POLITECNICO DI MILANO

Environmental, Hydraulic, Infrastructure and Surveying Engineering (DIIAR)

DOCTORAL PROGRAMME IN XXV Cycle

HDI4NVN– Hydrological Data Infrastructure for North Vietnam

Doctoral Dissertation of:

Truong Xuan Quang

Supervisor:

Prof. Maria Antonia Brovelli

Tutor:

Prof. Fernando Sansò

PhD coordinator:

Prof. Guadagnini Alberto

Year: 2013-Cycle XXV

ABSTRACT

Global climate change is increasing the risk of exposure of million of people all over the world at water stress and/or extreme events of flood and drought. The presented research aims at building a spatial infrastructure for monitoring the hydrological status of an area in the northern part of Vietnam. The area is of particular interest, being affected by natural calamities like droughts, floods, landslides and so on.

In this project we present the Intelligent Geoportal based on OGC Web Services (WMS, WFS, SOS, WPS) we implemented to manage, analyze and visualize about 50 years of hydrological and environmental data. . The Intelligent Geoportal is completely based on Free and Open Source Software (FOSS); it makes available the following functionalities: (1) storing and managing base map such as land use and land cover generated form free Landsat TM, DEM, aspect, hillshade from ASTER and SRTM and sensors' observations, (2) searching and querying data and observations including register sensors, insert observation values of hydrology and environmental data to database management system for both data sources from ground based and satellite data, (3) getting information such as statistical analysis and statistical chart from hydrological and environmental sensors at daily, monthly and yearly intervals. This project not only used historical rain gauges/environmental sensors (RG) but also precipitation data from satellite data (PERSIANN), however PERSIANN data need to be monitored before using because in some case rainfall is affected by topology of terrain.

We not only present our Intelligent Geoportal for the North of Vietnam but we also concentrate to present our DSS for monitoring satellite precipitation (PERSIANN) base on Web Service. DSS was developed as a tool it can compare directly or indirectly between RG and PERSIANN product by mean short-term or long-term. Users may compare in daily interval by using interpolation of RG and associate with visualization of PERSIANN. Otherwise we also can request statistical tool for estimating a correlation and percentage of simultaneous rain or no-rain. At last, statistical tool and RUSLE are able to estimate accuracy of PERSIANN comparing with RG data.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

At first, I would like to express my deepest gratitude to my supervisor Prof. Maria Antonia Brovelli for the continuous support of my PhD study and research, for her understanding, enthusiasm, insightful comments and highly valuable knowledge. Her guidance was invaluable to me in all the time of research and writing of this thesis.

Secondly, I would like to acknowledge my sincerest gratitude to my tutor Prof. Fernando Sansò for his guidance and direction of this work; he also gave me a lot of chances and opportunities for me during my PhD course.

I would like also to thank other professors at D.I.I.A.R faculty in Politecnico di Milano for their teaching, encouragement, and helpful instructions, which have contributed to the completion of my work.

My sincere thanks also goes to all my friends for offering me many supports since my very first day in Italy and throughout my three-year stay in this beautiful country, for all the happy moments that we had together.

I'm pleased to thank my colleagues who supported me and created a friendly working environment. I truly enjoyed working with them and owed them for all the advices shared with me.

I would like to thank Prof. Venkatesh Rahavan for his kind advices and valuable comments in general during my stay in Italy for my PhD course. I have aslo to thank the main developer of istSOS framework Dr. Massimiliano Cannata and ZOO-Project framework Mr. Gerald FENOY, I could not imagine that without their support how much work have done. And I also Thank Dr. Giorgio Zamboni who has answered almost my relevant questions for my works. I thank Mirko Reguzzoni, Marco Negretti, Maddalena Gilardoni, Andrea Gatti and all of PhD students in my laboratory when I was there at Geomatics Laboratory in Como Campus, thank Satetta Lucca, Luana Valentini, Laura Carcano and Marco Minghini, Do Hong Khanh for all their help.

I also would like to give special thanks to Prof. Andrea Castelletti and Dr. Xuan Quach Thi for having kindly provided the hydrological measurement for this work.

I would like to thank my family: my parents for their forever support and encouragement to me.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ASTER: Advanced Space borne Thermal Emission and Reflection Radiometer
CGI: Common Gateway Interface
DBMS: Database Management System
DEM: Digital Elevation Model
ETM: Enhance Thematic Mapper
GDEM: Digital Elevation Model
GEOSS: Global Earth Observation System of Systems
GI: Geographic Information
GIS: Geographical Information system
GLCF: Global Land Cover Facility
GMES: Global Monitoring for Environment and Security
INSPIRE: Infrastructure for Spatial Information in Europe
HDI4NVN: Hydrological Data Infrastructure for North of Vietnam
HTTP: Hyper Text Transport Protocol
HTML: Hyper Text Markup Language
JSON :JavaScript Object Notation
NASA: National Aeronautics and Space Administration
NSDI: National Spatial Data Infrastructure
OWS: Standards for Web Services
OGC: Open Geospatial Consortium
PERSIANN: Precipitation Estimation from Remote Sensed Information using Artificial Neural Networks
RDF: Resource Description Framework
RG: Rain Gauges
RS: Remote Sensing
RUSLE: Revised Universal Soil Loss Equation
SDI: Spatial Data Infrastructure
SDSS: Spatial Decision Support System
SEIS: Shared Environmental Information System
SII: spatial information infrastructure
SLD: Style Layer Descriptor Profile
SOS: Sensor Observation Service
SOAP: Simple Object Access Protocol
SP : Satellite Precipitation
SRTM: Shuttle Radar Topography Mission
SVG: Scalable Vector Graphics
TCP/IP : Transmission Control Protocol/Internet Protocol
WSDL: Web Services Description Language
WMS: Web Map Service
WFS: Web Feature Services
WCS:Web Coverage Service
WPS : Web Processing Service
XML: eXtensive Markup Language

CHAPTER I: INTRODUCTION

1.1 Overview

In Vietnam between 1980 and 2010, about 16000 people lost their lives as a result of natural disaster such as: storms, floods, and land slides (see Fig. 1). And almost 74 million people were affected as a result of natural disasters and estimated economic damages are around 8 billion USD (PW, 2012) see (Table 1***Table 1:*** *Natural Disaster Occurence Reported see PW, 2012*). According to the World Bank's 2008 Global Monitoring Report, Vietnam ranks eighth in the ten most vulnerable countries in East Asia to weather extremes. Recent study of Oxfam's experience estimated that 70% of country's populations live in areas subject to water related natural disasters (Oxfam, 2008). The study area on which we concentrated lies in a tropical zone; in addition, it includes the two biggest hydroelectric dams in the South East Asia, and also contains a number of faults, the most active of which with a maximum shaking intensity of 8-9 (MSK scale).

No of events:	159
No of people killed	16,099
Average killed per year	519
No of people affected	73,582,754
Average affected per year	2,373,637
Economic Damage (US$ X 1,000)	7,955,750
Economic Damage per year (US$ X 1,000)	256,637

Table 1: Natural Disaster Occurence Reported see PW, 2012

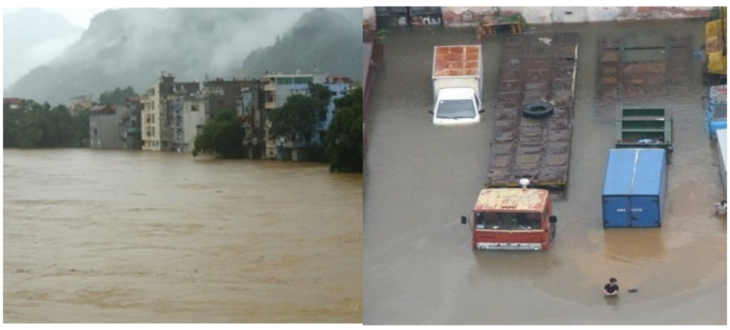


Figure 1: Left: A raising of water level along Lo river (III flood alert level) Jul-2007Right is Flooding on Dinh cong Street, Hanoi in Nov 2008

The frequency and intensity of natural disaster have increased in the last few year decades. According to the World Disaster Report (2011), some natural occur in an abrupt manner and affect large areas, therefore it is difficult to predict, for example: tsunamis, tornados, earthquakes. On the other hand, disaster such as: floods and landslides trend to be mapped more easily, and the people who will potentially be affected by disasters can be predicted in advance.

Based on the characteristic of Remote Sensing and Geographical Information system (GIS), they are playing important role in a suitable strategy for disaster management, monitoring, assessment and mitigation, etc. Due to their broad application, remote sensing (RS), Geographics Information System (GIS) are strong tool for studying above problems.

However, to effectively use RS and GIS techniques for risk and natural disaster management, it is build a Spatial Data Infrastructure (SDI). SDI is a system which all data and information must be available for all levels of government, commercial and no profit-sectors, users in general and scientists. A SDI has been proposed by scientists to address data sharing issues. At First, SDI is the idea from McLaughlin (1991) and nowadays SDIs have diffused across the world.

At the beginning of 2003, a new Geo-portal becomes available which hosted different Geospatial Web Services. Several studies discuss Geo-portal such as Bernard et al (2005) presented the initial steps in order to establish a European Geographic Information Portal (European Geoportal). Maguire and Longley (2005) discussed about contribution of Geoportal and SDI made to simplifying access to Geographic Information (GI) and Tait (2005) reviewed four key Geoportal projects and gained experiences through their implementation.

The idea in order to set up a Web-Based decision-support tool or to provide GIS functions over the Internet presented by Pandey (2001). Recently, many Spatial Decision Support System (SDSS) have been developed in various study areas over the world such as DSS for watershed management Jin-Yong et al (2005), earthquake disaster reduction Jin et al., (2009); flood prediction and monitoring, integrating hy-drological modeling and GIS Mioc et al., (2010). The advantage of the proposed SDSS consists in the fact that it is based on standard geographic web services and a relevant intelligent geoportal suitable for querying, analyzing the data, and for specific problems.

The rest of this chapter is organized as follows. In section 2, we review the Spatial Data Infrastructure (SDI). In section 3, we explore a Geoportal including technology used in a Geoportal and finally we present an Intelligent Geoportal. Section 4 we summarize some previous Web-Based Spatial Decision Support System (SDSS). Finally in Section 5, we present our motivation for building build internet-based hydrological data infrastructure including intelligent geoportal and SDSS for study area.

1.1 Spatial Data Infrastructure (SDI)

National Spatial Data Infrastructure was first presented by John McLaughlin 1991 at Canadian Conference on Geographics Information Systems; the ideas contained in McLaughlin's paper were developed by the United States National Research Council's Mapping Science Committee. After two decades, SDIs have applied over the world. The development of SDIs at global, national, and sub-national levels has been increasingly documented by number of studies, such as: Masser (2005, 2009), Williamson (2003), Vandenbroucke (2005).

There are 83 SDIs at the national level by the end of 2005 Crompvoets and Bregt (2007) and the number of SDIs has increased more than hundred until now. SDIs are created to facilitate sharing of data in the national context by mean standards Groot and McLaughlin (2000), the data such as: national spatial reference systems, a national topographic template, national elevation model, administrator boundaries, certain thematic data sets (soils, hydrology, vegetation population, etc.). The term Spatial Data Infrastructure is described in the SDI Cookbook Nebert (2004) as "the relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data".

SDI issues such as the Infrastructure for Spatial Information in Europe (INSPIRE), GMES (Global Monitoring for Environment and Security), GEOSS (Global Earth Observation System of Systems), SEIS (Shared Environmental Information System) powered by standardization efforts of international organizations such as OGC (Open Geospatial Consortium), etc.

SDI has more than a set of data or database, it hosts geographic data and attributes, sufficient documentation (metadata), SDI makes its data available for many levels of governments, organizations (commercial and no-profit sectors), academia and citizens in general in order to use spatial data in an efficient and flexible way. The SDI provides a basis for spatial data discovery, evaluation, and application for users. Some examples of existing SDI are the National Spatial Data Infrastructure (NSDI) in the United States; At the European side: INSPIRE is a European Commission initiative to build a European SDI and beyond, many national SDIs are existing such as Swiss Federal Spatial Data Infrastructure, Canadian Geospatial Data Infrastructure, etc.

The application of SDI continues to evolve, because it becomes a core of infrastructure in order to support for economic development, environmental management and social stability in developed and developing countries. Based on its dynamic and complex nature, it is still absent in some developing countries.

Building SDI is a complex task, not just because of the evolving nature of the SDI concept, but as much because of social, political, cultural and technological context to which such development must respond and the development of national SDI in developing countries with a view to assessing the success, challenges, and problem Williamson (2003).

1.2 Geoportal

A geoportal is a Web-accessible catalogue that specializes in the posting, discovery, and exchange of map based geographic information resources and is commonly installed as an element of a more comprehensive spatial data infrastructure (SDI). A Geoprotal can also understand as a website to be an entry point to geographic content on the web or, more simply, a geographic portal is a web site where the discovery of geographic content is a primary focus Tait (2005). Geoportals serve as standard user interfaces for SDIs and act as access points to metadata. The aim of these Geoportals is to enable users to discover, use and publish metadata. Advantages of using Geoportal are to increase the efficiency of GIS data distribution, and to reduce redundant data creation, and standardizing the quality of geospatial data used in timely operations.

Technology of Geoportals based on internet technology and distributed GIS. The internet technology includes of information technology standards, such as: Transmission Control Protocol/Internet Protocol (TCP/IP), Hyper Text Transport Protocol (HTTP), Hyper Text Markup Language (HTML), and eXtensive Markup Language (XML), software and network infrastructure. Distributed GIS is simply GIS technology that has been built and deployed using standards and software supporting Geospatial data available via Internet.

A geoportal is implemented using three components see Table 2: A web site presents the geographic application or portal; web service publishes geographic functionality as a web service; and data management software provides a management for both raster and vector geographic content.

Components	Elements	Environments	Functions
Web Portal	Geographic Web Site	HTML, HTTP, XSL, XML, JSP, ASP	Search, Map View, Publish, Administrate, etc..
	Web Control	Java Beans, .NET	Query, location, mapping, edit, Geocoding
Web Service	Geographic Web Services	XML, SOAP, WSDL, WMS, WFS, GML	Query, Map render/feature, Transaction, Geocode
Data Management	DBMS	SQL	Raster, Vector, Tabular

Table 2: Distributed Geoportal architecture Tail (2005)

A Geographic Web Site is developed and deployed using standardization way, and is also consists of two components: the website framework and the functional tools. The website framework presents the information of Geoportal via a graphical user interface. The second element is the functional tools, makes available to use GIS functions such as: geocoding, search, query, gazetteer linkage, statistical tools, etc.

Geographic web services implement functionality and publish geographic data, information technology standards such as eXtensible Markup Language (XML), Simple Object Access Protocol (SOAP), and Web Services Description Language (WSDL). Additionally, the other geographic web service is also diffused over the world and applied in many project works such as Open GIS consortium (OGC) standards. Geographic web service functionality is published in Geoportal includes: map rendering; feature steaming; queries; searches; metadata query and management; network analyses; 3D terrain visualization; statistical tools; etc.

Beside above Geoportal presented above, the term "Intelligent Geoportal" was presented by V. Rautenbach (2012), Geoportal was defined as typically provides access to spatial data and associated web services in an SDI, facilitating the discovery, display, editing and analysis of data. Otherwise, spatial information infrastructure (SII) was also presented as the power tool in order to provide access to information, and data inside SII should be completely processed, organized and presented though user's interface. Furthermore, Intelligent Geoportal was mentioned as a Geoportal that provides complex functionality through user interface for a user in a specific application domain Iwaniak et al., (2011).

1.3 Geoportal examples

In this section we present an overview some of the best Geoportal. Geoportal could be classified by: Geographical scope (local, regional, national, continental); Type of end-user (children, education, professional non scientist, environmental, consultants, geoscientists, etc.); Level and domain of application: fundamental research, environmental protection, energy and mineral exploration; Number of datasets available: only one, several, many (catalogue); Service level : discover, view, report, analysis, download.

1.3.1 European Geoportal

Infrastructure for Spatial Information in Europe (INSPIRE), a first release of the INSPIRE Geoportal was published in November 2011. The INSPIRE geoportal provides the means to search for spatial data sets and spatial data services, and subject to access restrictions, to view spatial data sets from the EU Member States within the framework of the INSPIRE directive, it includes a number of common components (see also Fig 2):

- INSPRIRE geoportal allows users to search, discover and access geographic information provided by European governmental, commercial and no-commercial organizations.
- Create data according to the INSPIRE implementation rules
- Test the compliancy of INSPIRE metadata with the INSPIRE Metadata Regulation.

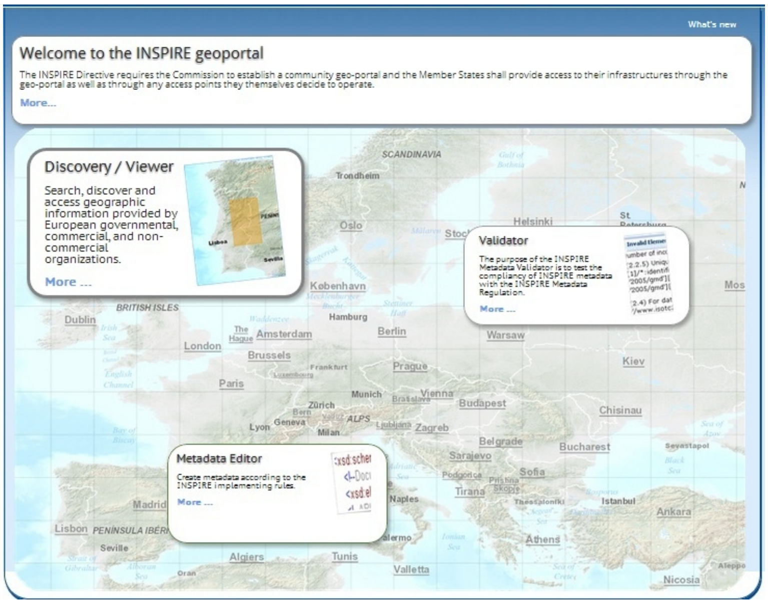


Figure 2*:* INSPIRE geoportal's interface

1.3.2 National Geoportal

a) Swiss federal GeoPortal (geo.admin.ch)

The Swiss federal geoportal is a publicly accessible platform for geographical information, data and services. Geographical information consists of geographically and spatially referenced data. It describes the characteristics of a country/region in the form of coordinates and place names. Through geo.admin.ch user can access a rich of information about geodata covering the most important aspects of life, including the environment, population, health and security, economy and many other themes. Principal of Swiss federal geoportal supports many thematic "map viewer" (Fig 3) facilitate for users in order to search, display and use of geographical information, it is also possible to access main functionality such as: map view, change language, size of the map control, display coordinates, zoom, geographical search, full text search for data, change with visibility of the selected data-set, measuring, drawing, WMS import, KML import, etc.

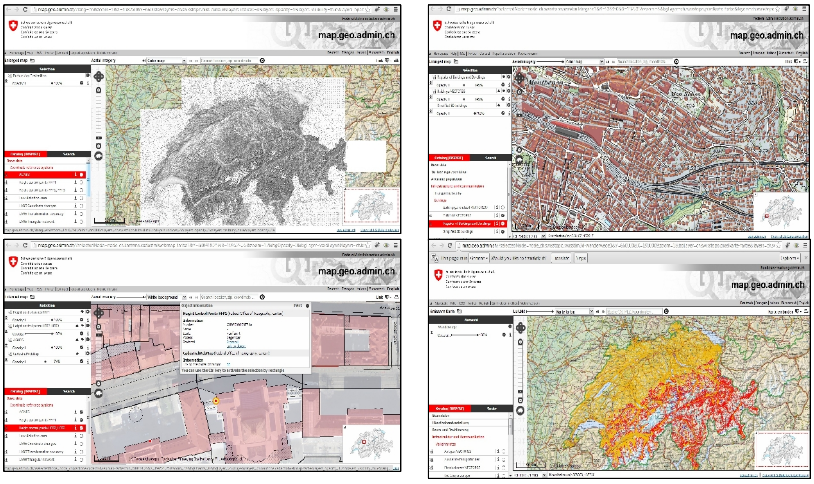


Figure 3: Swiss federal GeoPortal, the upper left is Cadastral Web Map, the upper right is Simplified 3D building, the lower left is Dufour Map, and the lower right is the Hiking trails

b) Danish portal

In addition to the simple interactive web view tools of Danish portal, another useful function is the option to print the information on a point. for instance, the Danish portal http://kort.arealinfo.dk/ proposes to export the result of the "I" option (getFeatureInfo) in Excel. The print tool (for the full map) directly sends to the print, or generates a PDF (Fig 4).

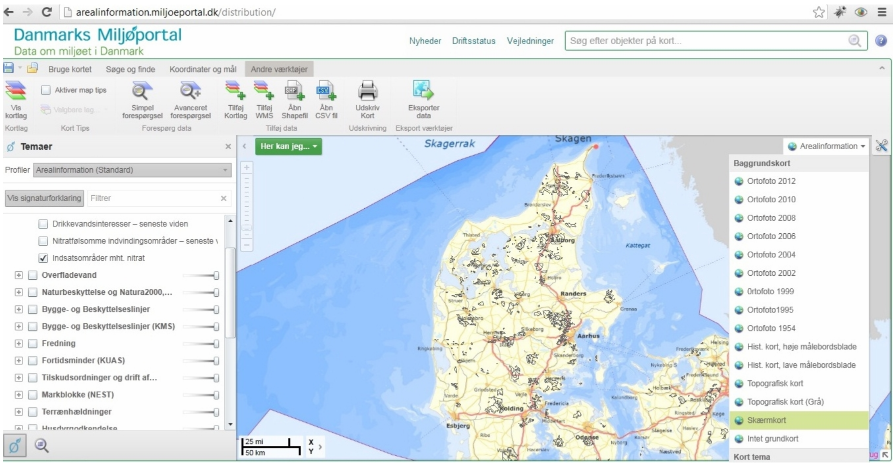


Figure 4: Danish portal.

