to make statistics and create graphs, and vii) to print map. The system has been tested and operates successfully at the Hochiminh city's Urban Drainage Company. Experiment results demonstrate usage capability of the system.

Keywords: GIS, WebGIS, Drainage Network, GeoServer, OpenLayers

1. Introduction

Hochiminh city's Urban Drainage Company (HUDC) has developed a desktop GIS to allow its staff to effectively manage drainage system in Hochiminh city. The challenge is that the company has one headquarter and six satellite firms located at different locations, each of them must be able to access, use, and update information belonging to their responsible part of the drainage network, whereas the headquarter must be able to manage the whole drainage network and database. To implement GIS citywide in a manner that would maximize the use of and access to 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 mainly 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 were considered: Minnesota MapServer, and GeoServer. Although both are reportedly compatible with The Open Geospatial Consortium (OGC)'s Web Map Service (WMS) interface, Mapserver only supports basic Web Feature Service (WFS) functions, while GeoServer fully

* Tran Trong Duc Tel.: (84.8) 38 655 142; fax: (84.8) 38 650 714.
E-mail address: ttduc@hcmut.edu.vn.

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 simple editing operation on its database, therefore GeoServer is chosen. For displaying map data, first data is uploaded to GeoServer, then OpenLayers - a free mapping library - is then used to display layer on map. Prior to developing WebGIS, HUDC's other web applications are also developed. These web applications use Internet Information Services (IIS) of Microsoft as web server. Therefore, the developed WebGIS also use IIS web server. Thus, integration of GeoServer, OpenLayers, and IIS web server forms the framework for the WebGIS system, which shall be described in the next part of the paper. Currently, drainage data is stored based on ESRI Multi-user Geodatabase and stored in SQL Server 2008. The HUDC aims to use this system to provide a web-based platform for geographic data sharing and managing between its several offices. To access drainage data, its staff just need to have a web browser and their computers connected to the internet. This paper will describe the main results achieved through developing such WebGIS.

2. Design of a WebGIS

For the purpose of testing the ability to exploit drainage data and perform simple function via the internet, the WebGIS system has been designed with components described as below.

2.1. System Architecture

To achieve the system flexibility, the popular three-tier architecture is deployed. It is composed of three layers: the *user interface* layer, the *application logic* layer and the *database* layer as shown in Fig. 1.

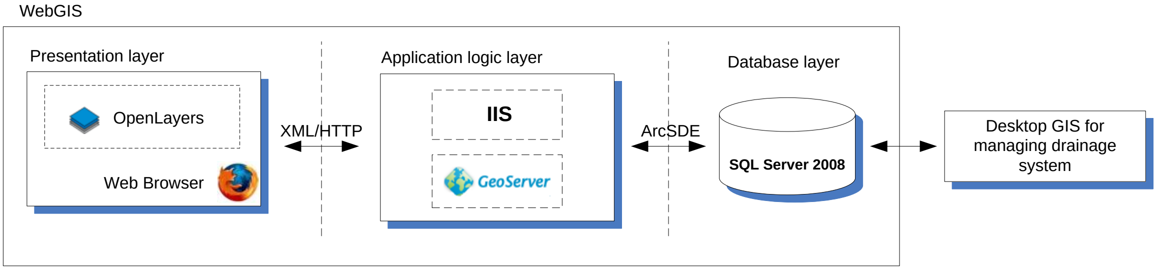


Fig. 1. WebGIS system architecture

- *Presentation layer*: simply a web browser to open Web application page. ASP.Net framework is used to produce dynamic web pages. OpenLayers is a pure JavaScript library used for displaying map data in most modern web browsers, with no server-side dependencies. Web application on client site is developed to have basic common functions as seen in desktop GIS. Detailed design of web application on client site is discussed later.
- *Application layer:* using the Microsoft Internet Information Services (IIS) as Web server and open source GeoServer as a Map Server to provide WMS/WFS data and map related services and provide spatial analysis operation through WPS.
- *Database layer:* HUDC has developed Desktop GIS to manage drainage data. These data are organized based on ESRI Geodatabase model and stored in SQL Server 2008 relational database. ArcSDE is ESRI'S technology for accessing and managing geospatial data within relational databases. To ensure data integrity the WebGIS system will share and exploit this database.