1.3.3 Geoportal for earth observations (GEOSS)

The GEOSS uses "Flash-like" technologies in map client http://www.geoportal.org, it could be a new generation of map client, very much interesting to non professional users (Fig 5).

Figure 5: GEOSS Geoportal interface

The GEOSS Common Infrastructure allows the user of Earth observations to access, search and use the data, information, tools and services available through the GEOSS system, it includes two main elements as bellows:

- The GEOSS Components and Services Registry are similar to a library. All of the governments and organizations that contribute components and services to GEOSS provide essential details about the name, contents, and management of their contribution.

- The GEOSS Standards and Interoperability Registry enable to configure their systems then they can share information with other systems.

1.3.4 Volta Basin Authority Geoportal

The Volta Basin Authority (VBA) geoportal was developed by the university of Bonn. VBA Geoportal based on GeoNetwork (http://geonetwork-opensource.org/), GeoNetWork is a free and open source cataloging application for spatially referenced resources. It provides powerful metadata editing and search functions as well as an embedded interactive web map viewer. It is currently used in numerous Spatial Data Infrastructure initiatives across the world.

VBA geoportal is a catalogue of location-oriented information. It is able to search related information correspond to areas around the world otherwise users also can access metadata and interactive with a map (Fig 6).

Figure 6: Volta Basin Authority geoportal interface

1.4.5 Geoportal to download datasets (GEO Portal)

Many geoportal allow to download a dataset. The option might propose different formats, and might be constrained by licence and security. The Asia and the Pacific GEO Data Portal http://geodata.rrcap.unep.org/ (Fig. 7) allows the users to search for a dataset, then download it. Several formats are proposed, depending on the dataset (PDF, ArcInfo e00, ASCII Grid, BIL Image, ESRI Shapefile, SVG, TIF image).

Figure 7: GEO portal interface.

1.3.6 German Environmental Information Portal (PortalU)

The PortalU (http://www.portalu.de/) provides various possibilities for searching official environmental information. PortalU enables user to search in both catalogues and environmental databases from the federation, the German federal states and local authorities. The search engine also discovers resources from the "Deep Web" that can not be found by other search engines. The following features will help user to specify their search (Fig. 8):

- Search: finding environment information, legally relevant topics, research projects and addresses of environmental authorities by using the full text search.
- Topic: finding information to particular environmental topic by using the PortalU subject research.
- Data: Browse the catalogue for official environmental monitoring data.

- Service: Under PortalU Service all the press releases, publications and event notes of German official environmental agencies are collected together. You can search in all data base fields or restrict your search to a certain topic (e.g. air pollution).
- Catalogue: Display metadata and objects from environmental catalogues.
- Map: Search for thematic maps of environmental authorities.

Figure 8: German Environmental Information Portal interface

1.3.7 Digitaler Atlas Nord Geoportal (regional geoportal)

The Digital Atlas North http://digitaleratlasnord.de constitutes a geodata portal of the Land Schleswig-Holstein, the municipalities of Schleswig-Holstein and the Hanseatic City of Hamburg. The data is available in the spatial information portal "Digital Atlas Nord" It is used for visualizing and access of information. The data services are provided by local and regional governance and are presented homogenously in the Digital Atlas North (Fig. 9).

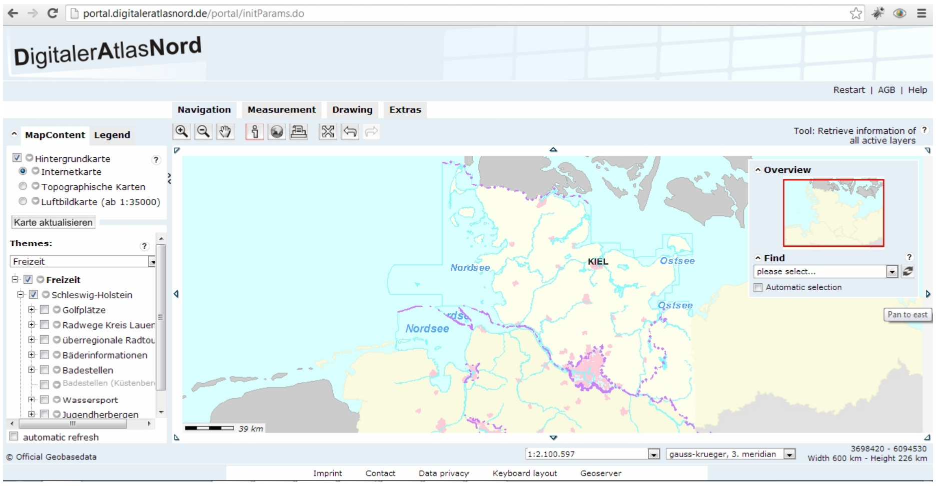


Figure 9: Digitaler Atlas Nord Geoportal interface

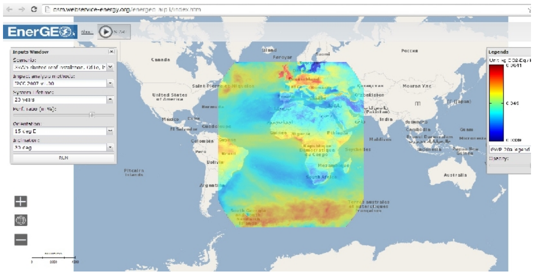
1.3.8 EnerGE Geoportal

The EnerGEO Geoportal (http://energeo.researchstudio.at) is part of the EU P7 EnerGEO project (http://www.energeo-project.eu/). It provides access to energy pilot mapping application. EnerGEO is focusing on integrating technology, policies, standards, human resources and related activities necessary to acquire, process, distribute, use, maintain, and preserve spatial information. EnerGEO geoportal including: a large number of thematic geoportal, in the list below is main thematic geoportals of EnerGEO sytem:

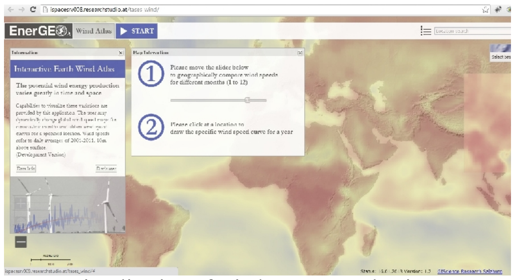
- Visualization of wind energy production: capabilities to visualize these variations are provided by this application. The user may dynamically change global wind speed maps for consecutive months and obtain wind speed curves for specified location
- Search the Geoportal for spatial resource
- The Wind Pilot: enables to assess the environmental performance of an wind farm over its life time. Maps of environmental performances are provided for a northern European coverage according to different maintenance and failure rate schemes, life times of the wind turbines, number of wind turbines per farm. These scenarios are to be defined by the user himself with the relevant menus (Version 1.0 release in September 2012)
- Bioenergy Assessment Pakistan: the service provides information on modeled annual bio-energy potentials within the time range from 2000 to 2010. User can using time scale-bar in order to see change of bio-energy map of Pakistan corresponding to time in scale-bar.

- Energy Map Based Scenario: a web-based tool to assess the environmental performance of the production, transportation and use of photovoltaic systems.
- Energy scenarios of Sauwald region
- EnerGEO Map-Viewer template: an Open-Source Web Map application for visualization and querying of WMS Layer using the Openlayer 2.11 (http://openlayers.org/) and ExtJS.

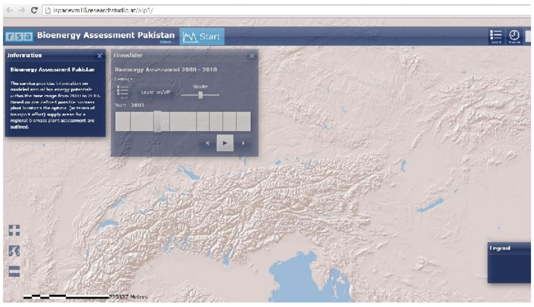
To illustrate for set of above geoportal in the (Fig 10)



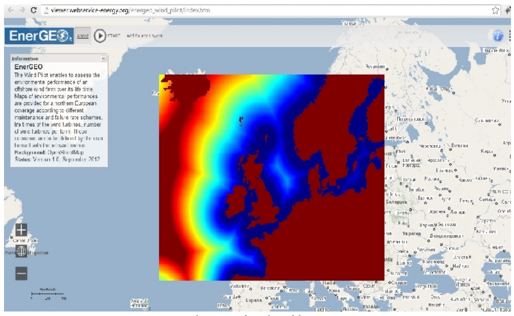
Energy Map Based Scenario



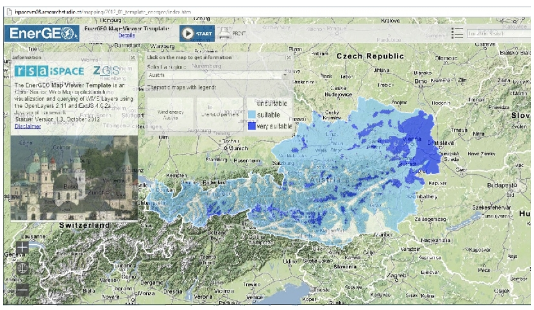
Visualization of wind energy production



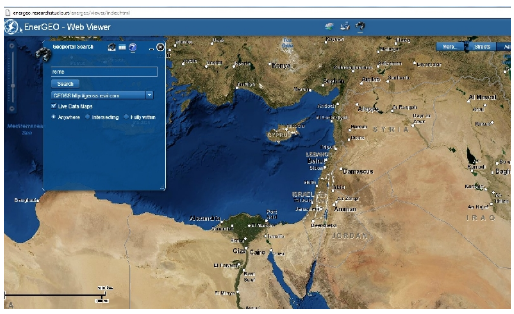
Bioenergy Assessment Pakistan



The Wind Pilot



EnerGEO Map-Viewer template



EnerGEO Web Viewer

Figure 10: some example of thematic geoportal interfaces of EnerGEO sytem

1.3.9 seNorge.no Hazard control

A seNorge.no, http://senorge.no is a geoportal presenting snow (snow amount, age, depth, melt, etc), weather (precipitation, temperature), water (rain and snow melt, soil water, ground water) and climate maps for Norway updated daily. Data are available as daily, monthly and annual values, as well as for climate periods and scenarios. Otherwise, User can request for the others information such as contours, hydropower, catchments, topography, and satellite image. In the navigation toolbar, it is flexible designed to control time (time navigation) , map (map navigation), in particular, user is able to request graph and data by using the navigation toolbar.

This geoportal is a good example of user-friendly interface (Fig. 11). The time navigation allows to display old maps and is very useful for hazard mitigation for flood, drought, energy supply shortages, avalanches and landslides and climate change.

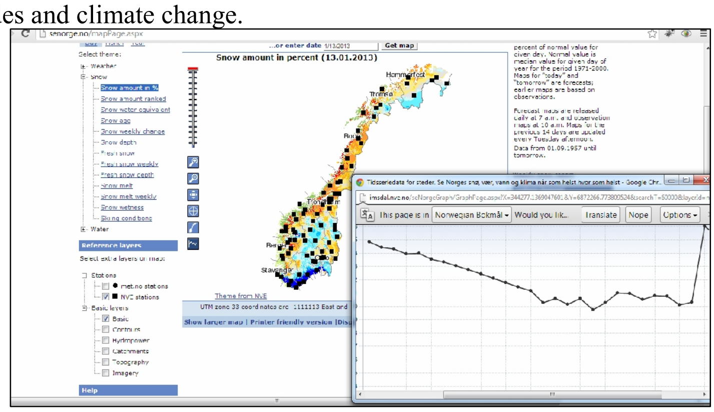


Figure 11: seNorge.no geoportal interface

1.3.10 European Environment Agency Geoportal

The European Environment Agency http://www.eea.europa.eu provides many functions (Fig. 12). One of them displays a map and explain what is it in a very used-friendly way. The interface proposes the map, but also the same data displayed as graphs and tables.

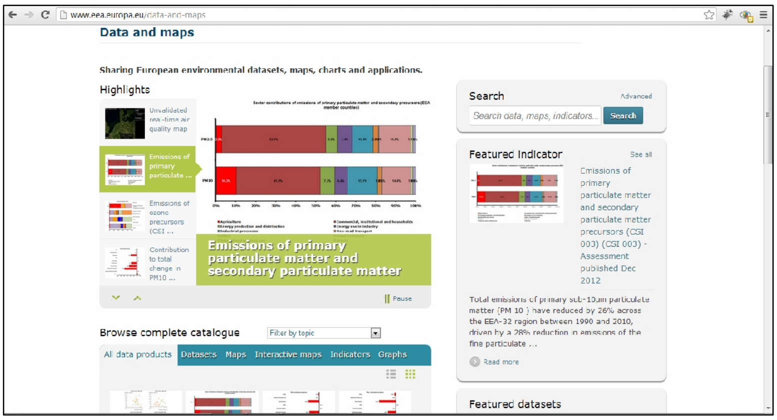


Figure 12: Data and maps site of European Environment Agency's geoportal in order to sharing European environmental datasets, maps, charts, and application.

1.4 Spatial Decision Support System (SDSS)

SDSSs were developed independently of one another and they are typically standalone systems incapable of sharing and reusing existing data and processing functions Zhang (2010). Since the early 1970s, spatial decision support systems (SDSS) technology and applications have evolved significantly. Many technological and organizational developments have exerted an impact on this evolution Shim (2002). At the beginning, SDSS had more limited database, modeling, and user interface functionality, but technological innovations have promoted far more powerful SDSS functionality. Based on advances in information technology, consists hardware, software, networks and communication equipments, provide potential solutions for the geospatial data accessibility. Advances in computational speed, storage ability, World Wide Web (WWW), software in general and specific GIS software, they provide great opportunities to develop Web-based Spatial Decision Support System (SDSS).

GIS tools are now commonly used in SDSS for many models such as hydrologic, landslide, flood, and soil erosion models. Otherwise GIS is also a main tool for data preparation. A Web-based SDSS needs computer (server) to host the SDSS application and it is connected to the user's computers (clients) by a computer network.

Although a web-based SDSS can combine several different components, in client interface of a web-based SDSS essentially is comprised of Hypertext Markup Language (HTML), JavaScript, Java applet, and the other internet protocols. Otherwise, Web-based SDSS can rely on Common Gateway Interface (CGI) Internet communication and server side operation. A conceptual web-based

SDSS framework using CGI has user interface HTML pages, a computational model, a Web-GIS application and a GIS database and file. This approach makes computation works in the server side and "light" client side, which is beneficial for data integrity, system management, communication speed and make HTML pages in order to send/receive through internet. Web-based SDSS user interfaces include menus, graphical maps, chart, table, control buttons and form. Client interface uses to control selections, input data and map display/queries, computation and visualization results, etc. The CGI works for communication including receiving data submitted, file management on the server side, running applications including computational models and writing results into file or standard out to the client browser.

A Web-based CGI performs Geospatial database manipulation including GIS data reading/writing (GIS software such as: GRASS, GDAL, OGR), querying (DBMS: PostgreSQL), image preparation (Web Map Server tools, such as: GeoServer, MapServer) for requests from the client based on HTML page standard. A computational model is a core application among web-based SDSS components (see Fig. 13).

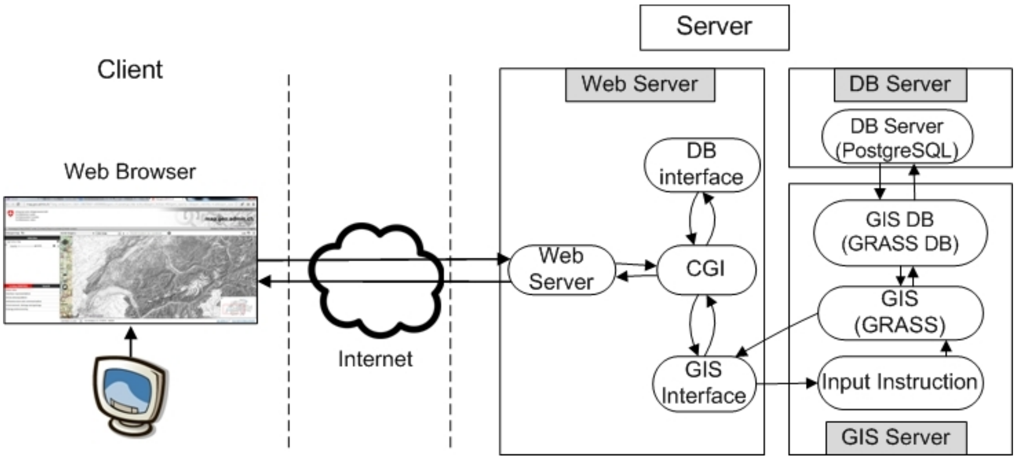


Figure 13: Components and Information work Flow presented by V.Raghavan et al (2002)

The main core of Web-based SDSSs is a framework which is supported by Web service-based decision-making systems Zhang (2010). The framework is based on independent OGC in order to provide geo-processing and geospatial data from various sources via service provider of OGC Web Processing Services (WPS) and service provider for geospatial data such as Web Feature Service (WFS), Web Map Service (WMS), Web Coverage Service (WCS). The OGC data services provide a basis to share spatial data with different data models and from different sources. Service generated from client (service client) send parameters to SDSS in order to display, manage data and access decision processing services.

Service client offers an easy/flexible interface to allow users to search and integrate geospatial data and geo-processing based on various service providers.

1.5 Motivations

Hydro-meteorological data is very essential for many fields of study such as: planning, development and management of the country's water resource and hazards. Precipitation data is the main factor of hydro-meteorological data, in addition, it is one of the major reasons for many types of hazards and major hazards frequently occur in developing countries.

National Centre for Hydro-Meteorological Forecasting, Vietnam Institute of Meteorology Hydrology and Environment (NCHMF), their aims is to meet all the requirements for the disaster prevention and preparedness, the socio-economic development and national security in Vietnam.NCHMF uses Mapper.js library (at http://www.netzgesta.de) in order to add automatic area highlighting to image maps and it supports requests to acquire hydro-meteorological data and weather forecast at daily, monthly intervals and season forecast (see at NCHMF). However, the NCHMF WebGIS cannot satisfy the requirements of many end-users and particular scientists because there is no method for users to access data in the past, and it is also missing statistical tools, flexibility search and query. Otherwise there is need to improve quality of base map such as land use/land cover, digital elevation model (DEM), slope and aspect map. To deal with this problem, it is necessary to improve quality and quantity of base map; moreover, it needs to build management, analysis, and visualization tools for historical hydrological and environmental data and so on.

Networks of hydro-meteorological stations are sparse in developing countries, where more pressing economic and social issues and the situation have is not improving appropriately. To mitigate natural disaster problems in the study are, one possibility compliant with the vision of 'Digital Earth' (Craglia et al., 2012), is to build up a Spatial Data Infrastructure (SDI) for monitoring the hydrological status. The study area is affected by natural calamities like floods, landslides, sea level rise, etc. The goal of the SDI is to make immediately and in near real time available the data for level of governments, commercials and none profit sectors, users and particularly for scientists. A geoportal typically provides access to geospatial data. On the opposite an intelligent geoportal provide access to resources including datasets (raster and vector data, imagery) and all the relevant Web GeoServices, which facilitate the discovery, display, editing and analysis of data. Nowadays, current advantages of computation speed, storage capacity, and specific software provide great opportunities to develop spatial decision support system (SDSS) which can be run through intelligent Geoportal. Due to the interoperable components of the system, it can be easily extended in order to integrate with other data or other models for processing the data. In its implementation the standard of Open Geospatial Consortium (OGC) are taken into account. Standards for Web Services (OWS) such as the well known Web Map Service (WMS), Web Feature Services (WFS) and Web Coverage Service (WCS) which allow the generation of maps made available respectively as images, geographic entities and coverage. These services are currently are widely diffused and successfully applied in many projects. Beside these "classic" services, the almost new OGC Sensor Web Enablement and its Sensor Observation Service (SOS) define opportunities for connecting in real-time heterogeneous sensors over the internet. This allows a better exploitation in a synergic way of the existing networks of remote, in-situ, and proximal sensors to monitor the status of the Earth. Furthermore, one of the most recent interoperability standards is OGC Web Processing Service. It provides standardized interface that facilitates the publishing of geospatial process, and discovery of and binding to those processes by clients. "Process" includes any algorithm, calculation or model that operates on spatially referenced data.

From all reasons above, in this thesis we not only present our intelligent geoportal for the study area (North of Vietnam) but also our DSS for monitoring satellite precipitation based on Web Services. DSS was developed as a tool that can compare directly or indirectly rain gauges (RG) and satellite precipitation (SP) products at daily, monthly and yearly intervals. Users may compare them at daily interval by interpolating RG data and associate those data with visualization of SP. Otherwise one can also request statistical tool for checking correlation and percentage of simultaneous rain or no-rain. At last, statistical tool and the Revised Universal Soil Loss Equation (RUSLE) are able to estimate accuracy of SP comparing with RG data.