2.2. System Database

Drainage data are stored in VN-2000 National coordinate system and organized into two categories:

- *Base data*: basic geographic data, such as boundary, river, streets, vegetation data ….. These data are used as landmark on field to find thematic data through them.
- *Thematic data*: drainage network data such as drainage pipeline, manhole, pump station, discharge gate …

2.3. System function architecture

Functions needed to manage drainage systems are very diverse, but for testing purposes, only some of the most basic functions of a GIS system are built, including:

a) Load data: dynamically load and remove WMS/WFS drainage data
b) Display map: dynamically display layers with different colors, symbols.
c) Map navigation functions: Zoom in, Zoom out, pan, Zoom to full extent, move back previous window, move to next window, distance measurement, area measurement, move to specific location
d) Query functions: The functions allow
   - To retrieve attributes by direct interact with drainage features on map
   - To search/locate drainage features based on querying their attributes
   - To search for features through their spatial relationships with other features.
e) Statistics: calculations of number of manholes, drainage pipelines, pump stations, discharge gate … on each basin, administration unit, or on each street
f) Graph: plot terrain profile along the chosen manholes of drainage pipeline
g) Print map

These functions are organized into menu or toolbars as described in Fig. 2.

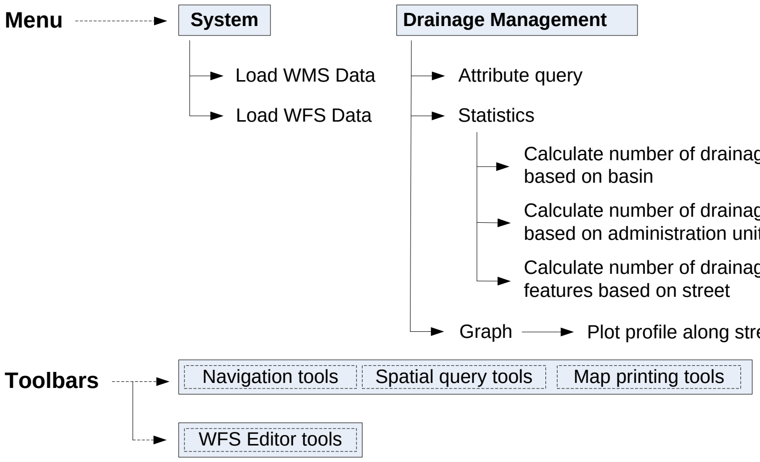

Menu
System
Load WMS Data
Load WFS Data
Drainage Management
Attribute query
Statistics
Calculate number of drainage features based on basin
Calculate number of drainage features based on administration unit
Calculate number of drainage features based on street
Graph
Plot profile along street
Toolbars
Navigation tools
Spatial query tools
Map printing tools
WFS Editor tools

Fig. 2. System functions

2.4. System security

The developed WebGIS in this experiment do not allow registering new user, only validate registered user of the current desktop GIS system of the company through their user name and password. The user who is not authenticated by the system is not able to access the map content of the WebGIS.

3. Developing of the WebGIS for exploiting drainage data through the internet

3.1. *WebGIS layout*

Fig. 4 illustrates the layout of the main web page of the WebGIS system including menu, toolbars and map window showing some of the drainage layer of the study area.