Chapter 2: DATA COLLECTION

The chapter contains the description of the data used in the project.

The study area is in the northern part of Vietnam and it involves the Red River Basin. Red River (RR) is the second largest river in Viet Nam. It located in 20° North and 25°30 N, 100° E and 107°10E. The total area of the basin is approximated 169,000 km2, approximately 48% in China and 51.35% in Vietnam and the rest in Lao. The Red River, starting at China's Yunnan province, is about 1,200 kilometers long. The Red river (Hong River) in the Viet Nam part divided two parts, the first is upstream area, and it includes three upstream tributaries: Da, Thao, and Lo river (Fig. 14). is the part starting from Viet Tri city, includes Red River Delta, is a flat area, and total area is about 15,000 square kilometers.

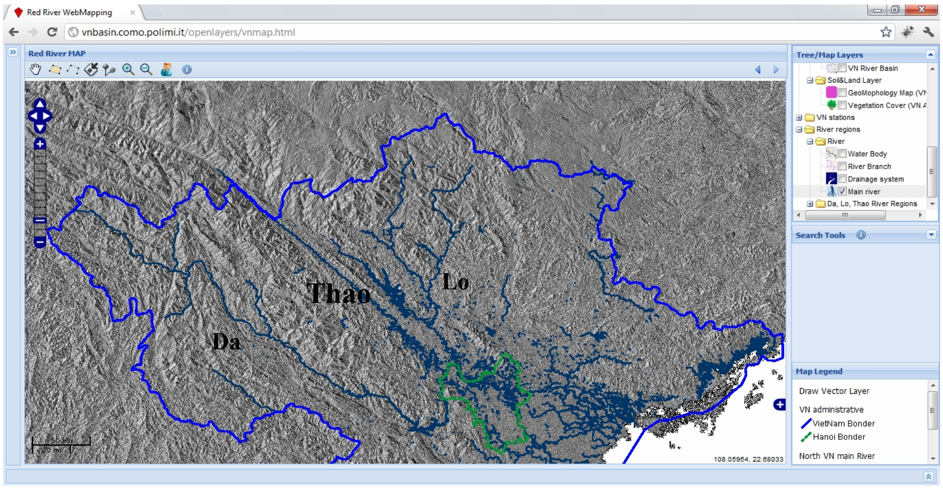


Figure 14: Web-GIS showing upstream tributaries including Da, Thao, and Lo River in dark blue, base map is aspect image.

There are several types of data which have been used for this research work. The data used can be subdivided into two distinct groups, the former group can be named the remote sensed data (RS) and GIS (Geographic Information System) data, the second group consists of measured data. Remote sensing and GIS data are given in vector, or raster format derived from various sources. On the opposite, measured data was issued by the Institute of Water Resources Planning of Vietnam (WRP) will be used in this research.

The data used can be summarized as shown in table (3), (4) for first group and (Table 5) for second group.

The main characteristic we have considered for the first group are: the date of acquisition, the frequency band (only for imagery), the ground resolution (in meters) and the source in (Table 3). In (Table 4) shows the type of data, thematic maps, date of acquisition, and resources.

For the second group we considered: the period of collection (starting and ending time), the number of stations, the format those data have been made available and again the source.

Remote Sensed Data (group 1.a)				
GIS Data	**Data of acquisition**	**Band/Color**	**Resolution**	**Source**
Landsat TM	1992,1993,1994 1989, 2004,2006	Multi-spectral	30m	http://glcf.umiacs.umd.edu/
Landsat ETM+	2000,2001,1999	Multi-spectral Panchromatic	30m 15m	http://glcf.umiacs.umd.edu/
ASTER GDEM	Released by ASTER GDEM, October, 2011	Short-way infrared	30m	www.gdem.aster.ersdac.or.jp
SRTM	Last update: August 2008	C-band, X-band	90m	srtm.csi.cgiar.org
Rainfall data	2004-2007	IR image brightness temperature	0.25^{o}	PERSIANN

Table 3: Data used for the purpose of study: Remote sensed data.

GIS Data (group 1.b)			
Name/Type	Maps	Date	Resource
Points	Stations	2001	WRP
	Points of interest	2001	Atlas-Vietnam
Polylines/Lines	Rivers, Drainage System, Main Rivers, River Branch	2001	WRP
	Roads	2004	Atlas-Vietnam
	Bounders	2000	
	Rainfall iso-line	2000	WRP
	Contour line	2000	Atlas-Vietnam
Polygons	Provinces	2000	Atlas-Vietnam
	Municipalities	2000	Atlas-Vietnam
	River Basins	2000	Atlas-Vietnam
	Soil map	2004	Allas-Vietnam
	Forest map	2004	Atlas-Vietnam

Table 4: Data we used based on data issued form two sources, WRP in 2001 and Atlas of Vietnam in 2004.

Measuring data				
Type	Info			
	Year	Num of Stations	Data type	Sources
Rainfall	1956-2004	19	xls, file	WRP
Evaporation	1961-2002	14	xls, file	WRP
Discharge	1956-2004	10	xls, file	WRP
Water level	1956-2005	7	xls, file	WRP
Inflow & Reserior	1956-2008	1	xls, file	WRP

Table 5: Data used for the purpose of study: group 2

2.1 Remote sensed data: group 1

Remote sensing data can support human for significant recognition capability of land-use land-cover, and many useful applications in field of geology hazards, urban planning, environment, and so on. One of the most powerful tool and low-cost possibility for studying above problems is using free data sources. Landsat data is one of the free data sources and it can be freely download via internet. Global Land Cover Facility (GLCF) is one of the free data sources supported by a center for land cover science with a focus on research using remote sensed satellite data and products to access land cover change from local to global system. Otherwise, another type of satellite derived information which supports the people for getting information about Earth's terrain and creating thematic maps such as landform, geomorphology, volcanic and so on, it is Digital Elevation Model (DEM). DEM is a digital model or 3-D representation of terrain's surface; all global datasets are captured by satellite as Shuttle Radar Topography Mission (SRTM) or ASTER Global Digital Elevation Model (ASTER GDEM). From DEM we can use surface analysis tools to create hillshade, slope, aspect, contours map or others 3-D analysis. Finally, in this research work, we use a satellite-based precipitation measurement named PERSIANN (Precipitation Estimation from Remote Sensed Information using Artificial Neural Networks), PERSIANN is global precipitation from satellite image using artificial neural networks issued by Center for Hydrometeorology Remote Sensing, University of California, Irvine.

2.1.1 Landsat Data

Landsat is the United States's oldest land-surface observation satellite system. The diversity of Landsat applications makes it unique among Earth observation satellites. Images acquired by Landsat satellites were used to produce the first composite multi-spectral mosaic of the 48 contiguous United States. They have been used to monitor timber losses in the U.S. Pacific Northwest, map the extent of winter snow pack, and measure forest cover at the state level. In addition, Landsat has been used to locate mineral deposits, monitor strip mining, and assess natural changes due to fires and insect infestations.

NASA launched the first satellite in the Landsat series (originally called the Earth Resources Technology Satellites) in 1972. Landsat 5 was launched in March 1985 and it is still returning images. NASA was named the lead agency working with NOAA and the U.S Geological Survey (USGS) for Landsat 7, Landsat Thematic Mapper (TM). The instrument on board Landsat 7 is the Enhance Thematic Mapper Plus (ETM+). ETM+ is a passive sensor that measures solar radiation reflected or emitted by the Earth's surface. The instrument has eight bands sensitive to different wavelengths of visible and infrared radiation and has better resolution in the thermal infrared band than the Thematic. The resolution for the TM sensor is shown below (in Table 6):

Wavelength	Resolution	Meters	Spectral Color
Band 1	0.45-0.52	30	Blue
Band 2	0.52-0.60	30	Green
Band 3	0.63-0.69	30	Red
Band 4	0.76-0.90	30	Near Infrared
Band 5	1.55-1.75	30	Mid-Infrared
Band 6	10.40-12.50	120	Thermal Infrared
Band 7	2.08-2.35	30	Mid-Infrared

Table 6: List of seven bands of Landsat TM data.

Band 1- Visible blue, useful for mapping coastal water areas, differentiating between soil and vegetation, forest type mapping, and detecting cultural features.

Band 2- Visible Green, corresponds to the green reflectance of healthy vegetation. Also useful for cultural feature identification.

Band 3- Visible Red, useful for discriminating between many plant species. It is also useful for detecting soil boundary and geological boundary delineations as well as cultural features.

Band 4- Near-infrared band is especially responsive the amount of vegetation biomass present in a scene. It is useful for crop identification and emphasizes soil/crop and land/water contacts.

Band 5- Mid-infrared, this band is sensitive to the amount of water in plants, which is useful in crop drought studies applications, and for locating thermal pollution. It can also be used to locate geothermal activity.

Band 6- Thermal-infrared, this band is useful for vegetation and crop stress detection, heat intensity, insecticide application, and for locating thermal pollution. It can also be used to locate geothermal activity.

Band 7- Mid-infrared, this band is important for the discrimination of geologic rock type and soil boundaries, as well as soil and vegetation moisture content.

Landsat imagery can apply for many fields of study e.g: help geographers evaluate how different urban planning programs, population growth effects and land use changes, performing multi-year studies in a basin to determine the year-to-year changes in deforestation rates; thermal band on Landsat can detect crop health by seeing plants transpire or lose moisture through their leaves. Landsat TM data is a powerful tool for studying land use and land cover, flood map, water land, non-water land, urban area, agricultural area and so on.

2.1.1.1 Image collection

Data is available to free download at http://glcf.umiacs.umd.edu, GLCF website. The original projection is UTM with datum WGS84. The table below shows the entire image acquired for whole study area. We concentrate on getting whole Landsat TM images acquired during flood season or dry season. In the North of Vietnam flood season usually starts from May-Jun to September. A difference of water levels between highest in the flood season and lowest in dry season is significance for studying flood. Example highest water level in the Hong River at HaNoi station in 1996 is 12.35m and lowest water level in the Hong River at HaNoi station is approximate 1m high from 10/2009 to 2/2010. In the Figure 15 and Table 7 present data available to download in study area.

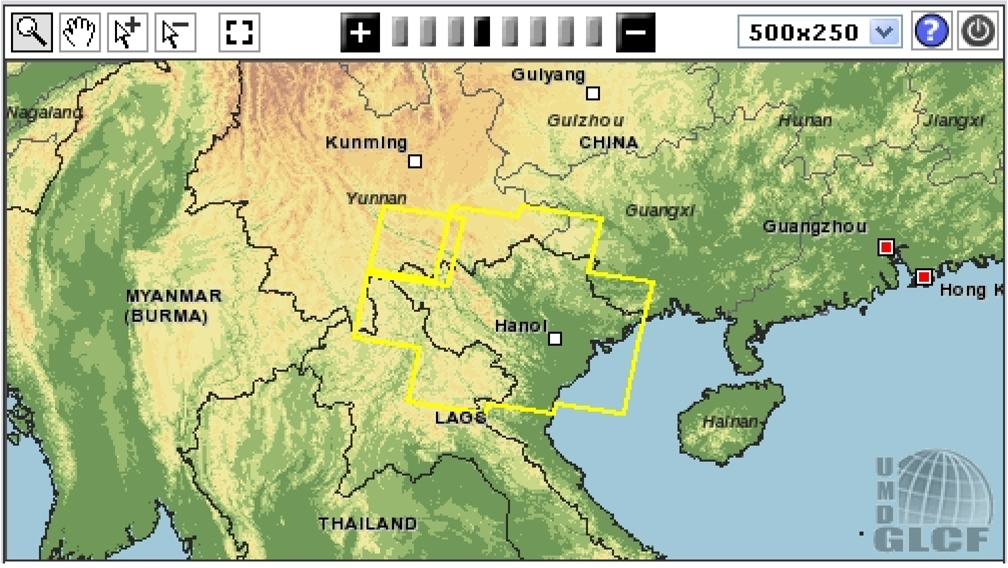


Figure 15: Rectangle is study area and all the paths and rows of need data.

Path	Row	Date	Dataset		Date
126	45	2001-09-29	ETM+	TM	2004-12-02
126	46	2001-09-29	ETM+	TM	1989-11-23
127	44	2000-04-11	ETM+	TM	1994-10-27
127	45	2000-04-11	ETM+	TM	1994-10-27
127	46	2000-04-11	ETM+	TM	1993-12-27
128	44	1999-12-27	ETM+	TM	1990-08-04
128	45	1999-12-27	ETM+	TM	1993-01-02
128	46	1999-12-27	ETM+	TM	2006-12-22
129	44	1999-11-02	ETM+	TM	2006-05-19
129	45	1999-11-02	ETM+	TM	2002-02-09

Table 7: Number of row and path cover all study area responding rectangular in the (figure 15).

2.1.1.2 Land use/cover

Landsat TM imagery can be used for creating land use/cover maps; there are many research work which concentrate in detecting land use and land cover based on Landsat TM images. Landsat TM can be used to map change of urban base on classification comparison of Landsat TM (Ursa et al 2011). Wetland area, flood area are detected based on reflectance features of water versus non-water target on pair of Landsat 7 images (Ho et al 2010). In this work, we used a simple method which is a combination of bands 4, 3 and 2. This method is called “False Color” Composite [NASA, 2011]. This very popular band combination is useful for vegetation studies, monitoring drainage, seeing soil patterns, and determining various stages of crop grow, a result looks similar to color-infrared (CIR) area photography. The bands are displayed in Red, Green and Blue channel, with these band combination vegetation appears in shades of red, urban areas are cyan blue, and soils are shown in shades of browns (darker shades indicate higher moisture levels). Generally, bright reds indicate broadleaf vegetation and vigorously growing vegetation, while lighter reds signify grasslands or sparsely vegetation areas.

The primary advantage of the false-color composite method is that spatial relationships of the features in the constituent images are perceived directly (Toby Kurk al 2010). In cases where information is convoluted across the constituent images the color-composite method confers an additional advantage since the combined information from all three channels results in a significant aid to perception and interpretation.

In the flow chart below (Fig. 16) presents a based map of Web-GIS generated from “false-color” composite of Landsat TM.

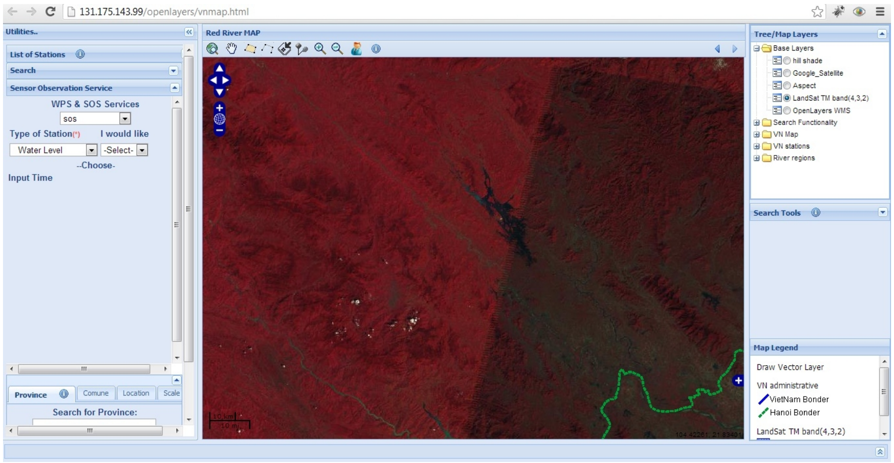


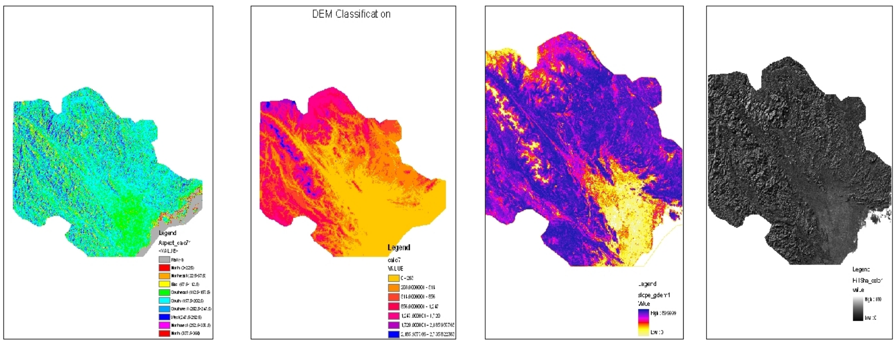
Figure 16: Base map of WebGIS is an image generated from false-color composite of Landsat TM.

2.1.2 ASTER

ASTER (Advanced Space borne Thermal Emission and Reflection Radiometer) is an image instrument flying on Terra, a satellite launched in December 1999 as part of NASA’s Earth Observing System (EOS). ASTER is a cooperative effort between NASA, Japan’s Ministry of Economy, Trade and Industry (METI) and Japan’s Earth Remote Sensing Data Analysis Center (ERSDAC). ASTER is being used to obtain detailed maps of land surface temperature, reflectance and elevation. The ASTER GDEM is generated with spatial resolution of about 30m; the ASTER GDEM contains highest resolution of freely available DEMs for users in internet. The Aster GDEM covers land surfaces between 83°N and 83°S with estimated accuracies of 20 meters at 95% confidence for vertical data and 30 meters for at 95% for horizontal data. Furthermore it is aimed to be used as a platform for analysis of the data in the fields of disaster monitoring (e.g., volcanic of flood hazard map), hydrology (e.g., water resource management), energy (e.g., oil resource exploration), and Stereoscopic visualization (e.g., for Bird's-eye views and flight simulations) (ERSDAC, 2010).

DEM is generated from ASTER satellite data; it could be useful for studying geomorphologic mapping (ERSDAC el al 2003). Using DEM is a platform for simulating lava flow, volcanic hazard maps, simulating a process of rainfall, catchment, runoff and inundation to generate "flood maps", otherwise using DEM to calculate catchment area, to measure strike and dip from DEM (ERSDAC, 2011). DEM offers the most common method for extracting vital topographic information and it is also accurately represent terrain. Using DEM generated from ASTER satellite data is useful to describe classification of elevation and also 3D views such as hillshade, aspect, slope, contour maps.

We used the 18 sub-images of ASTER data (from ASTGTM_N20E103 to ASTGTM_N20E107, from ASTGTM_N21E103 to ASTGTM_N21E107, from ASTGTM_N22E103 to ASTGTM_N22E106 and from ASTGTM_N23E103 to ASTGTM_N23E106) to have general view of topographic map, 3D views and furthermore to generate geomorphology of the study area. Study area located in the terrain includes towns, cities, agriculture lands, mountain areas, rivers, lakes, river basins, drainage systems, roads, dams, and others type of land form. Otherwise, it contains many types of geology hazards such as landslide, erosion, weathering, hydro-logical problems, and so on.



(a) (b) (c) (d)

Figure 17: (a) Classification of ASTER DEM, (b) Slope map, (c) Aspect, (d) Hill shade

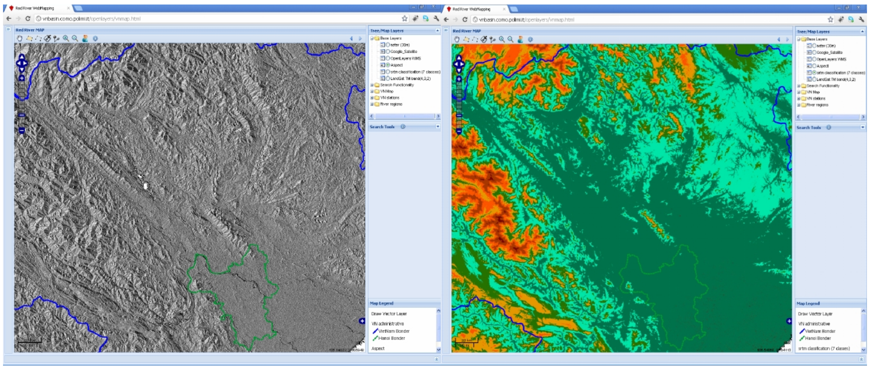


Figure 18: WebGIS showing hill-shade (left) and DEM classification (right)

In Figure 17 (a) DEM was classified to 7 classes, the landform are divided into mountain and plains, the surface of the ground can be classified into flat surfaces, gentle slopes, cliffs, and on the mountain area was divided low plateaus, plateaus, high altitude plateaus, and high mountains. (b) Slope map corresponds to DEM, (c) topography can be generalized into eight aspects, and this may to identify landform feature, (d) presents where is shadow exist in the DEM, depending on the origin (direction) of the light source and the elevations that exist. Figure 18 shows Web-GIS's interface present based map such as hill-shade in left and DEM classification in right.

2.1.3 Shuttle Radar Topography Mission (SRTM)

The Shuttle Radar Topography Mission (SRTM) obtained elevation data on near-global scale to generate the most complete high-resolution digital topographic database of Earth. SRTM is a joint project between the National Aeronautics and Space Administration (NASA) and the Department of Defence's National Geospatial – Intelligence Agency (NGA). The NASA SRTM has provided digital elevation data (DEMs) for over 80% of the globe. This data is currently distributed free of charge by U.S Gelogical Survey (USGS) and is available to download online. The SRTM data is available as 3 arc second (approx. 90m resolution) DEMs, 1 arc second (approx. 30m resolution) for some part of the world. The vertical error of the DEMs 90m resolution is reported to be less than 16m. The data currently being distributed by NASA/USGC contains "no-data" holes where water or heavy shadow prevented the quantification of elevation. These are generally small holes, which nevertheless render the data less useful, especially in fields of hydrological modeling.

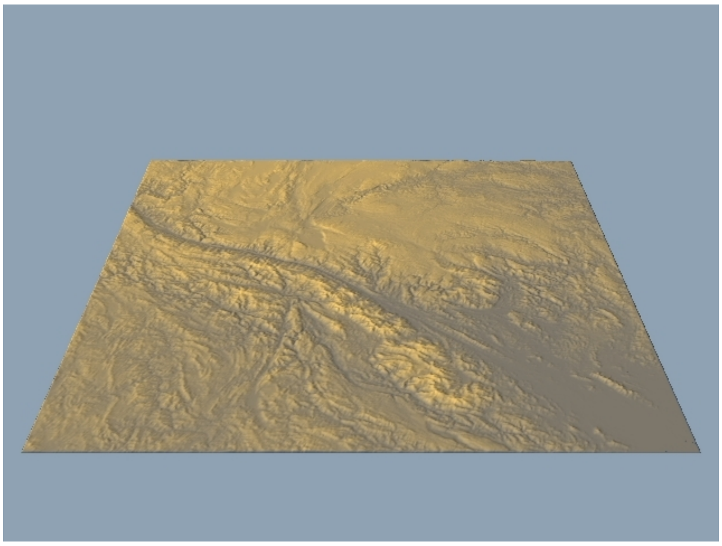


Figure 19: 3D scene of Red-River Delta and Basin was generated by 3DEM software version 20.7.

Some of data product applications of SRTM mission are Scientific Applications such as geology, geophysics, earthquake research, volcano monitoring, hydrologic modeling, co-registration of remotely-acquired image data. Civilian Applications: enhanced ground proximity, warning systems for aircraft, civil engineering, and land use planning, Line of sight determination for communications (e.g. cell phones). Military Applications: flight simulators, logistical planning, trafficability, missile and weapons guidance systems battlefield management, tactics. Example SRTM for study area presents in (Fig. 19)

Base on landform study, this method using DEM, the land form in the study area can be roughly divided into mountain and plain areas, as the (figure 20.a) presents classification of DEM divided to seven classes such as flat surfaces, gentle slopes, cliffs, and mountain area is divided low plateaus, plateaus, high altitude plateaus.

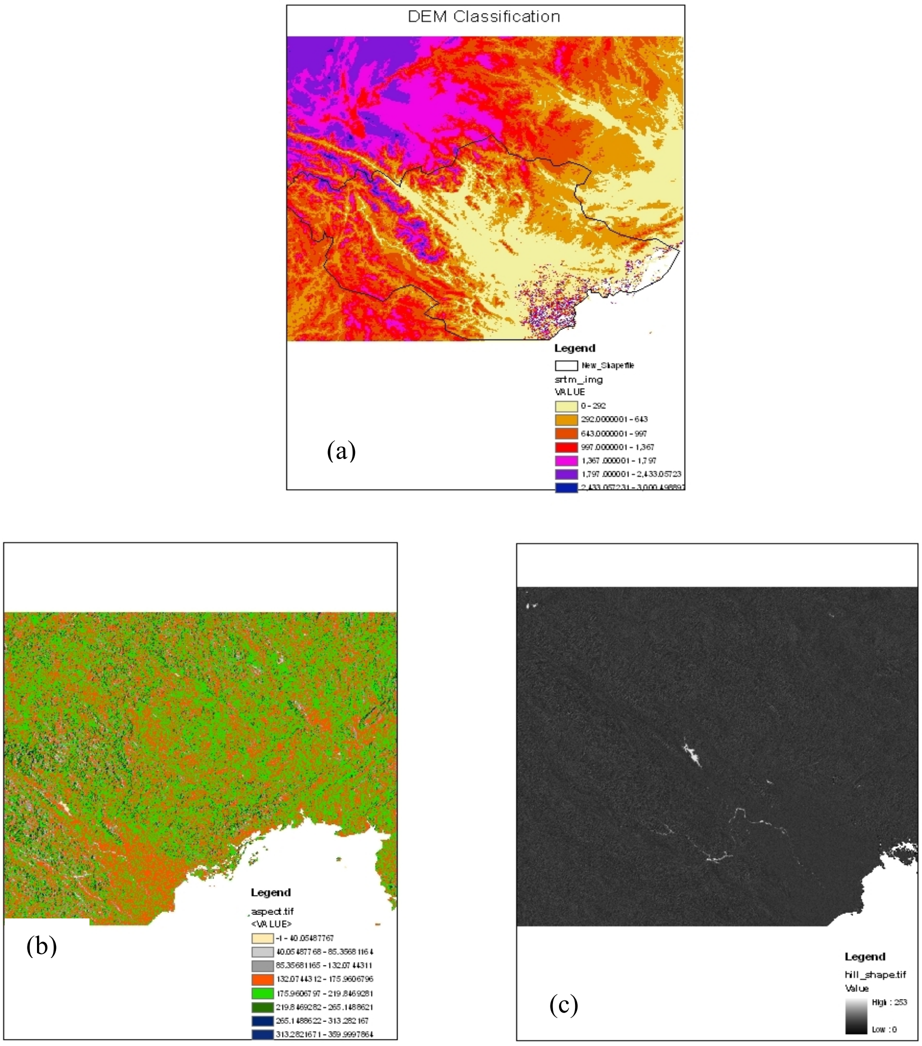


Figure 20: DEM generation form SRTM satellite data (a) DEM Classification, (b) aspect map, (c) hillshade map

2.1.4 Precipitation estimation from Satellite Information using Artificial Neural Network (PERSIANN)

Recent development of satellite remote sensing techniques provides a unique opportunity for better observation of precipitation for region where ground measurement is limited, PERSIANN, provides near global rainfall estimates at hourly 0.25x0.25.

The Current operational PERSIANN system uses neural network function classification/approximation procedures to compute an estimate of rainfall rate at each 0.25° x 0.25° pixel of the infrared brightness temperature image provided by geostationary satellites. An adaptive training feature facilitates updating of the network parameters whenever independent estimates of rainfall are available. The PERSIANN system was based on geostationary infrared imagery and later extended to include the use of both infrared and daytime visible imagery. Rainfall product covers 50°S to 50°N globally.

The system uses grid infrared images of global geosynchronous satellites (GOES-8, GOES-10, GMS-5, Metsat-6, and Metsat-7) provided by CPC, NOAA to generate 30-minute rain rates are aggregated to 6-hour accumulated rainfall. Model parameters are regularly updated using rainfall estimates from low-orbital satellites, including TRMM, NOAA-15, -16, -17, DMSP F13, F14, F15 (CHRS).

About ANN model, at the first, the traditional approach for estimating precipitation from satellite information based on statistical relationship between Infrared measurements of top cloud brightness temperature and mean rainfall rate. In fact, many factors including such as cloud type and local atmospheric condition. To produce precipitation estimates at finer temporal and spatial resolutions, it is necessary to develop new models relating satellite information, the models should satisfy conditions such as: Extract information from multiple input features (variables) those are related to precipitation; perform sophisticated classification and mapping in a multidimensional input-output space; rapid process the large amounts of data; be able to automatic update itself whenever ground-base or other sources of data are available. The model uses methods that synthesize computed-based artificial intelligence techniques with the theory of precipitation process and practical experience. Artificial neural network (ANN) has potential to resolve all of the above requirements. The ANN structure allows the mapping of any complicated nonlinear function to an arbitrary degree of accuracy through intelligently "learning" these functions using an automatic "training process".

2.1.4.1 Using Rapid Access to Country Data

Use this tool to access recent precipitation and the climatologically aridity images from PERSIANN global precipitation data set. Full monthly archives of PERSIANN's final 0.25o are available at 6 hourly data and daily data, the data format with 4-byte, 0.25° x 0.25° resolution, spatial coverage 60^o to – 60^o lat and 0 to 360^o long. Otherwise, data are available for our study area from 2000 up to 2007.

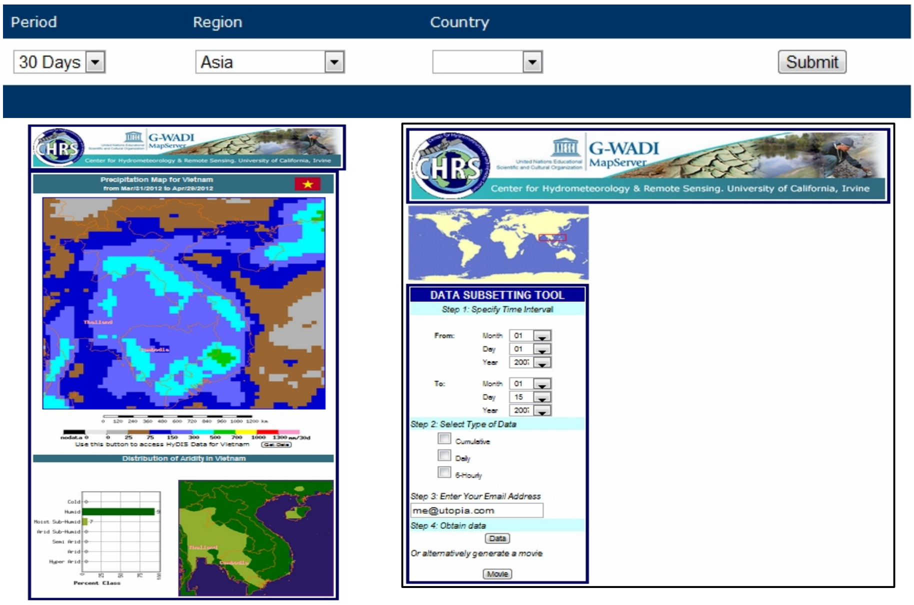


Figure 21: Country tool's interface to access images for precipitation accumulation for PERSIANN global

2.1.4.2 Data Extracting

We can use country tool (Fig. 21) to access images for recent precipitation accumulation for PERSIANN global precipitation data set. Start by selecting the desired accumulation period, and then select region and country users can obtain a month of data for their country in daily or 6 hourly data, in our research field, we are interested in daily precipitation totals. The data download it should be in tar file, after file was decompressed, the files include two files format: YYDOY.gif and YYDOY.asc. YYDOY.asc with each its each cell was geo-referenced in WGS84 reference coordinate system (see Fig. 22).

Where YY 2 digit year starting from 00 for 2000
DOY: Day of year, (to translate to month/day, see at:
http://www.esrl.noaa.gov/gmd/grad/neubrew/Calendar.jsp

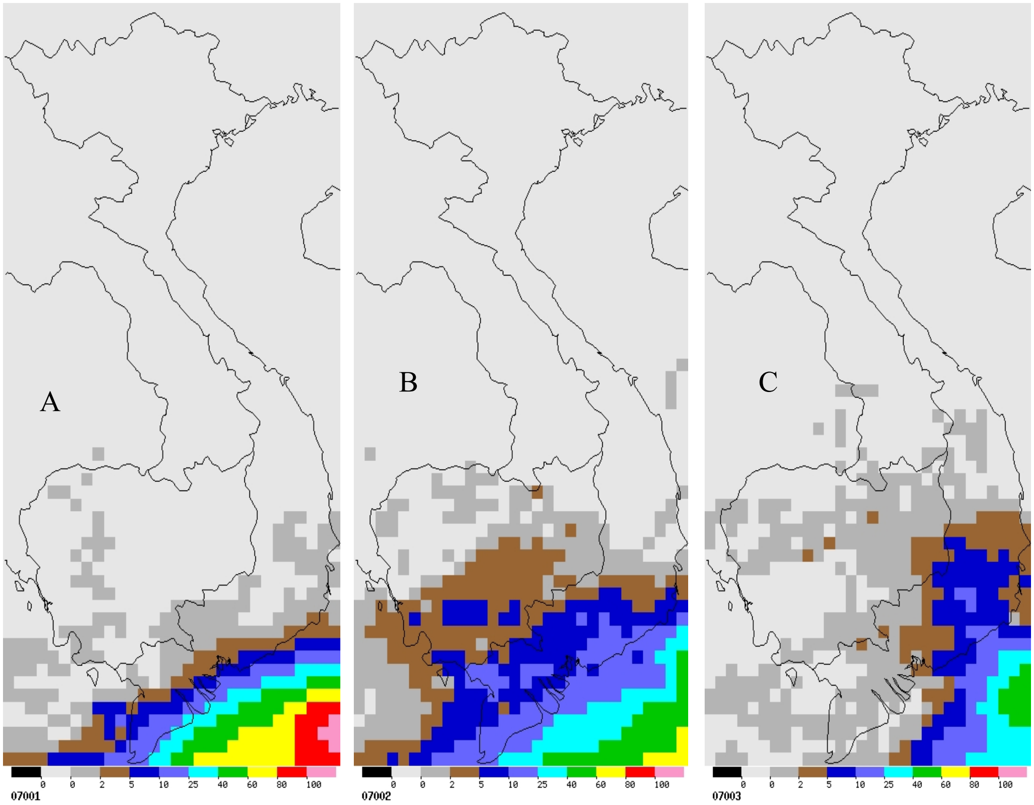


Figure 22: Precipitation intensity (mm/day) A (07001) is measured in 1st Jan 2007 B (07002) in 2nd Jan 2007 and C (07003) in 3rd Jan 2007.

An image covers total land, the area of land located at lower-left (102.125, 8.625) upper-left (102.125, 23.625) and lower-right (109.625, 8.625), upper-right (109.625, 23. 625). An image was divided by 30 cols by 60 rows with 0.25o x 0.25o resolution. The format of *.asc file of PERSIANN presents as below example in case of 07001.asc file

ncols 30
nrows 60
xllcorner 102.125000
yllcorner 8.625000
cellsize 0.2500000000

nodata_value -9999.0

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	…… 0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	…… 0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	…… 0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	….. ..0.00	0.00
	………..								
1.32	0.89	0.88	0.23	1.54	2.41	4.07	3.32	……. 8.95	10.14
2.36	2.34	2.86	3.12	5.89	6.14	8.32	7.90	…....11.77	14.51

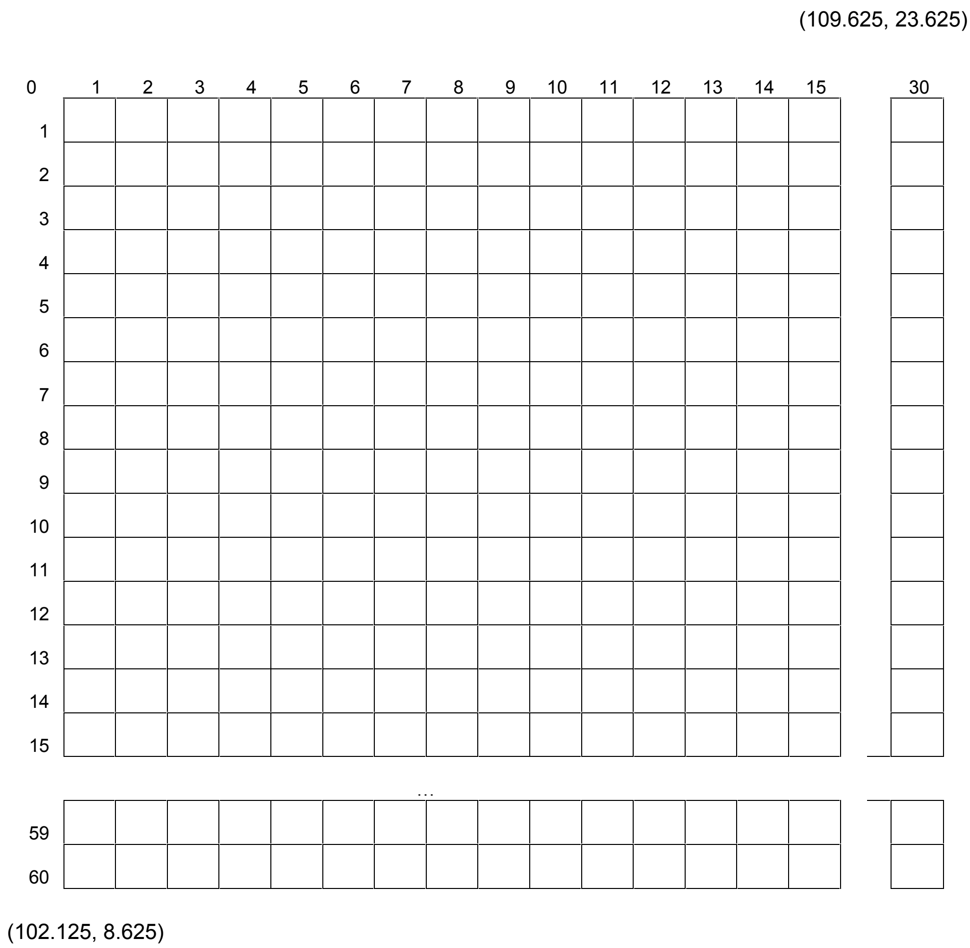


Figure 23: PERSIANN data product includes 30 x 60 cells, 0.25ox0.25o resolution, grid covers whole area starting from (102.125, 8.625) to (109.625, 23.625)

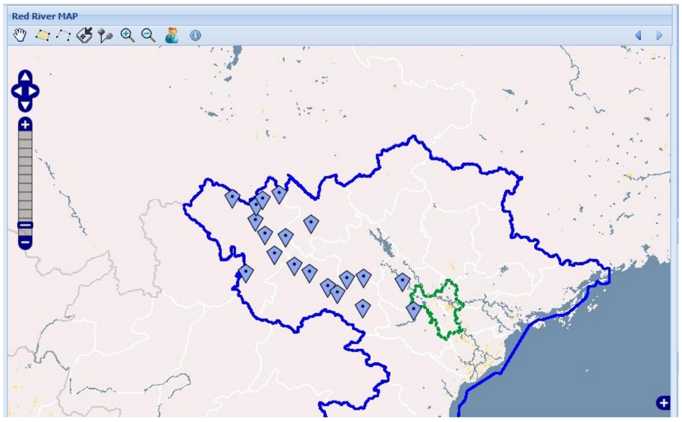


Figure 24: WebGIS showing map of meteorology stations on ground-based, rainfall was measured by gauges at 19 stations

Nº	Station's Name	Location	ID	PERSIANN	
		Lat	Long		Year
1	Dien Bien	103.01758	21.33223	DB	2000-2007
2	Nam Muc	103.283333	21.866667	NM	2000-2007
3	Than Uyen	103.916667	22.016667	TY	2000-2007
4	Nam Giang	103.16406	22.25946	NG	2000-2007
5	Tuan Giao	103.416667	21.583333	TG	2000-2007
6	Hoa Binh	105.333333	20.816667	HB	2000-2007
7	Lai Chau	103.15	22.05	LC	2000-2007
8	Muong te	102.833333	22.366667	MT	2000-2007
9	Sin Ho	103.25	22.35	SH	2000-2007
10	Quynh Nhai	103.566667	21.833333	QN	2000-2007
11	Son La	103.9	21.333333	SL	2000-2007
12	Tam Duong	103.483333	22.416667	TD	2000-2007
13	Phu Yen	104.65	21.266667	PY	2000-2007
14	Bac Yen	104.416667	21.25	BY	2000-2007
15	Co Noi	104.15	21.133333	CN	2000-2007
16	Moc Chau	104.633333	20.85	MC	2000-2007
17	Thuan Chau	104.6833333	21.43333	TC	2000-2007
18	Yen Chau	104.283333	21.05	YC	2000-2007
19	Thanh Son	105.183333	21.216667	TS	2000-2007

Table 8: Table of rainfall stations (RFs) location, the 6th column showing available years to download from PERSIANN product's store.

All RFs in (table 8) are covered by a rectangle located at lower-left (20.8167, 103.0176) and upper-right (22.4167, 105.33333) in (Fig. 23). It means whole 19 locations (Fig. 24) in group of RFs in the ground-based locate inner boundary of PERSIANN data product shown in (Fig. 25).