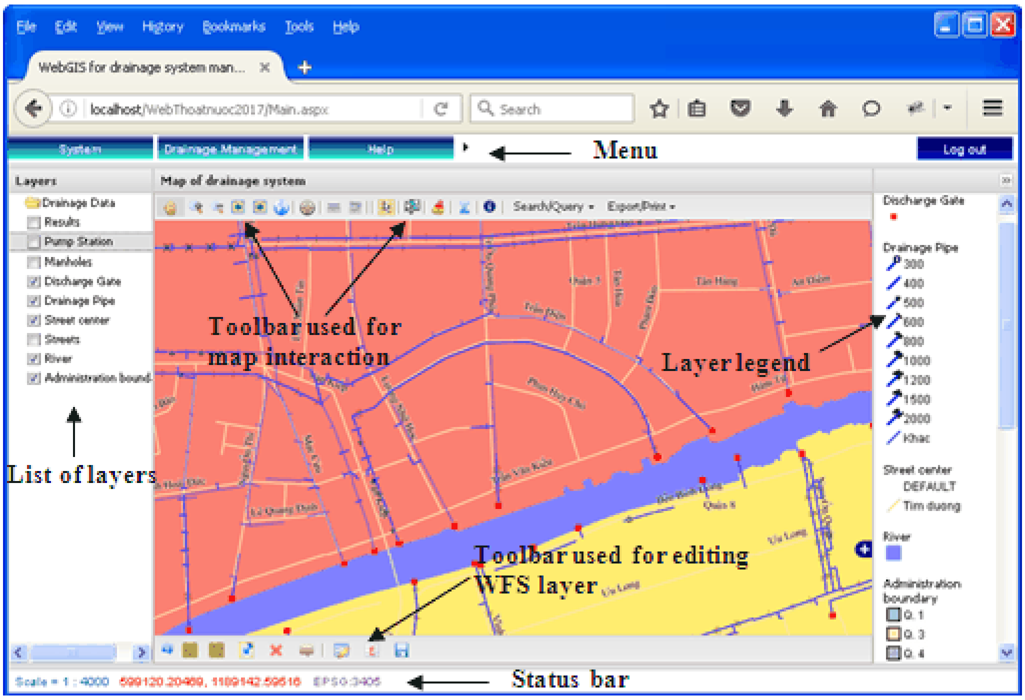


Fig. 4. WebGIS layout

3.2. Map display

The number of layers in the WebGIS are large, while users normally do not require to simultaneously display all data layers. Therefore, the system allows users to select and load only the necessary data layer when needed. Besides, when any layer is not needed anymore, user can remove it from displayed list of layers. Layers are created by requesting Web Map Services or Web Feature Services from web map GeoServer. A Web Map Service (WMS) is a standard protocol for serving (over the Internet) geo-referenced image generated by a Web map server using data from a GIS database. WMS image can't be edited. In contrast to the images served by WMS, WFS allows direct access to each geographic feature. The WFS web service allows features to be queried, updated, created, or deleted by the client. The limitation of requesting WFS layer is time consuming when number of features in layer is huge. This is especially true for drainage data layer like manholes layer which contains nearly sixty three thousands manholes. So, to solve this problem, the system was developed to allow user to decide when to load drainage layer as WMS image or as WFS geographic features. WMS layer is created by using "OpenLayers.Layer.WMS" constructor while WFS layer is created by using "OpenLayers.Layer.vector" constructor with properties like *protocol* and *strategies*. *Protocol* property 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. A Web Map is generated by using "*OpenLayers.Map*" constructor. The created layers are then added to web map by using function "*map.addLayer*" while web map is generated by using "*OpenLayers.Map*" constructor.

3.3. Map Interaction

Interaction with web map created by OpenLayers is done through the *controls*. Before the controls are available for use in map, they must be manually added to the map through function "*map.addControl*". Some of controls are used in our WebGIS system as follows:

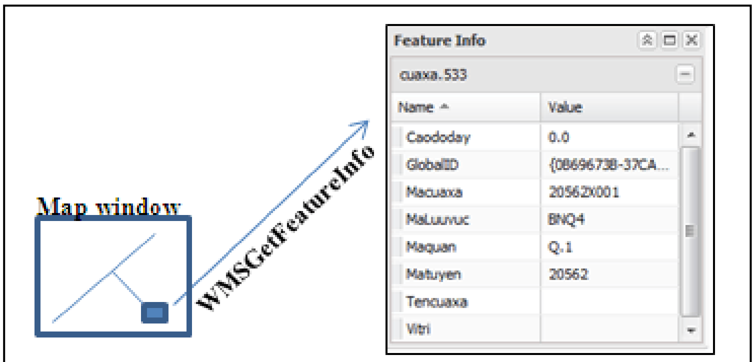
- Control *Navigation* for creating panning tool ✋ used to move map.
- Control *ZoomBox* for creating tools 🔍, 🔍 used to *Zoom in, and Zoom out* map.
- Control *NavigationHistory* for creating tools ⬅, ➡ used to move to previous or next map.
- Control *ZoomToMaxExtent* for creating tool 🌐 used to zoom to maximum extent of map.
- Control *Button* for creating tool ⊕ used to move map to the point with coordinates supplied by user.
- Control *Measure* for creating tools 📏, 📐 used to measure length and area on map.

- Control *GetFeature* for creating tool used to create graphics which indicate features of WMS layer are selected. These graphics are stored in an editable temporary WFS layer. This tool is enable only when WMS layer is selected in list of layers.
- Control *SelectFeature* for creating tool used to create graphics which indicate features of WFS layer are selected. These graphics are stored in an editable temporary WFS layer. This tool is enable only when WFS layer is selected in list of layers.

For more details on how to create tools from controls, please refer to user guide of OpenLayers (http://dev.openlayers.org/apidocs/files/OpenLayers-js.html). The above tools are arranged into top most toolbar as shown in Fig. 4.

3.4. Attribute Query

Inquire of attribute of map feature is done through *identity tool* ℹ. Control *WMSGetFeatureInfo* is used to create this tool. After selecting this tool, simply mouse click feature on the map to get attribute information. When the feature is found, the attributes of the feature will be loaded in the popup window and displays as shown in Fig. 5.



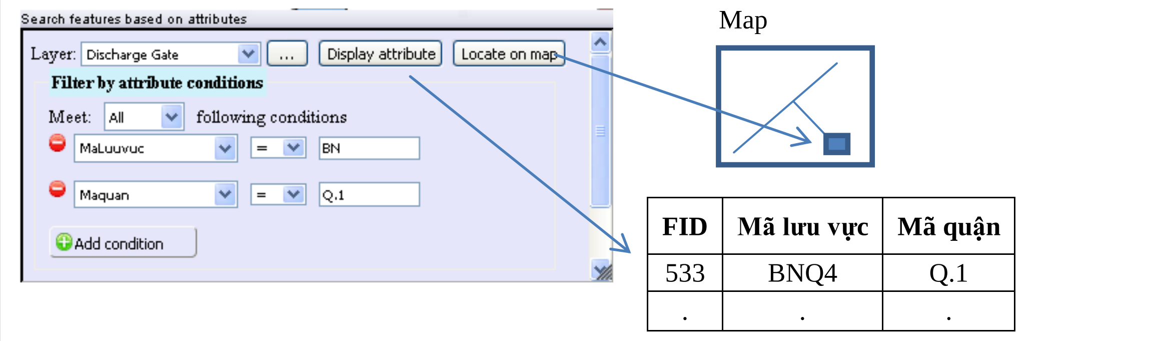
cuaxa.533

Name	Value
Caododay	0.0
GlobalID	{0B696738-37CA...
Macuaxa	20562X001
MaLuuvuc	BNQ4
Maquan	Q.1
Matuyen	20562
Tencuaxa	
Vitri	

Fig. 5. Map interact to get feature attributes through tool ℹ

3.5. Feature Search based on feature attributes

The search for features based on their attributes is done by using function "OpenLayers.Request.Get" or "OpenLayers.Request.Post" and attribute filter <OGC Filter>. To assist in the search for features based on attributes, a search dialog which is provided like in Fig. 6 allows users to search for data of any data layer, and the search is based on the number of unlimited conditions. When features are found, attribute data of found features are loaded into dialog. User can also select option to zoom in and move map to the areas where features are found.



Layer: Discharge Gate

Filter by attribute conditions

Meet: All following conditions

MaLuuvuc = BN

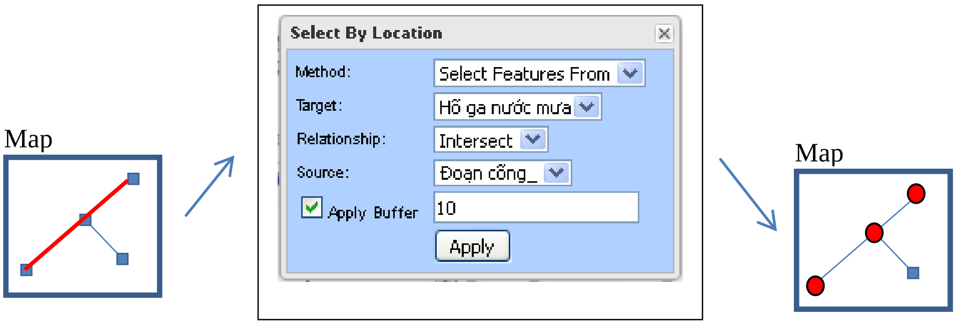
Maquan = Q.1

FID	Mã lưu vực	Mã quận
533	BNQ4	Q.1
.	.	.

Fig. 6. Search features based on condition expressions

3.6. Feature Search based on spatial relationships

Function *Select by location* helps identify features based on spatial relationship with other features. The dialog of the function is shown in Fig. 7. With the parameters entered like in Fig. 7, when implemented it will select manholes (square symbols) within distance of 10m from selected drainage pipeline (when selected on map they will become red line). The process is implemented through two main steps. First, the buffer area with buffer distance equal 10m is created around selected drainage pipeline. To do buffering, *wpsClient* is initialized. It is a protocol for communicating with GeoServer Web Processing Services (WPS). Next, call *wpsClient.execute* to execute operation *JTS:buffer* on geometry of selected pipeline from GeoServer. The second step uses the output buffer to search for manholes intersect with the output buffer by using OpenLayers.Request.POST to send request to server. The request parameters include WFS:GetFeature, name of search layer (in this example, manholes layer), geometry of the output buffer.