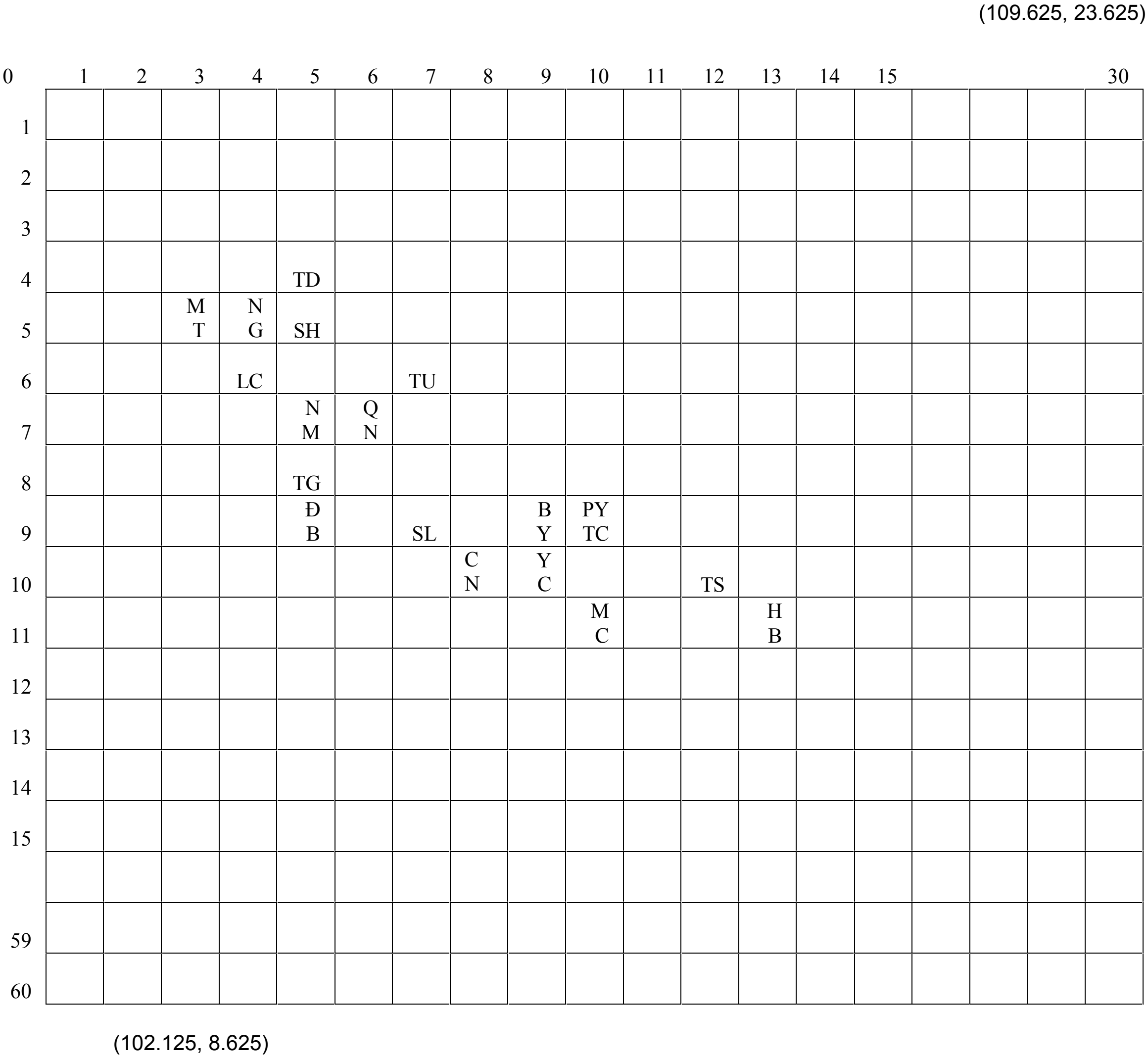


Figure 25: PERSIANN grid (30 x 60 cells) Overlay 19 locations of RFs in study area. PERSIANN data product's grid covers whole area from (102.125, 8.625) to (109.625, 23.625).

Data for whole 19 stations can be extracted form grid of PERSIANN product by using (Eq 1.1), we estimate only overlay or nearest pixel located in grid of PERSIANN product where those cells correspond to each location's station in the group of 19 RFs, which reads:

$$M = \left| 23.625 - RFS_{lat}(i) \right| \qquad 0 \le M \le 60,\; i = \overline{1,..19}$$
$$N = \left\lceil 109.625 - RFS_{long}(i) \right\rceil \qquad 0 \le N \le 30 \qquad (0)$$

$\lceil\ \rceil$ is a ceiling, RFSlat(i) and RFSlong (i) are a latitude of the number i of rainfall station and a longitude of the of the number i of rainfall station. Otherwise, 23.625 is the max latitude and 109.625 is the max longitude of PERSIANN product in this study area.

For extracting the data from PERSIANN product we following the methodology chart as follow (Fig. 26):

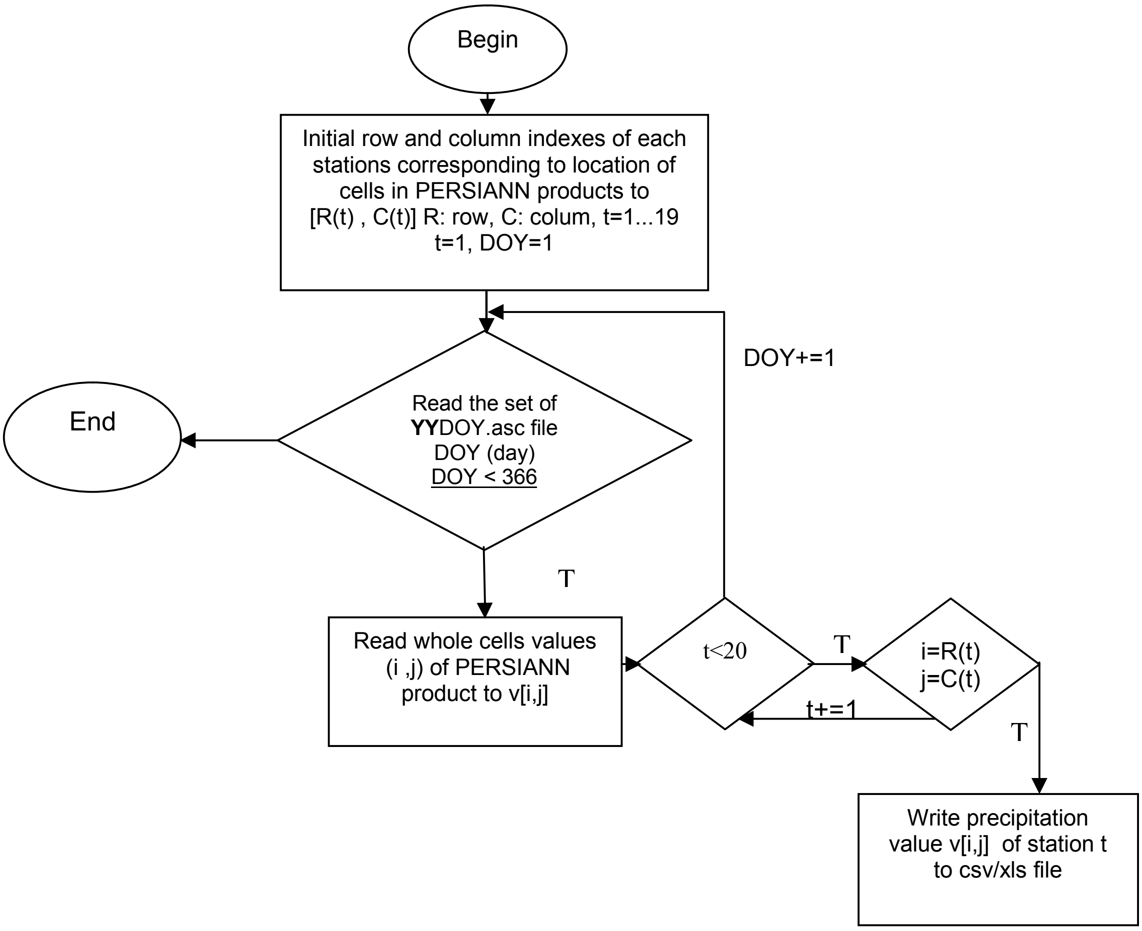


Figure 26: Flow chart of algorithm for extracting PERSIANN product

The methodology above was implemented in Python, a result after extracting the data what we need will be stored in csv file or xls file (see example in Fig. 27).

DienBien	NamMuc	ThanUyen	NamGiang	TuanGiao	HoaBinh	LaiChau	MuongTe	SinHo	QuynhNhai
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.24	1	0.3	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

SonLa	TamDuong	PhuYen	BacYen	CoNoi	MocChau	ThuanChau	YenChau	ThanhSon	DAY
0	0	0	0	0	0	0	0	0	01/01/2007
0	0	0	0	0	0	0	0	0	02/01/2007
0	0	0	0	0	0	0	0	0	03/01/2007
0	0	0	0	0	0	0	0	0	04/01/2007
0	0	0	0	0	0	0	0	0	05/01/2007
0	0	0	0	0	0	0	0	0	06/01/2007
0	0.24	0	0	0	0	0	0	0	07/01/2007
0	0	0	0	0	0	0	0	0	08/01/2007
0	0	0	0	0	0	0	0	0	09/01/2007

Figure 27: showing xls file of the result after extracting data from PERSIANN product of 15 days starting from 01/01/2007 to 15/01/2007.

2.2 Measuring Data: group 2

2.2.1 Data Source

The precursor to the WRP (Institute of Water Resources Planning) was professional office with main functionality is to control flood and to develop Hong River, it was established in 1961.At the beginning, WRP was an agency with professional functionality to assist Hong River Commission belong to Ministry of Water Resources of Vietnam in previous. At the present, the Institute has tasks to do water resources development planning in the country for the purpose of regulating water use, protecting and sustainably developing water resources and water environment, for further more information about WRP http://www.iwarp.org.vn/en/ .

2.2.2 Measurement station system

There are three kinds of station: hydrological, meteorological and hydro-meteorological stations. Both hydrological and meteorological data were collected starting from the end of 18th century in Ha Noi, Hoa Binh, Lai Chau, Yen Bai, and other places in tributaries. However, they were not continuous data. The measurement system was synchronized from 1956, but a lot of station have collected discontinue measurements due to Sino-Vietnamese Wars (Third Indochina War). Whole information was observed with daily resolution. In the following figures (From Fig. 28 to Fig. 30) the locations of the different stations is shown. In the (Table 9 and Table 11) presents whole data available and data format in domain of this research. (Figure 31) and (32) presents Web-GIS's based map of all hydrological and meteorological station of study area.

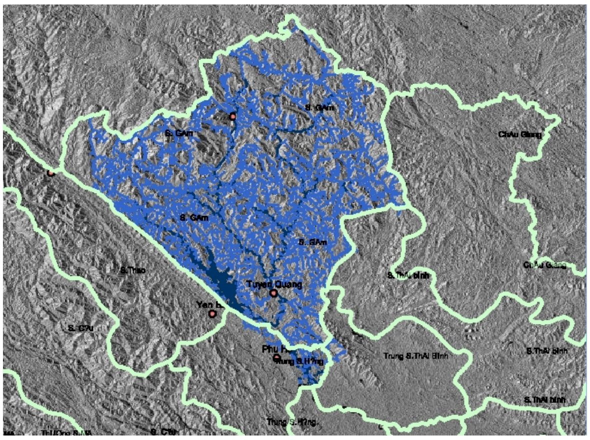


Figure 28: Map hydrological stations along Lo-River

No	NAME	X	Y
1	Lao Cai	103.950	22.500
2	Yen Bai	104.867	21.700
3	Ha Giang	104.983	22.817
4	Tuyen Quang	105.217	21.817
5	Phu Ho	105.233	21.450

Table 9: showing list of station in the (Figure 28).

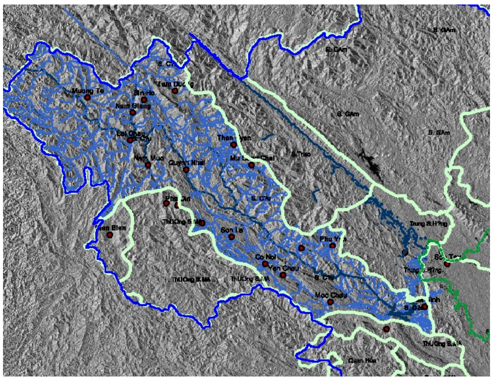


Figure 29: Map of hydrological stations along Da-River

No	NAME of STATION	X	Y
1	Dien Bien	103.000	21.350
2	Muong Te	102.833	22.367
3	Tuan Giao	103.417	21.583
4	Sin Ho	103.250	22.350
5	Lai Chau	103.150	22.050
6	Pha Din	103.500	21.567
7	Tam Duong	103.483	22.417
8	Son La	103.900	21.333
9	Quynh Nhai	103.567	21.833
10	Yen Chau	104.283	21.050
11	Moc Chau	104.633	20.850
12	Phu Yen	104.650	21.267
13	Bac Yen	104.417	21.250
14	Thuan Chau	103.683	21.433
15	Co Noi	104.150	21.133
16	Than Uyen	103.917	22.017
17	Mu Cang Chai	104.833	21.083
18	Thanh Son	105.183	21.217
19	Son Tay	105.500	21.133
20	Mai Chau	105.050	20.650
21	Hoa Binh	105.333	20.817
22	Nam Muc	103.283	21.867
23	Nam Giang	103.283	21.866

Table 10: showing list of station in the (Figure 29)

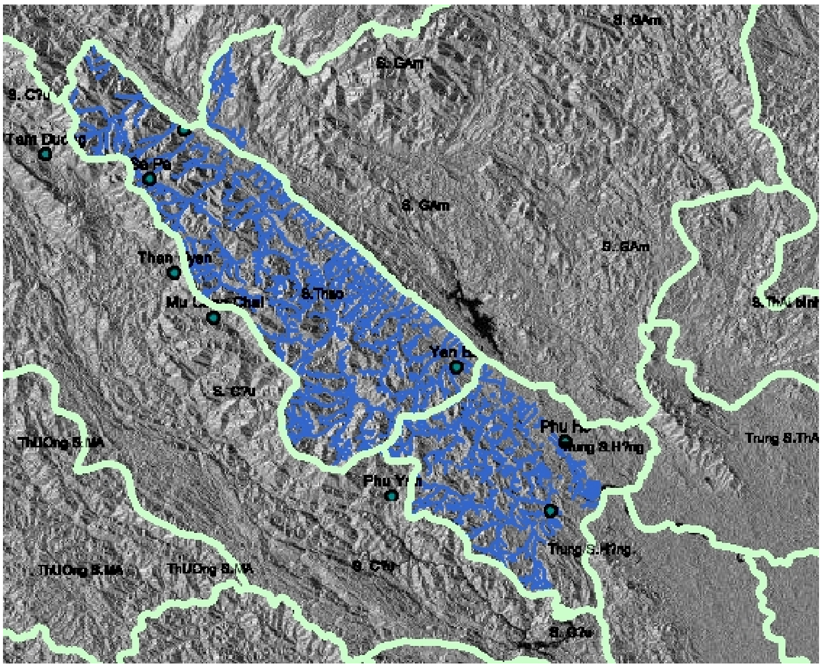


Figure 30: Map of hydrological stations along Lo-River area

No	NAME	X	Y
1	Tam Duong	103.483	22.417
2	Phu Yen	104.650	21.267
3	Sa Pa	103.833	22.333
4	Lao Cai	103.950	22.500
5	Than Uyen	103.917	22.017
6	Mu Cang Chai	104.833	21.083
7	Yen Bai	104.867	21.700
8	Phu Ho	105.233	21.450
9	Thanh Son	105.183	21.217

Table 11: showing list of station in the (figure 30)

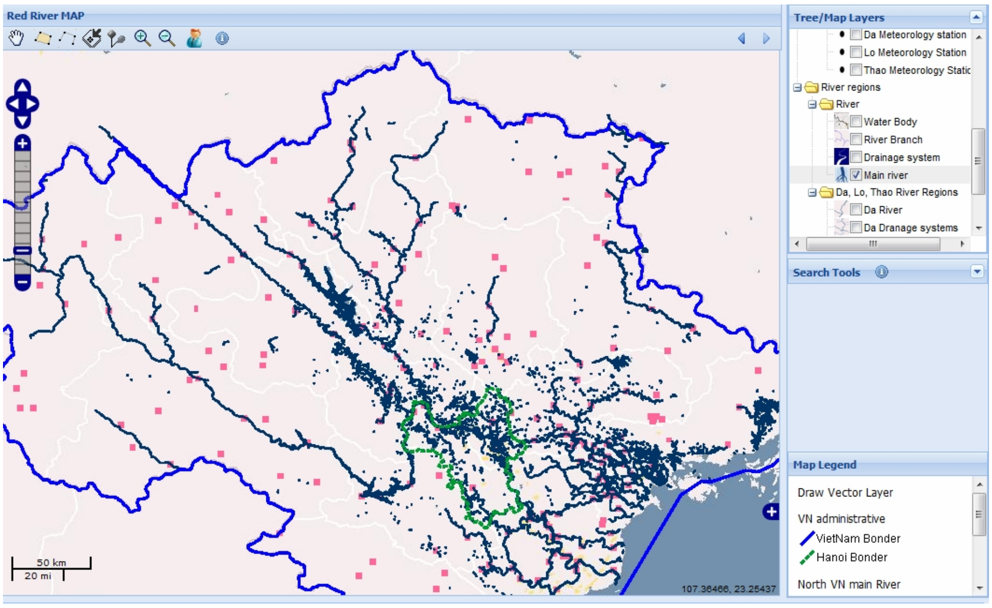


Figure 31: Web-GIS showing Map of hydrology stations in study area.

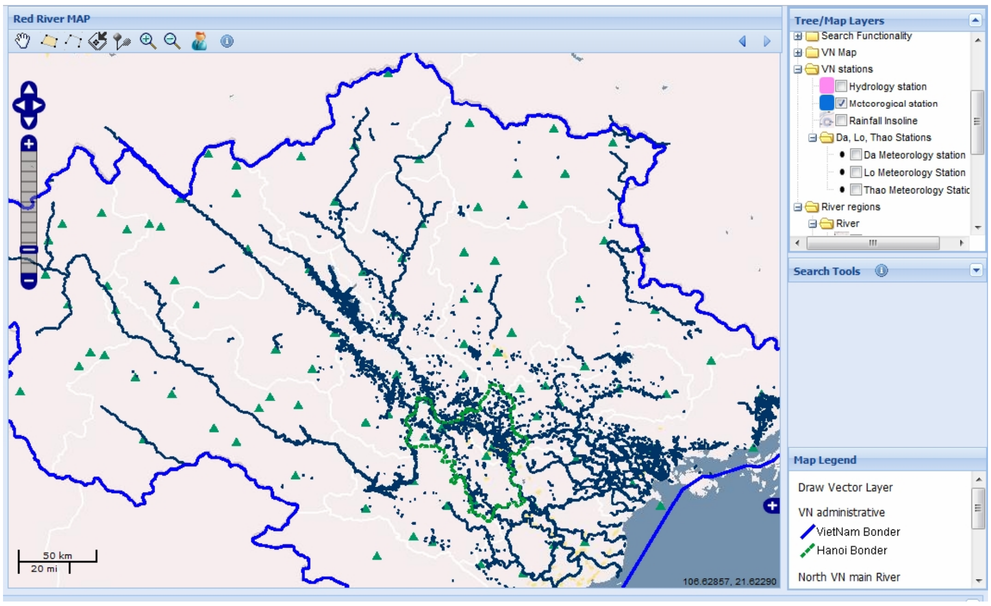


Figure 32: Web-GIS showing Map of meteorological stations in study area

Name Of Station	Rainfall	Discharge	Eva-poration	Water Level	Inflow & reservior
Dien Bien	1963 2004	X	X	X	X
Nam Muc	1964 2004	1960-2004	X	X	X
Than Uyen	1961 2004	X	1961 2002	X	X
Nam Giang	1974 2004	1965 2004	X	X	X
Tuan Giao	1958 2004	X	X	1982 2002	X
Hoa Binh	1957 2004	1956 2004	X	1956 2004	56-08 93-2008
Lai Chau	1957 2004	1957 2004	1982 2002	X	X
Muong Te	1961 2004	X	1982 2002	X	X
Sin Ho	1961 2004	X	1982 2002	X	X
Quynh Nhai	1960 2004	X	X	X	X
Ta Bu	X	1960 2004	X	X	X
Yen Bai	X	1956 2004	X	1958 2004	X
Mu Cang Chai	X	X	1982 2002	X	X
Pha Din	X	X	1982 2002	X	X
Thuong Cat	X	X	X	1957 2004	X

(a)

Name Of Station	Rainfall	Discharge	Eva-poration	Water Level	Inflow & reservior
Son La	1961 2004	X	1965 2002	X	X
Tam Duong	1970 2004	X	1982-2002	X	X
Phu Yen	1961 2004	X	61-76 82-02	X	X
Bac Yen	1973 2004	X	73-77 82-02	X	X
Co Noi	1964 2004	X	1982 2002	X	X
Moc Chau	1961 2004	X	1982 2002	X	X
Thuan Chau	1960 2004	1957 2004	X	X	X
Yen CHau	1961 2004	X	1982 2002	X	X
Thanh Son	1960 2004	X	X	X	X
Son Tay	X	1956 2004	X	1957 2004	X
Ha Noi	X	1956 2004	X	1956 2004	X
Vu Quang	X	1956 2004	X	1972 2005	X
Mai Chau	X	X	1982 2002	X	X

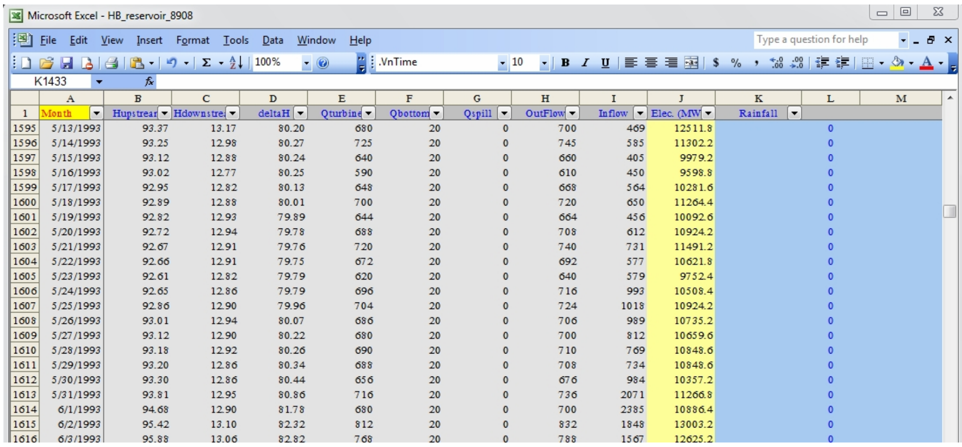
(b)

Table 12 (a) and (b): Represents data used in Red-River Area: in each cell contains starting year and ending year of observations. Discharge and Rainfall were measured by Hydrological Station. Evaporation and Rainfall were estimated from Meteorological Station, and the last column is inflow and reservoir data both data were measured in Hoa Binh Dam.