Method: Select Features From

Target: Hố ga nước mưa

Relationship: Intersect

Source: Đoạn cống_

☑ Apply Buffer 10

Fig. 7. Dialog of function *Select by location* and performance result of function

3.7. Statistics

Calculation of number of drainage features based on basin, administration unit or street is done on SQL Server 2008. In this WebGIS, requirements of calculation are already prepared and written and saved as the *store procedure* in SQL Server. When user needs to perform any specified calculation, they just press appropriate command button on menu to request implementation of store procedure. Returned results are displayed on web page.

3.8. Draw terrain profile along street

To assist user in drawing terrain profile along street, a profile tool is built to allow users to select manholes along drainage pipeline. This tool is created based on control *WMSGetFeatureInfo*. The control helps to get information about the manholes and drainage pipeline which user selected on map. On the basic of information received from the manholes and drainage pipeline, terrain profile is drawn as shown in Fig. 8.

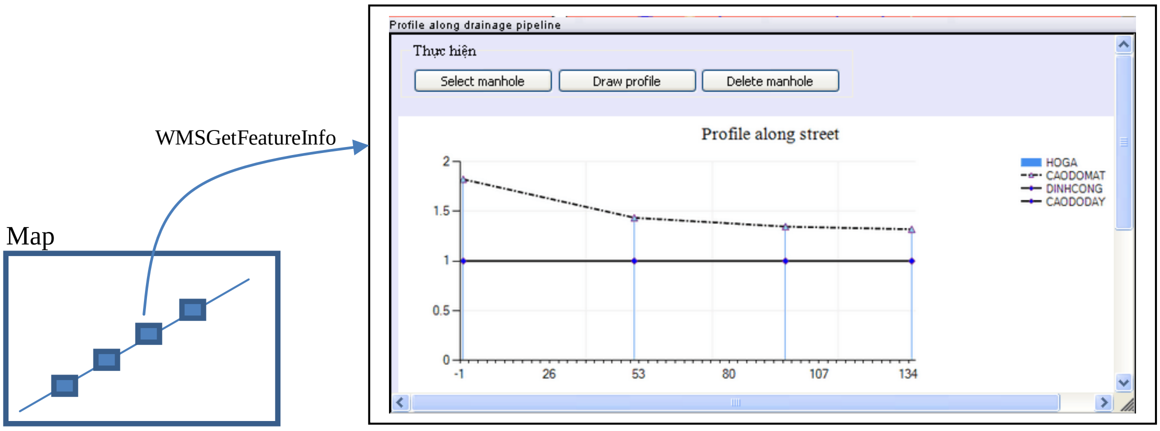


Fig. 8. Terrain profile along pipeline

3.9. WFS layer editing

To allow a WFS vector layer to be edited, besides the standard way of layer creation using “OpenLayers.Layer.vector” constructor, 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 experiment 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*. The above controls are used to create tools that are arrange into bottom toolbar as shown in Fig. 4.

4. Conclusion

Integrating GIS and Web technologies have opened up many opportunities for the sharing of information through the internet. This paper introduces how this opportunity was captured in the design of a WebGIS which is used to exploit an urban drainage system information through the Internet. Focusing on how to reduce investment costs, the system was built using open source software like GeoServer and OpenLayers. However, to take advantage of well-known Microsoft web framework, the system is built base on ASP.NET technology platform and IIS. The system facilitates users from HUDC’s different satellites with no special expertise in GIS to be able to: i) access to a common database, ii) display map information, iii) inquiry feature attributes, iv) search for features based on condition expression, v) search for features in spatial relationships with other features on map, vi) make simple editing operations, vii) create terrain profiles along street and viii) print map. Preliminary results from a pilot at HUDC demonstrates that the use of WebGIS for exploiting large and complex urban drainage network on the internet is effective.

References

GeoServer, 2016. Community website http://www.geoserver.org/
MapServer, 2016. Community website http://mapserver.org/
OpenLayers, 2016. Community website http:// www.openlayers.org/