Microsoft Excel - Discharges

	A	B	C	D	E	F	G	H	I	J
1	QLaichau	QNamgiang	QNammuc	Qtabu	Qhoabinh	QYenbai	QVuquang	QSontay	Qhanoi	
2	0	0	0	0	548	315	0	1120	1060	1/1/1956
3	0	0	0	0	540	306	0	1100	1030	1/2/1956
4	0	0	0	0	532	298	0	1080	1000	1/3/1956
5	0	0	0	0	520	292	0	1080	975	1/4/1956
6	0	0	0	0	513	290	0	1060	940	1/5/1956
7	0	0	0	0	506	282	0	1050	920	1/6/1956
8	0	0	0	0	496	278	0	1050	915	1/7/1956
9	0	0	0	0	488	274	0	1030	905	1/8/1956
10	0	0	0	0	482	276	0	1020	900	1/9/1956
11	0	0	0	0	478	274	0	1000	885	1/10/1956
12	0	0	0	0	474	270	0	996	880	1/11/1956
13	0	0	0	0	474	268	0	992	860	1/12/1956
14	0	0	0	0	474	268	0	988	850	1/13/1956
15	0	0	0	0	471	266	0	984	835	1/14/1956
16	0	0	0	0	468	264	0	980	830	1/15/1956
17	0	0	0	0	460	262	0	976	825	1/16/1956
18	0	0	0	0	454	258	0	968	820	1/17/1956

Figure 33: Excel file showing discharge of water at 9 hydrological stations: LaiChau, NamGiang, Nammuc, Tabu, Hoa Binh, Yen Bai, Vu Quang, Son Tay, Ha Noi.



Microsoft Excel - HB_reservoir_8908

1	Month	Hupstrear	Hdownstre	deltaH	Qturbine	Qbottom	Qspill	OutFlow	Inflow	Elec. (MW	Rainfall		
1595	5/13/1993	93.37	13.17	80.20	680	20	0	700	469	12511.8		0	
1596	5/14/1993	93.25	12.98	80.27	725	20	0	745	585	11302.2		0	
1597	5/15/1993	93.12	12.88	80.24	640	20	0	660	405	9979.2		0	
1598	5/16/1993	93.02	12.77	80.25	590	20	0	610	450	9598.8		0	
1599	5/17/1993	92.95	12.82	80.13	648	20	0	668	564	10281.6		0	
1600	5/18/1993	92.89	12.88	80.01	700	20	0	720	650	11264.4		0	
1601	5/19/1993	92.82	12.93	79.89	644	20	0	664	456	10092.6		0	
1602	5/20/1993	92.72	12.94	79.78	688	20	0	708	612	10924.2		0	
1603	5/21/1993	92.67	12.91	79.76	720	20	0	740	731	11491.2		0	
1604	5/22/1993	92.66	12.91	79.75	672	20	0	692	577	10621.8		0	
1605	5/23/1993	92.61	12.82	79.79	620	20	0	640	579	9752.4		0	
1606	5/24/1993	92.65	12.86	79.79	696	20	0	716	993	10508.4		0	
1607	5/25/1993	92.86	12.90	79.96	704	20	0	724	1018	10924.2		0	
1608	5/26/1993	93.01	12.94	80.07	686	20	0	706	989	10735.2		0	
1609	5/27/1993	93.12	12.90	80.22	680	20	0	700	812	10659.6		0	
1610	5/28/1993	93.18	12.92	80.26	690	20	0	710	769	10848.6		0	
1611	5/29/1993	93.20	12.86	80.34	688	20	0	708	734	10848.6		0	
1612	5/30/1993	93.30	12.86	80.44	656	20	0	676	984	10357.2		0	
1613	5/31/1993	93.81	12.95	80.86	716	20	0	736	2071	11266.8		0	
1614	6/1/1993	94.68	12.90	81.78	680	20	0	700	2385	10886.4		0	
1615	6/2/1993	95.42	13.10	82.32	812	20	0	832	1848	13003.2		0	
1616	6/3/1993	95.88	13.06	82.82	768	20	0	788	1567	12625.2		0	

Figure 34: Excel file of data at Hoa Binh reservoir.

Table 12 presents all data available for study area. (Fig. 33) and (Fig.34) showing the two sample files are discharge and reservoir in the set of database file including Rainfall (RF), Evaporation (EV), Discharge (DR), Water Level (WL), Inflow (IF), Reservior at Hoa Binh (RV).

Chapter 3: WEB-GIS

3.1 Introduction

A WebGIS is combined of two technologies, one is World Wide Web technology and the other one is Geographic Information System, it uses interactive Internet technology to expand and enhance advantage of Geographic Information System. With a growing rapidly of internet technology, quick developing and up-to-date of current Geographic Information System technology, WebGIS have been developing toward opening and standardization.

At first, Internet and GIS advanced in independent fields. GIS installed and independent systems, addition devices with its database designed in local; there are hardly to share database and results. The first WebGIS, Xerox PARC Map Viewer, was issued in June 1993 by Steve Putz at Xerox Corporation's Palo Alto Research Center, as an experiment in providing interactive information retrieval via the World Wide Web, it is the first mapserver based on CGI/Perl, allowed re-projection styling and definition of map extent. In the last decade there has been a rapid increate in the use of web-base GIS applications in field like education, transport, marketing, sociology, business, and disaster recovery. Nowadays, application of WebGIS has gone deeply into each field in our society. It has been widely accepted in governmental agencies and educational institutions and among geospatial data producers and users. They are playing very importance roles not only support information, and sharing database but also widely used as a supporting tool to develop spatial decision support systems.

Web GIS is independent of user equipment and hardware location, and avoid installing expensive software (Fig. 35). For instance when viewing satellite imagery, categorizing land use data, or comparing the changes of land cover before/after the disasters, they are very important messages from nature. These important messages can be received by hundreds of millions people almost immediately via Internet GIS Tsou (2012). Many major commercial companies (such as Digital Globe and Space Imaging) and GIS vendors (ESRI and Google) are generously offering free satellite images and GIS data/maps for the general public and the media. People can access many GIS data and maps in near real-time whenever they are available. For example: maps.google.com is the best examples of Internet GIS applications.

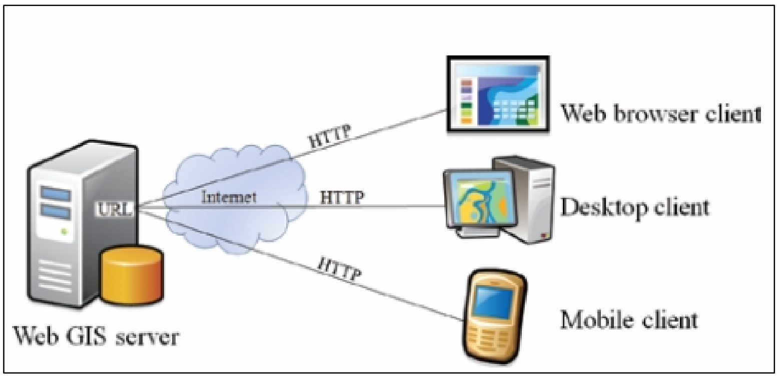


Figure 35: Simple Web GIS system architecture

The rest of this chapter is organized as follows. In section 2, History of WebGIS. In section 3, we explore Networking fundamentals of Web GIS. Section 4 we summarize some of WebGIS strategy. Finally, in Section 5, we presented about Transferred Geo Data.

3.2 History of Web-GIS

In 1982 the Internet Protocol Suite (TCP/IP) was standardized. The main concept of a world-wide network of fully interconnected TCP/IP networks called the Internet was introduced. The success of the Internet and popular adoption of TCP/IP pushed the development of telecommunication into a new age. In 1993, World Wide Web (WWW) was declared, which was regarded as a significant milestone for the Information Revolution (Thoen, 1999). After the emergence of the WWW, Xerox PARC Map Viewer was released in 1993. This application provided quite rudimentary ability to present a map of the world with a few operations such as: zooming at predefine scales, controlling the visibility of rivers and border features. After the first release, some of the milestone of Web-GIS in the last 10 years is presented as bellows (wikipedia, accessed 2012):

- 1994: the Canadian National Atlas Information Service released the first online version of the National Atlas of Canada.
- 1995: the Univeristy of California, Santa Barbara, leading to implement a Webs site portal for the National Geospatial Data Clearninghouse.
- 1996: MapQuest released its Web-mapping application. It allowed people to view maps, look for local businesses, find the optimum route to desired location, and plan trips.
- 1997: The University ò Minnesota (UMN) released MapServer 1.0, an open source development environment for building spatially-enabled internet applications

- 2000: ESRI released Arc Internet Map Server (ArcIMS) a commercial tool for publishing GIS data over the Internet.
- In the same year, in 2000 The Open Geospatial Consortium (OGC) has developed a set of standards for Web GIS. The first was the Web Mapping Service specifications.
- 2001: MapScript 1.0 for UMN MapServer, Adds a lot of flexibility to UMN MapServer solutions.
- 2002: UMN MapServer 3.5, Added support for PostGIS (PostGIS adds support for geographic objects to the PostgreSQL) and ArcSDE (Spatial Database Engine). Version 3.6 adds initial OGC WMS support.
- 2003: NASA World Wind Released. An open virtual globe that loads data from distributed resources across the internet. Terrain and buildings can be viewed 3 dimensionally. The (XML based) markup language allows users to integrate their own personal content. This virtual globe needs special software and doesn't run in a web browser.
- 2004: OpenStreetMap was released
- 2005-02: Google Maps, The first version of Google Maps. Based on raster tiles organized in a quad tree scheme, data loading done with XMLHttpRequests. This mapping application became highly popular on the web, also because it allowed other people to integrate google map services into their own website.
- 2005-04: UMN MapServer 4.6, Adds support for SVG (Scalable Vector Graphics).
- 2005-06: Google Earth, The first version of Google Earth was released building on the virtual globe metaphor. Terrain and buildings can be viewed 3 dimensionally. The KML (XML based) markup language allows users to integrate their own personal content. This virtual globe needs special software and doesn't run in a web browser.
- 2006-05: WikiMapia Launched.

3.3 Networking fundamentals of Web GIS

Purpose of this section is to present main contents of fundamentals of Web GIS, and summery of the needs for Web-GIS system.

3.3.1 Internet

As the Wikipedia defines “The Internet is a global system of interconnected computer networks that use the standard Internet protocol suite, protocol often called TCP/IP, although not all applications use TCP) to serve billions of users worldwide”. It is a network of networks that serves billions of users worldwide or consists of millions private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies.

3.3.2 Computer Network

A computer network, or simply a network, is a collection of two or many computers and other hardware components interconnected by communication channels. It can exchange information and resources. At least one process in one device is able to send/receive data to/from at least one process residing in a remote device, then the two devices are said above to be in a network. Nowadays in the world exits many scales of Computer Networks, there are often classified by their physical or organizational extent or their purpose such as: personal area network (PAN), local area network (LAN), campus area network (ACN); or larger scale: metropolitan area network (MAN), wide area network (WAN) and the most important computer network is Internet, it allows to connect multiple computer networks in the world.

3.3.3 TCP/IP

Transmission Control Protocol (TCP) and Internet Protocol (IP) were developed by a Department of Defense (DOD) research project to connect a number different networks designed by different vendors into a network of networks (the "Internet").

TCP/IP provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination. It has four abstraction layers including link layer; internet layer; transport layer; application layer. TCP can simply understand as a communication between applications, if one application wants to communicate with another via TCP, it sends a communication request. This request must be sent to an exact address, after a connection is successfully created between two applications, TCP will set up a “bridge” in order to communicate between the two applications. The bridge will occupy the communication line between the two computers until it is closed by one of the two applications. IP is for communication between computers. IP reduces the need for network lines. Each line can be used for communication between many different computers at the same time. With IP, messages (or other data) are broken up into small independent "packets" and sent between computers via the Internet.

TCP/IP was successful because it delivered a few essential services such as: file transfer, electronic mail, remote logon across a very large number of client and server systems. Several computers in a small department/company can use TCP/IP on a single LAN.

3.3.4 WWW

The World Wide Web (WWW) is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia, and navigate between them via hyperlinks. Viewing a web page on the World Wide Web normally begins either by calling the URL of the page into a web browser or by following a hyperlink to that page or resource. Web Browsers are well known nowadays such as Internet Explorer, FireFox, Chrome, etc.

3.3.5 HTTP

Hyper Text Transfer Protocol (HTTP) is protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands. For example: when client submits an HTTP request message to the server, the server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. HTTP also defines important methods such as GET, POST and HEAD or added 5 new methods: OPTIONS, PUT, DELETE and CONNECT.

3.3.6 URL

A Uniform Resource Locator (URL) it is the global address of documents and other resources on the World Wide Web. The first part of the URL is called a protocol identifier and it indicates what protocol to use and the second part is called a resource name and it specifies the IP address or the domain name where the resource is located. The protocol identifier and the resource name are separated by a colon and two forward slashes. For example, the two URLs below point to two different files at the domain polimi.it. The first specifies an executable file that should be fetched using the FTP protocol; the second specifies a Web page that should be fetched using the HTTP protocol:

ftp://www.polimi.it; http:// www.polimi.it/home.html

3.3.7 HTML

HyperText Markup Language (HTML) is the main markup language for displaying web pages and other information that can be displayed in a web browser. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags are unpaired, for example <img>. In between these tags web designers can add text, tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible on web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

3.4 Web-GIS strategy

3.4.1 Web client/server architecture

Web-GIS apply the client/server concept in performing GIS analysis tasks. Most of Web-GIS models nowadays use the client/server in which client can request for data or results of specific model from the server. The server both performs a job itself and sends the results back to the client through the network. Only the data are located in the server. Software components are centrally controlled, which makes data easy to maintain, update and support. In case, software updates are only needed for the server. (Fig. 36) bellows showing basic follow chart of client-server model.

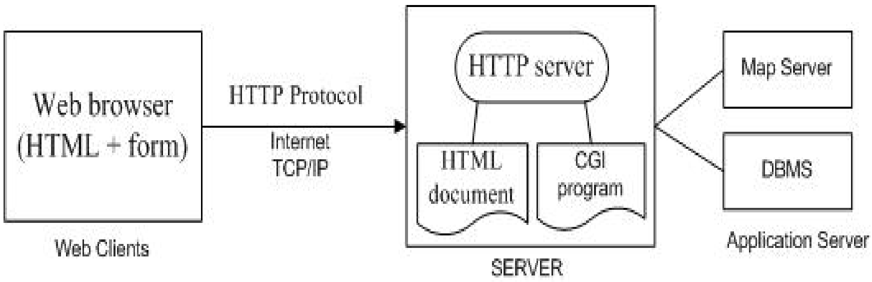


Figure 36: client-server model Basic client server model

The concept of client/server involves tasks between the server and client. A client/server application has two main components: a client and a HTTP server in general. Each of them is supported by specific software and hardware. The client sends a request to the server, server processes the request and returns the result to the client, the client then manipulates the data and/or results and presents to the user over Web Browser. In the server side, server is not only able to read the client's request but also it invokes the others data inside its own disk or invokes GIS programs to process data. The connections between the client and server are established according to a communication protocol such as the TCP/IP. Basically, Client/Server technology includes two major technical approaches: (a) HTTP with Common Gateway Interfaces (CGI); (b) Interact Web Mapping with the help of programs or script such as: GIS plug-ins, GIS ActiveX Controls, GIS Java applets, Scripts, Servlets.

3.4.2 HTTP server with Common Gateway Interfaces (CGI) model

Web pages had no way of interpreting reader reaction except through interaction with the server they were running on. This interaction is done through scripts and programs that use common gateway interface (CGI) to create interactive programs on your Web pages.

The web server receives the HTTP message but it cannot answer, since it can understand only HTML documents. It then passes the request via the CGI to applications able to fulfill it (the CGI program interacts with other applications in the server such as a map server and a DBMS server). The role of map server is to process the user request and to send back the results to the web server via CGI protocol, which reformats the result in an HTML format and ends it to the Web Server. The Web Server then forwards the results to the Web client (see Fig. 37).

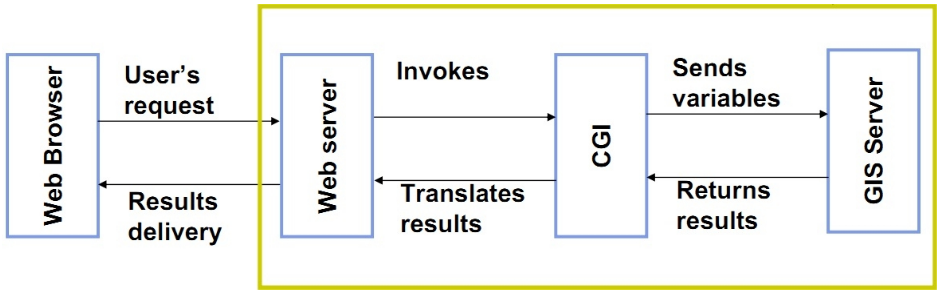


Figure 37: HTTP server with CGI, (M. Brovelli, 2012).

3.4.3 Interact Web-GIS (Web mapping)

a) GIS plug-ins

GIS plug-ins or helper programs are created to work inside the Internet Web browser such as Netscape and Internet Explorer to handle GIS data and maps. The GIS plug-in is small application downloaded from the Web server and installed in Web browser. It operated in the front end at the user's local machine. It acts as a helper application to handle GIS data type that the current HTML (HyperText Markup Language) cannot recognize. Microsoft ActiveX Document is a special helper program.

b) Microsoft ActiveX Controls

The second client-side application is to use Microsoft ActiveX Controls to create GIS Controls. GIS Controls are similar to GIS Plug-ins in that they are created to extend the capabilities of the HTML to handle spatial maps and data. GIS Controls are downloaded and executed in the user's local machine.

c) Java applets

The most recent development is to use Internet programming language Java to create GIS Java applets, which are executable small Java applications. GIS Java applets are downloaded from the Internet servers and are executed inside the Web browser immediately. All the GIS functionalities and data are encapsulated in the package and shipped to the client on demand. No plug-in or helper program is needed to be installed at the client's Web browser. Because the GIS applications are executed at the user's local machine, no unnecessary information exchanges between the Internet users and servers.

d) Scripts

Almost, Scripts uses JavaScript or VBScript technology. Using this technology, it makes easy to put a dynamic map in any web page. It can display maps and makers load from any source.

e) Servlet

A servlet is an implementation of Java that aims to provide the same service as CGI does (see 3.4.1.1), but instead of programs compiled in the native operating system, it compiles into the Java byte-code which is then run in the Java virtual machine (JVM). Though Java programs can be compiled into the native code, they still prefer to compile in the Java byte-code.

One of the advantages of servlets is its platform independence. Servlets can run on any operating system just as long as a Java Virtual Machine is installed inside a system.

3.4.4 Server-Side

In general, Web Server relies on HTTP to communicate with the Web client. Server-side mainly performs analysis and generates output including HTML, text, images, and so on. A user at Web browser client initiates the request that is sent through the Internet to the server. The server processes the request and sent back the results to the user client. The server-side Web-GIS applications include a simple HTML, CGI approach and application-sharing approach using Internet communications.

The initial development of connecting GIS with Internet is by linking geospatial data and its functionality with the Internet server through a Common Gateway Interface (CGI). The GIS server runs on the background at the GIS server and is linked with the Web server. The Internet browser acts as a friendly user interface for the end user to request geospatial data and service. The GIS performs the requested analysis and provides the graphics and text output to the Web server, which in turn presents the results to users or clients. The CGI script handles the information exchange between the Web server and the GIS server.

This approach is mainly for simple map display and does not facilitate much direct interaction between the Internet user and GIS analysis.

Server-side strategy with high advantage such as: user access complex and "heavy" data, user also can simply execute geographic analyses or GIS models in short time by using low-performance computers. However there are also exit some disadvantages in this strategy, example: every operations must through the internet and they take times, response from server depends on connection speed.

3.4.5 Client-Side Internet-GIS

Client is places for users to interact with maps and spatial objects. From very simple HTML interfaces with forms to dynamic HTML or advance client-side with extension client applications, client-side allows GIS analysis and data processing to be executed on the Web browser in the user's local machine Z-R Peng (1999). GIS data and analysis tools initially reside in a server and are downloaded by the client on demand. Users usually request for data and process tools from the server side, which sends the data and analysis modules to the client for local processing. For advance client-side needs client-site applications, they include three major technical approaches: GIS plug-ins and helper programs, GIS ActiveX Controls, and GIS Java applets.

3.5 Transferred Geo Data

A map transmission to the client is converted in to no space raster or a suitable vector format. When raster data is transfer, in Web browser displays GIF or JPEG. That means the data on the server has to be converted to a raster format before send a map to the client side Alesheikh (2002). The data volume due to the known image Techsize and the original data on the server is safe as only an image is sent to the client. The disadvantage of using raster data is the lack of comfort of handling and regarding cartographic aspects, moving over object with mouse cannot highlight single objects.

A vector data is low of volume, it transmits faster than raster. Vector data handled by a standard Web browser with extend functionality. The user gets a more functionality with vector data. For example, single objects can be selected directly or highlighted. One more advantage of using vector data is the possibility of local processing. To avoid data redundancy in client side, dynamic generalization must be provided. Distributing vector data may also endanger copyright rules.

3.5.1 Data for Web GIS

Web mapping services integrate with existing GIS data layers to create high performance web mapping interfaces for web browsers. Since GIS systems produce data in both raster and vector formats, tools and techniques are required to integrate both data formats for existing GIS systems and merge them with Web GIS capabilities. The vector and raster data used can be summarized as bellow:

a) Vector Data consists of combination of spatial primitives: points, lines, polygons, and they are defined in mathematical ways. It is possible to attach event handlers directly to primitive features, producing highly interactive user interfaces. Moreover, attributes can be directly associated to individual spatial features. Common vector data formats for web-base GIS application include JSON/GeoJSON and XML such as KML, GML or GeoRSS. GIS vector datasets need to be transmitted as one of these formats for use.

b) Raster Data – Raster data can include native images, like aerial imagery e.g. GeoTIFF, or raster images rendered from large vector datasets at the server, e.g. land use, land cover maps. Raster data formats supports in Web mapping technologies as jpeg, png or PDF images.

c) Raster + Vector data – Many Web GIS applications take advantage of hybrid raster/vector approaches. Imagery or large vector data sets are displayed as images, either directly (simple method) or in cached tile pyramids (advance method). Smaller vector sets can be displayed as additional vector overlays. User can flexible selection can directly access any vector layers in the server side, but must make server requests for additional information associated with features in raster layers.

Chapter 4: OGC STANSARDS

The Open Geospatial Consortium (OGC) it known an international voluntary consensus standards organization (OGC, 2012), originated in 1994, In the OGC more than 478 commercial, governmental, nonprofit and research organizations worldwide collaborate in a consensus process encouraging development and implementation of open standards for geospatial content and services, GIS data processing and data sharing. The OGC standards baseline comprises more than 30 standards, including:

- Catalogue Service:
  - Catalogue Service Core (CS Core)
  - Catalogue Service Web (CS for the Web to access catalog information)
- Processing Services
  - OpenLS Core Services
  - Sensor Planning Service (SPS)
  - Web Processing Service (WPS)
  - Coordinate Transformation Service (CTS)
  - WCS Processing (with WCS)
- Encodings
  - Geography Markup Language (GML)
  - Filter Encoding (FE)
  - GML in JPEG 2000
  - KML
  - NetCDF
  - Observations & Measurements (O&M)
  - Open GeoSMS
  - Sensor Model Language (SensorML)
  - Symbology Layer Descriptor (SLD)
  - SWE Common
  - Web Map Context (WMC)
- Data Services
  - Simple Feature (SQL)
  - Web Coverage Service (WCS)
  - Sensor Observation Service (SOS)
  - Table Join Service (TJS)
  - Web Feature Service (WFS)
- Portray Service
  - Web Map Service (WMS)
  - Web Map Tilling Service

- Others
    - GeoXACML
    - GeoAPI
    - OWS Common

In this thesis domain, we would like to mention only some of OGC web services and encoding which have already used in domain of my thesis.

4.1 OGC encoding

4.1.1 Geographic Markup Language (GML)

Basically, GML is the XML grammar defined by the OGC to express geographical feature. GML presents a role as a modeling language for geographical feature as well as an open interchange format for geographic transactions on the Internet. The feature in GML includes not only conventional “vector” or distance objects, but also coverages and sensor data. The ability of GML is to integrate all forms of geographic information. GML contains a rich set of primitives which are used to build application specific or application languages. These primitives include:

- Feature
- Geometry
- Coordinate reference system
- Topology
- Time
- Dynamic feature
- Coverage (including geographic images)
- Unit of measure
- Directions
- Observations
- Map presentation styling rules

The original GML model was based on the World Wide Web Consortium's framework for representing information in the Web called Resource Description Framework (RDF) see at http://www.w3.org/RDF/. Subsequently, the OGC introduced XML schemas into GML's structure to help connect the various existing geographic databases. The resulting XML schema based GML retains many features of RDF, including the idea of child elements as properties of the parent object (RDF Schema) and the use of remote property references.

GML defines features distinct from geometry objects. A feature is an application object that represents a physical entity, e.g. a building, a river, or a person. A geometry object defines a location or region of a physical entity. For instance in case coordinate, it can be presented in GML structure such as: <gml:coordinates> the <gml:coordinates> element can be used, as follows:

```
<gml:Point gml:id="p21"
srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
   <gml:coordinates>45.67, 88.56</gml:coordinates>
 </gml:Point>
```

4.1.2 GeoJSON

GeoJSON is an open format for encoding a variety of geographic data structures. GeoJSON supports the following geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollection. It relies on JSON (JavaScript Object Notation). GeoJSON allows geographic data to be stored and is generally more compact than XML. GeoJSON also allows specifying a geographic coordinate system, using the OGC crs (coordinate reference system) property (preferred) or with an EPSG code.

Example:

```
{
   "type":"Feature",
   "id":"OpenLayers.Feature.Vector_314",
   "properties":{},
   "geometry":{
      "type":"Point",
      "coordinates":[20.00, 105.00]
   },
   "crs":{
      "type":"name",
      "properties":{
         "name":"urn:ogc:def:crs:OGC:1.3:CRS84"
      }
   }
 }
```

4.2 Web Map Service (WMS)

WMS is interoperable Web mapping technology; it is a standard interface between client and server via WWW. A Web map service provides maps of georeferenced data which are visual representations of geospatial data. These maps are rendered in normal image formats such as PNG, GIF, JPEG, or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats. Web Map Service (WMS, 2006) operations can be invoked using a standard web browser by submitting requests in the form of Uniform Resource Locators (URLs). The content of such URLs depends on which operation is requested. In particular, when requesting a map the URL indicates what information is to be shown on the map, what portion of the map is to be mapped on the desired coordinate reference system, and the output image type, width and height; when two or more maps are produced with the same geographic parameters and output size, the results can be accurately overlaid to produce a composite map. Example bellow is the URL request:

http://localhost/mapserver.cgi?VERSION=1.3.0&REQUEST=GetMap&CRS=CRS:84
&BBOX=-97.105,24.913,-78.794,36.358& WIDTH=560&HEIGHT=350&
LAYERS=AVHRR&STYLES=& FORMAT=image/png&EXCEPTIONS=INIMAGE

There operations defined for a WMS a GetCapabilities, GetMap, GetFeatureInfo. GetFeatureInfo is optional. The overview for each request will be shown bellow. International Standard defines two classes, one for a basic WMS, and the other for a queryable WMS data:

- Basic WMS shall support the basic service elements including the GetCapabilities operation and the GetMap operation.
- A queryable WMS shall satisfy all the requirements for a basic WMS, and shall also support the GetFeatureInfo operation.

The most use requests represented by WMS are GetMap, GetFeatureInfo. GetMap operation returns a map. GetFeatureInfo provide feature information by identifying a point on a map.

1. GetCapabilities

In the general form of a WMS request, the GetCapabilities request of a WMS server, it is necessary to indicate that the client seeks information about the

Web Map Service in particular. Request of parameter is presented in the Table 13 bellow:

Request parameter	Mandatory/Optional	Description
VERSION=version	O	Request version
SERVICE=WMS	M	Service type
REQUEST=GetCapabilities	M	Request name
FORMAT=MINE_type	O	Output format of service metadata
UPDATESEQUENCE=string	O	Sequence number or string for cache control

Table 13: Parameters of GetCapabilities request WMS (2006)

The example bellow shows the minimum number of parameters needed to make a successful GetCapabilities request

```
http://webservices.nationalatlas.gov/wms/agriculture?
SERVICE=WMS&
REQUEST=GetCapabilities
```

2. GetMap

The example bellow shows the minimum number of parameters need to make a successful GetMap request:

```
http://localhost/mapserver.cgi?
VERSION=1.3.0&
REQUEST=GetMap&
CRS=CRS:84&
BBOX=-97.105,24.913,-78.794,36.358&
WIDTH=560&HEIGHT=350&LAYERS=AVHRR0927&
STYLES=& FORMAT=image/png
```

GetMap Operation returns a map, the parameters of a GetMap request shown in the Table 14 below:

Request parameter	Mandatory or Optional	Descirption
VERSION=1.3.0	M	Request version
REQUEST=GetMap	M	Request name
LAYER=layer_list	M	Comma-separated
STYLE=style_list	M	Comma-separated list of one rendering style per requested layer
CRS=namespace:identifier	M	Coordinate reference system
BBOX=minx,miny,maxx,maxy	M	Bounding box corners in CRS units
WIDTH=output_width	M	Width in pixels of map picture
HEIGHT=output_height	M	Height in pixels of map picture
FORMAT=output_format	M	Output format of map
TRANSPARENT=TRUE|FALSE	O	Background transparency of map
BGCOLOR=color_value	O	Hexadecimal re-green-blue color value fro the background color
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS
TIME=time	O	Time value of layer desired
ELEVATION=elevation	O	Elevation of layer desired
Other sample dimension(s)	O	Value of other dimensions as appropriate

Table 14: Showing mandatory and optional parameters of the GetMap request WMS (2006)

3. GetFeatureInfo

Provides feature information by identifying a point on a map based on its pixel location. The GetFeatureInfo request embeds many of parameters required in the GetMap request along with parameters specific to query layers. In the OGC WMS, GetFeatureInfo is optional

The following parameters can be appended to the GetFeatureInfo request is shown in the Table 15 bellow:

Request parameter	Mandatory/ optional	Description
VERSION=1.3.0	M	Request version
REQUEST=GetFeatureInfo	M	Request name
Map request part	M	Partial copy of the Map request parameters that generated the map for which information is desired
QUERY_LAYERS=layer_list	M	Comma-separated list of one or more layers
INFO_FORMAT=output_format	M	Return format of feature information
FEATURE_COUNT=number	O	Number of features about which to return information
I=pixel_colum	M	i coordinate in pixels of feature in Map CS
J=pixel_row	M	J coordinate in pixels of feature in Map CS
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS.

Table 15: Showing mandatory and optional parameters of the GetFeatureInfo request WMS (2006).

4.3 Style Layer Descriptor Encoding (SLD) profile of the Web Map Service

The importance of the visual portrayal of geographic data because it is needed to support style a data set based on attributes of feature type. For example, hydrology system consisting of lines (rivers and streams) and polygons (lakes, pounds, oceans, etc.), a user might want to fill color inside of all polygons in a blue and color the boundaries of all polygons and all lines in a darker blue.

Web Map Service can be extended to allow user-defined styling (SLD, 2007). An SLD file is similar to an XML file that follows the OGC SLD syntax so that it can be understood in server and client communication. Logically, one such SLD style defines symbolization rules for specified sets of features within a specified WMS layer. One style can have various rules and each rule can have one filter describing which features should be rendered.

In the WMS specification, the request for a map is encoded as an HTTP-GET or HTTP-POST request and the appearance for a map portrayal is specified by the LAYERS and STYLES parameters. The below example is an example of SLD applied for total main river in North and the boundary of Vietnam (Fig. 38).

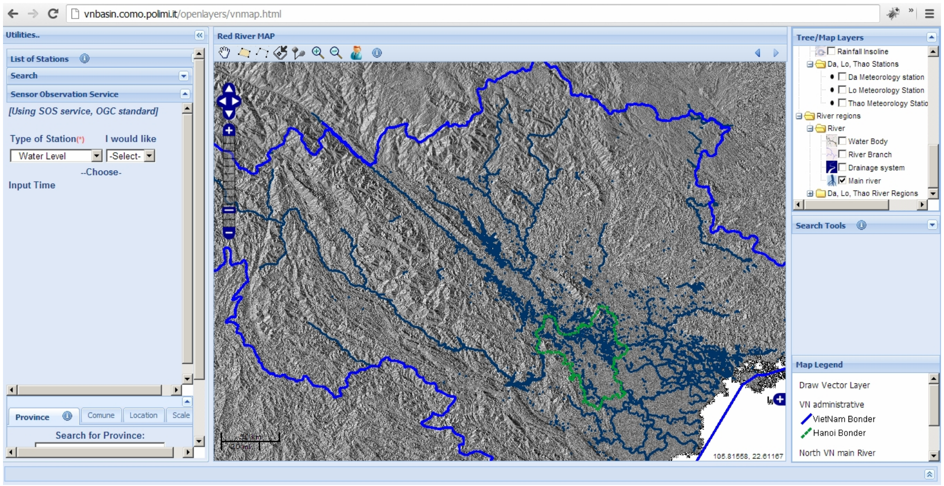


Figure 38: SLD visual portrayal for all the main river in the north of Vietnam in dark-blue and boundary of Vietnam in lighter blue.

4.4 Web Feature Service (WFS)

This International Standard specifies the behaviour of a service that provides transactions on and access to geographic feature. It specifies discovery operations, query operations, locking operations, transaction operations to manage stored parameterized query expression. Transaction operations allow features to be created, changed, replaced and deleted from the underlying data store. Stored query operations allow clients to create, drop, list and describe parameterized query expressions that are stored by the server and can be repeatedly invoked using different parameter values. The Standard operations includes such as bellow operations:

- GetCapablilities (discovery operation)
- DescribleFeatureType (discovery operation)
- GetPropertyValue (query operation)
- GetFeature (query operation)
- GetFeatureWithLock (query and locking operation)
- LockFeature (locking operation)
- Transaction (transaction operation)
- CreateStoredQuery (store query operation)
- DropStoredQuery (stored query operation)
- ListStoreQueries (stored and operation)
- DescribeStoredQueries (Stored query operation)

Some of above operations are the most used in many projects and regions, such as GetCapabilities, DescribleFeatureType, they are concepts shown as detail bellow:

a) GetCapablilities

A web feature service provides user to access its capabilities. Specifically, it must select which feature types it can be service and what operations are supported on each feature type. An example of GetCapabilities being shown as bellow:

To issue GET request using HTTP

```
http://localhost/geoserver/wfs?
Service=wfs&
Version=1.1.0&
Request=GetCapabilities
```

In the bellow example is the XML fragment illustrates POST request:

```
<GetCapabilities
 service="WFS"
 xmlns="http://www.opengis.net/wfs"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=”http://www.opengis.net/wfs
  http://schemas.opengis.net/wfs/1.0.0/wfs.xsd"/>
```

The parameters for GetCapabilities are presented in Table 16:

Parameter	Require	Description
Service	Yes	Service name- Value is WFS
Version	Yes	Service version
Request	Yes	Operation name

Table 16: showing parameters of GetCapabilities

b) DescribeFeatureType

A web feature service must be able, upon request, to describe structure of any feature type it can service an example of GetCapabilities being shown as bellow:

Example of GET Request:

```
http://example.com/wfs?
service=WFS&version=1.0.0&
REQUEST=DescribeFeatureType
```

POST Request

The request is most often constructed as an XML document and passed to the server using an HTTP Post request. The following XML schema is HTTP Post example:

```
<?xml version="1.0"?>
<wfs:DescribeFeatureType
        version="1.0.0"service="WFS"
        xmlns:wfs="http://www.opengis.net/wfs"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.opengis.net/wfs../wfs/1.0.0/WFS-
basic.xsd">
</wfs:DescribeFeatureType>
```

Request contains parameters as the following in Table 17:

Input	Require	Description
REQUEST	Mandatory	Name of request
SERVICE=WFS	Mandatory	Service type, must be WFS
OUTPUTFORMAT	Optional	XML SCHEMA
TYPENAME	Optional	A comma separated list of feature type to describe
VERSION=1.0.0	Optional	Request version number

Table 17: showing parameters of DescribeFeatureType

c) GetFeature

A web feature service must be able to service a request to retrieve features. In addition, the client should be able to specify which feature properties (Table 18) to obtain and should be able to restrict the query spatially and non-spatially.

For example of GetFeature GET

```
http://localhost/test/wfs?request=getfeature&
typename=cities&
service=wfs&
version=1.0.0
```

The following response is from the GetFeature POST request

```
	<wfs:GetFeature xmlns:wfs="http://www.opengis.net/wfs"
	xmlns:ogc="http://www.opengis.net/ogc"
xmlns:gml="http://www.opengis.net/gml"
	xmlns:cgf="http://www.opengis.net/cite/geometry"
outputFormat="GML2" version="1.0.0"	service="WFS"><wfs:Query
typeName="esri:cities"></wfs:Query>
	</wfs:GetFeature>
```

Input	Require	Description
REQUEST	Mandatory	Name of WFS request
SERVICE	Mandatory	Service type
TYPENAME	Mandatory	A list of feature type names to query
BBOX	Optional	To return features in only those that are not disjoint with a bounding box definded by BBOX
FEATUREID	Optional	List of feature instances to be fetch by a WFS
FILTER	Optional	A filter specification describes a set of features to operate upon.
MAXFEATURES	Optional	A positive integer indicating the maximum number of features that should be returned by WFS to a request.
OUTPUTFORMAT	Optional	The output format used to describe feature data returned by WFS
PROPERTYNAME	Optional	A list of properties may be specified for each feature type that is being queried
VERSION	Optional	Request version number

Table 18: showing parameters of GetFeature

4.5 Sensor Web Enablement (SWE)

OGC members are specifying interoperability interfaces and metadata encodings that enable real time integration of heterogeneous sensor webs into the information infrastructure. Users will use these specifications in creating their our applications, platforms, and products involving Web-connected devices such as flood gauges, air pollution monitors, stress gauges on bridges, mobile heart monitors, Webcams, and robots as well as space and airborne earth imaging devices (SWE, 2012).OGC members have developed and tested the following candidate specifications. Others are planned.

1. Observations & Measurements (O&M) - Standard models and XML Schema for encoding observations and measurements from a sensor.

2. Sensor Model Language (SensorML) - Standard models and XML Schema for describing sensors systems and processes associated with sensor observations; provides information needed for discovery of sensors.
3. Transducer Model Language (TransducerML or TML) - The conceptual model and XML Schema for describing transducers and supporting real-time streaming of data to and from sensor systems.
4. Sensor Observations Service (SOS) - Standard web service interface for requesting, filtering, and retrieving observations and sensor system information. This is the intermediary between a client and an observation repository or near real-time sensor channel.
5. Sensor Planning Service (SPS) - Standard web service interface for requesting user-driven acquisitions and observations. This is the intermediary between a client and a sensor collection management environment.
6. Sensor Alert Service (SAS) - Standard web service interface for publishing and subscribing to alerts from sensors.
7. Web Notification Services (WNS) - Standard web service interface for asynchronous delivery of messages or alerts from SAS and SPS web services and other elements of service workflows.

4.5 Observations & Measurements (O&M)

An O&M provides general models and schema for supporting the packaging of observations from sensor system and sensor-related processing. The core of the standard provides the Observation schema. An observation can receive results in the estimate of the value of a feature property, and involves application of a specified procedure, such as a sensor, instrument, and algorithm or process chain. The procedure may be applied in-situ, remotely, with respect to the sampling location. Observation details are also important for data discovery and for data quality estimation. O&M defines a core set of properties for an observation:

- feature of interest
- observed property
- result
- procedure – the instrument, algorithm or process used (which may be described using SensorML)
- phenomenon time – the real – world time associated with the result
- result time – the time when the result was generated
- valid time – the period during which the result may be used

4.5.1 SensorML

SensorML is an approved Open Geospatial Consortium standard. SensorML (SensorML, 2012) provides standard models and an XML encoding for describing any process, including the process of measurement by sensors and instructions for deriving higher-level information from observations, including both dynamic and stationary platforms and both in-situ and remote sensors. The purpose of SensorML should respect to some aspect bellow:

- Provide descriptions of sensors and sensor systems
- Provide sensor and process information in order to support of resource and observation discovery
- Support the processing and analysis of the sensor observations
- Support the geolocation of observed values
- Provide performance characteristics
- Provide an explicit description of the process by which an observation was obtained
- Provide an executable process chain for deriving new data products on demand
- Archive fundamental properties and assumptions regarding sensor systems

4.5.2 Sensor Observation Services (SOS)

SOS is a web service to query real-tine sensor data and sensor data time series and its part of Sensor Web. The offered sensor data comprises descriptions of sensors themselves, which are encoded in the SensorML and the measured values in the O&M encoding format. An operation of SOS has three components, transactional operations (transaction profile) and extended operations (enhanced profile).

- SOS core operators include: GetCapabilities for request a description of service and sensors; DescribeSensor for retrieving information of specific sensors; GetObservation for querying observations of certain sensors or phenomena. The response to a GetObservation request is encoded in Observations & Measurements O&M which is the general models and XML encodings for sensor observations and measurements.

- Transactional operations include RegisterSensor allowing to register a new sensor in an deployed SOS; InsertObservation can be used insert data of already registered sensors in the SOS.

- Extened operations (enhanced profile) include: GetResult for providing the ability to query for sensor readings without the metadata given consistent metadata; GetFeatureOfInterest for returning the geoobject whose properties are monitored by sensors in Geography Markup Language (GML).

4.6 Web Processing Service (WPS)

WPS defines a standardized interface (WPS, 2007) that facilitates the publishing of geospatial processes, and the discovery of and binding to those processes by clients. "Process" includes any algorithm, calculation or model those operate on spatially referenced data. "Publishing" means making available machine-readable binding information as well as human readable metadata that allows service discovery and use.

A WPS can be configured to offer any sort of GIS functionality to clients across a network, A WPS may offer calculations as simple as buffering points or line segments or polygons, or complicated such as hydraulic models or global climate change model. The data required by the WPS can be delivered across a network afterward process in server side and finally responses all data available over a network. The aim of WPS is processing both vector and raster data.

This interface specification provides mechanisms to identify the spatially referenced data required by the calculation, initiate the calculation, and manage the output from the calculation so that the client can access it.

The WPS specification is designed to allow clients to input data and execute the process with no specialized knowledge of the underlying physical process interface or API. The WPS interface standardizes the way processes and their inputs/outputs are described, how a client can request the execution of a process, and how the output from a process is handled.

The data require by the service can be delivered across a network, or available at the server. This data can be image data formats or data exchange standards such as Geography Markup Language (GML), GeoJSON, KML. The calculation can be as simple as subtracting or adding one set of spatially referenced numbers from another or to another, or as complicated as a hydraulic model or climate change model.

4.6.1 WPS Operations

The WPS interface specifies three operations that can be requested by a client and performed by a WPS server, all mandatory implementation by all servers. Those operations are:

a) *GetCapabilities* – This operation allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation. The GetCapabilties operation provides the names and general descriptions of each of the processes offered by a WPS instance. This operation also supports negotiation of the specification version being used for client-server interactions.

b) *DescribeProcess* – This operation allows a client to request and receive back detailed information about the process that can be run on the service instance, including the inputs required, their allowable formats, and the outputs that can be produced.

c) *Execute* – This operation allows a client to run a specified process implemented by the WPS, using provided input parameter values and returning the outputs produced.

These operations have many similarities to other OGC Web Services, including the WMS, WFS, and WCS. Otherwise, WPS is also compatible with both Web Services Description Language (WSDL) see at http://www.w3.org/TR/wsdl and Simple Object Access Protocol (SOAP) see at http://www.w3.org/TR/soap/; SOAP can be used to package WPS requests and responses. WPS support for WSDL to identifie how a service should be described, but not what service interface should look like, because WPS describes a significant portion – the common portion of what nay service should look like WPS supports the use of WSDL for an individual WPS process, as well as for the entire WPS instance that may include several processes. It is not possible to generate a single generic WSDL document that describes all WPS implementations.

4.6.1.1 GetCapabilties operation

The GetCapabilties operation (mandatory), the mandatory GetCapabilities operation allows clients to retrieve service metadata from a server. The response to a GetCapabilities request shall be a XML document containing service metadata about the server, including brief metadata describing all the processes implemented (Table 19)

ame	Multiplicity
Service	One (mandatory)
Request	One (mandatory)
Version	Optional
Language	Optional

Table 19: the optional list of parameter in the GetCapabilities

A client could issue the following key/value pair encoded of GetCapabilities operation request:

```
http://localhost/cgi-bin/wps.cgi?
REQUEST=DescribeProcess&
SERVICE=WPS&
version=1.0.0&
Identifier=Buffer
```

GetCapabilities operation response, the service metadata document shall be an XML Capabilities document that contains the parameters and sections such as: service, version, service Identification, service provider, operations metadata, etc.

4.6.1.2 DescribeProcess operation

The mandatory DescribeProcess operation allows WPS clients to request all details of description of one or more processes that can be executed by the "execute" operation. This description includes the input and output parameter and formats. All WPS servers shall implement HTTP GET transfer of the DescribeProcess operation request, using encoding. The encoding of the DescribeProcess operation request shall use the parameter specified in (table 4.8).

Name and Example	Optionality	Definition and format
Service=WPS	Mandatory	Service type identifier
Request=DescribeProcess	Mandatory	Operation name
Version=1.0.0	Mandatory	WPS specification and schema version for this operation
Language=en-CA	Optional	Language of the human readable text in response
Identifier=buffer, union	Mandatory	List of one or more process identifiers as listed in the Capabilities document, separated by commas.

Table 20: DescribeProcess operation request URL parameters

A client could issue the following key/value pair encoded of DescribeProcess operation request:

```
http://127.0.0.1/cgi-bin/wps.cgi?
ServiceProvider=&metapath=&
Service=WPS&
Request=DescribeProcess&
Version=1.0.0&
Identifier= buffer
```

A DescribeProcess operation request shall be a Process Descriptions data structure which contains one or more Process Descriptions for each request. Each Process Description includes the brief information returned in the ProcessOfferings section of the service metadata document. Below is the example of list of metadata's DescribeProcess:

- Identifier, title, abstract (and further metadata)
- List of inputs
- List of outputs
- Whether output storage is supported
- Whether asynchronous execution is supported

Example of structure of a *describeProcess* response document

```
<wps:ProcessDescriptions
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd"
service="WPS" version="1.0.0" xml:lang="en-CA"><wps:ProcessDescription
wps:processVersion="1.0" storeSupported="true" statusSupported="true">
<ows:Identifier>reprojectCoords</ows:Identifier>
<ows:Title>reproject coordinates</ows:Title>
<ows:Abstract>reprojectCoords uses gdaltransform to reproject a list of
coordinates from one
projection to another</ows:Abstract>
<ows:Metadata></ows:Metadata>
[..]
  <DataInputs>
    <Input>[..]</Input
    <Input>[..]</Input>[..]
  </DataInputs>
  <ProcessOutputs>
    <Output>[..]</Output>
    <Output>[..]</Output>
    [..]</ProcessOutputs>
```

```
</wps:ProcessDescriptions>
```

Inputs data is represented inside the wps:DataInputs/wps:Input tag elements, it describes input parameters expected by this process:

- Identifier, title, abstract, further metadata
- Category of the input data:
  - Complex data
  - Literal data
  - Bounding box data
- Min/max occurences (input parameter cardinality)

For each input parameter, the process can indicate that it needs one of the following:

ComplexData

ComplexData type is used to describe vector, raster (or any other data). There are two ways, how complex data are handled:

- Embedded in the XML content, for example a GML's content inside the ComplexData element or a binary raster data encoded.
- URL reference to the data or service that will provide data (like a WFS URL).

Describes the XML content of ComplexData as bellow:

```
<ows:Identifier>inputPolygon</ows:Identifier>
<ows:Title>Polygon to be buffered</ows:Title>
<ows:Abstract>GML that describes the polygon to be
buffered</ows:Abstract>
<ComplexData>
 <Default>
 <Format>
 <MimeType>application/gml+xml</MimeType>
 <Encoding>UTF-8</Encoding>
```

```
<Schema>http://schemas.opengis.net/gml/3.1.0/base/geometryBasic2d
.xsd</Schema>
 </Format>
 </Default>
<Supported>
[..]
</Supported>
</ComplexData>
<Input minOccurs="1" maxOccurs="1">
<ows:Identifier>inputImage</ows:Identifier>
<ows:Title>Input image</ows:Title>
<ComplexData>
 <Default>
 <Format>
 <MimeType>image/tiff</MimeType>
 </Format>
</Input>
```

LiteralData

LiteralData can be any character string, float, date, etc. It normally described as primitive data type in the W3C XML Schema standard, see at (http://www.w3.org/TR/xmlschema-2/). The WPS standard also allows the use of UOM (Unit of Measures), in case of missing inputs for an optional input the service will use a default value. Below is the simple example of XML schema for LiteralData:

```
<Input minOccurs="0" maxOccurs="1">
 <ows:Identifier>BufferDistance</ows:Identifier>
```

```
<ows:Title>Buffer Distance</ows:Title>

<ows:Abstract>Distance to be used to calculate buffer.</ows:Abstract>

<LiteralData>

<ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#float">float</ows:DataType>

<UOMs>

    <Default>

    <ows:UOM>meters</ows:UOM>

    </Default>

    <Supported>

    <ows:UOM>meters</ows:UOM>

    <ows:UOM>feet</ows:UOM>

    </Supported>

</UOMs>

<ows:AnyValue/>

<DefaultValue>100</DefaultValue>

</LiteralData>

</Input>
```

BoundingBoxData

Bounding Box (BBOX) is the last data type (the third data type) and it is used to describe a bounding box area. The input description must state the default coordinate reference system (CRS) (normally a URI of the EPSG code system) and other CRS supported.

4.6.1.3 Execute operation

Execute request will launch the specified process implemented by WPS which was configured in the server side. Each request, it uses the input parameter values provided and returning the output values after processing a request. Inputs can be invoked directly in the Execute request, or reference web accessible resources. The outputs can be returned in an XML form response document, either embedded within the response document or stored as web accessible resources. If the outputs are stored, the Execute response shall consist of a XML document that includes a URL for each stored output, which the client can use to retrieve those outputs. Alternatively, for a single output, the server can be directed to return that output in its raw form without being wrapped in an XML response document.

In an Execute request a client must specify:

Process identifier as define in the list of GetCapabilities

Input value as defines in the DescribeProcess

Version and language

Type of Output either:

1. Store in the server
2. Contained inside the XML response
3. Raw response of single output.

- The server shall return a status document
- The input data should be returned in the response document

Execute operation request in Table 21 bellow:

Name	Definition	Data type and value	Multiplicity and use
Service	Service type identifier	Character String type	mandatory
Request	Operation name	Character String type "Execute"	mandatory
Version	Specified version	Charater string type	mandatory
Identifier	Name of process	String type, name of process used in Capabilities document	mandatory
DataInputs	List of inputs provides to this process execution	DataIputs data structure see (a)	Optional, include if have any input
Resonse Form	Defines the response type of the WPS, either raw data or XML document. If absent, the response shall be a response document which includes all outputs encoded in the response.	ResponseForm type data structure (b)	Zero or one (optional)
Language	Language identifier	Character string type, RFC4646 language code of the human readable text	Optional

Table 21: Table of execute operation request's parameters

a) DataInputs data structure

- Identifier
- Title
- Abstract
- InputData (choice)
    - Reference (Identifies this input data as a web accessible resource, and references that resource), InputReference data structure includes: mimeType, encoding, schema, href, method, hearder, body, BodyReference
    - Data (Identifies this input data as being encapsulated in the Execute request) eg: ComplexData, LiteralData, BoundingBoxData.

b) Response Form data

- Response XML Document (Indicates that the outputs shall be returned as part of a WPS responses document.).
- RawData Output (Indicates that the output shall be returned directly as raw data, without a WPS response document)

c) Encoding

Encoding of the DataInputs, ResponseDocument, and RawDataOutput value fields shall be as follows:

- A semicolon (;) shall be used to separate one input from the next
- An equal sign (=) shall be used to separate an input name from its value and attributes, and an attribute name from its value
- An at symbol (@) shall be used to separate an input value from its attributes and one attribute from another.
- Incorrect field names and attribute names shall raise an InvalidParameterException. Missing mandatory field names shall raise a MissingParameterValue. All field values and attribute values shall be encoded using the standard Internet practice for encoding URLs.

An example Execute operation request using encoding is:

```
http://localhost/wps?
request=Execute&
service=WPS&
version=1.0.0&
 language=en-CA&
Identifier=Buffer&
DataIputs=Object=@xlink:href=http%3A%2F%2Ffoo.bar%2Ffoo;BufferDist
ance=10&
ResponseDocument=BufferedPolygon&
StoreExecuteResponse=true
```

Or the other example of Execute operation request using encoding

```
http://localhost/cgi-bin/wps.cgi=WPS&version=1.0.0&
Request=Execute&
Identifier=ProcessName&
DataInputs=fieldName=value1@
attributeName=value2@...;nextFieldName=value_i...
```

Examples (for clarity the values are not URL encoded):

Literal Data example:

```
width=35@datatype=xs:integer@uom=meter
```

BoundingBox Data example:

```
bboxInput=46,102,47,103,urn:ogc:def:crs:EPSG:6.6:4326,2
```

Reference example:

```
fieldName=xml@Format=text/xml@Encoding=utf8@
Schema=xsd@asReference=true
```

d) Execute response parameters

The form of the response from Execute operation request depends on the value of the response form parameter in the execute request side. A response form of RawDataOutput is requested, when a process execution is successful, only one complex output is produced, and then the Execute operation response will consist simply of that one complex output in its raw form returned directly to the client. For example, if in the case where a WPS process creates one GIF image as its output, that GIF image would be returned to the. In other cases, the response to a valid Execute operation request is an ExecuteResponse XML document. The contents of this ExecuteResponse document depend upon the value of the Execute, ResponseForm, ResponseDocument element.

Chapter 5: OGC FRAMEWORKS

WebGIS being one of the most new subjects in Geographic Information system, WebGIS technologies and softwares are being developing in the both ways, commercial and open source. About commercial software, purpose of commercial company is to make their software easy to use, more advance features with well document and the other supports, but the price is high. While open source software is developed by communities and can be used free of charge but they may have poor documentation, but recently, open source system is developing very rapid and increasing large number of users and developers day by day, two Web-GIS technologies can be listed as bellow:

- Commercial: ArcGIS Server (ESRI), GeoMedia WebMap by Intergraph, mapXtreme by MapInfo and MapGuide by Autodesk.
- Open source: GeoServer, MapServer, OpenLayers, Degree, specific tools support for one or more OGC web services, etc.

The other two technologies are mentioned and already successfully applied in many previous works presented bellow:

- Public APIs: Google Maps, Yahoo! Maps, Microsoft Virtual Earth (2D),MapQuest's OpenAPI.
- Globes: Google Earth, Microsoft Virtual Earth (3D), ArcGIS Explorer, NASA WorldWind.

The advantages of the public APIs are free and the API provider supplies a set of base layers with symbology and resolution can be changed logically as the map's scale is changed. Some advantage of base layers may also become a disadvantage if users don't like them or developers want to change something in, because they have a little or no support for user in some case, e.g: they do not support some specific web map service such as SOS or WPS, etc. And also user also faces with other problems such as: Google will charges based on API usage (in case using Google Maps). Finally, the advantage of globe technologies is realism and the fact, that is not present in the 2-D technologies. But they require the installation of either a separate application or a browser plug-in to be viewed, unlike the 2-D maps that are viewable in standard web browsers.

Many organizations are interested in to distribute map and processing tools without time and location restriction to users Alesheikh (2002). Web GIS become a cheap and easy way to distribute geospatial data over the world and enrich their functionality. In this research we used only Open Source Software (OSS) for implementing Web-GIS and web service for that reason in the domain of chapter mainly presents OSS and concentrates to their specific tools. Web-GIS not only develop trend of GIS technologies but also it can display GIS functions in Internet. There are two interactive methods: client and server sides.

5.1 Client and Server software

Client send a request and server side receive, then process, finally send back results client. This method is mainly use in basic map operations, such as: zooming, bar, pan, etc. In this case, it needs:

- Web server software such as: Apache ms4w web server includes map server to manage geographic data (see at http://www.maptools.org/) or Apache Tomcat with GeoServer (http://docs.geoserver.org/) to build up web server and setup geographic data in server side.
- Software supports for client side such as: openlayers, p.mapper, mapfish, ect.

In some case, the client needs to request some complicated or specific process, client send request, server accept the request, and then process request in the server side, e.g: generate a map, connect to PostgeSQL in order to retrieve data, invoke GRASS's command to calculate data, etc. this method is mainly needs to use Map Server, DataBase Management System (DBMS) software and GRASS software also, see example in Figure 39.

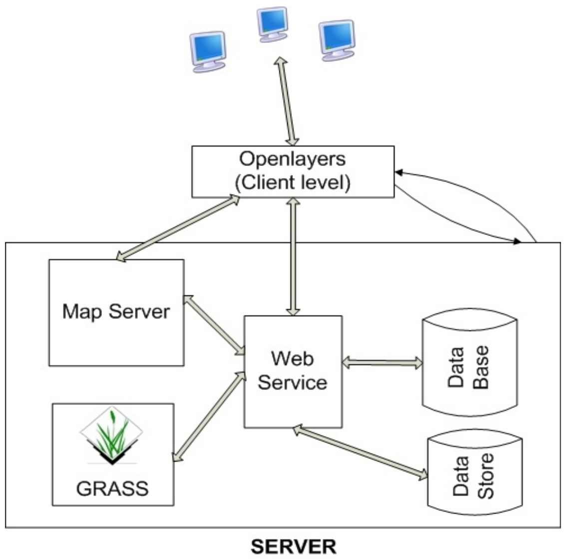


Figure 39: Showing Architecture of Web-GIS with GRASS invoked by using WPS

Client side request, server side accept the request and send back the valid data after processing data to client side, the client side will process data timely and display. This method is mainly used in part process of spatial analysis. Example: client side request statistic chart of rainfall data for 30 days, when client send request to a server. In the server side, it connect to DBMS in order to extract attributes of rainfall data and then build XML document, finally, respond to the client. Client side can handle result from XML to create independently static's chart (Fig 40).

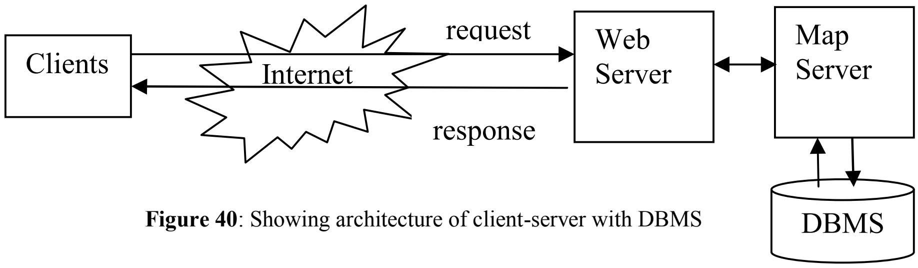


Figure 40: Showing architecture of client-server with DBMS

5.2 Server Software

Open Source software map server (OSSMS) enables the sharing, adding, converting and editing of geospatial data on the Internet via open standards. Among the different kinds of Open Source map server software we have decided to focus on MapServer and GeoServer, since these are the most common geographic servers used and supported by a large developer's community.

5.2.1 Map Server

Map Server is an Open Source project whose purpose is to display dynamic spatial maps over the Internet. The main features are listed below:

- Support for display and querying of hundreds of raster, vector and database formats;
- Run on different operating systems (Windows, Linux, Mac, etc.);
- Support of script languages and development environments (PHP, Python, Perl, Ruby, Java, .NET);
- High quality map rendering;
- On-the-fly projections;
- Fully customizable application output.

Additionally, MapServer is a CGI program that sits inactive on the Web server. When a request is sent to MapServer, it uses information passed in the request URL and the mapfile to create an image of the requested map. MapServer supports numerous OGC specifications as WMS, WFS, WCS and SOS (Fig. 41).

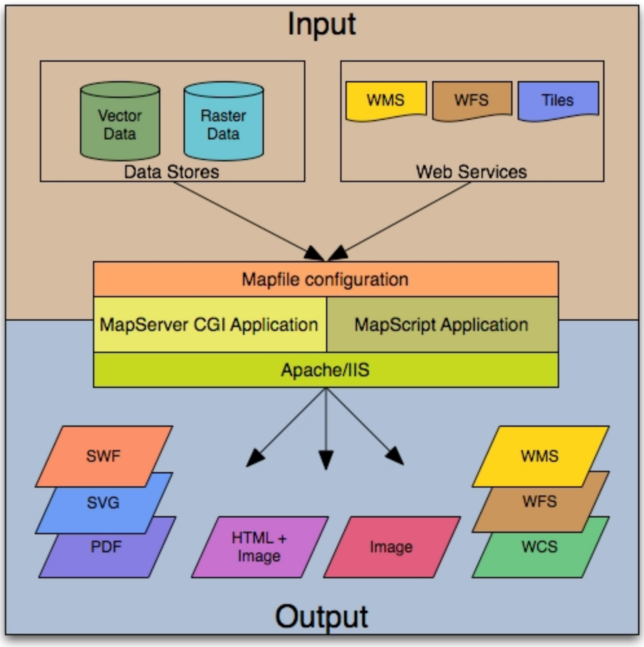


Figure 41: The basic architecture of MapServer applications.

The Mapfile is a structured text file for MapServer configuration. It defines the relationships between objects, points MapServer to where data are located and defines how things are to be drawn.

5.2.2 GeoServer

GeoServer is an Open Source Software server written in Java that allows users to share and edit geospatial data (Geoserver 2012). GeoServer is packaged as a standalone Servlet designed for interoperability; it publishes data from any major spatial data source using open standards. GeoServer is the reference implementation of OGC Web Feature Service (WFS) and Web Coverage Service (WCS) standard, as well as a high performance certified compliant Web Map Service (WMS). GeoServer can display data on any of the popular mapping applications such as Google Maps, Google Earth, Yahoo Maps, and Microsoft Virtual Earth. In addition, GeoServer can connect with traditional GIS architecture such as ESRI ArcGIS. Geoserver reads data in a wide variety of formats from PostGIS, OracleSpatial, ArcSDE to shapefiles and geotiff. It can also generate KML, GML, Shapefiles, GeoRSS, GeoJSON and multitudes of other formats. Otherwise, GeoServer is constructed to store extension data e.g: GeoWebCache, Image Pyramid or Image Mosaics, etc. The architecture of GeoServer interface is shown in Figure 42.

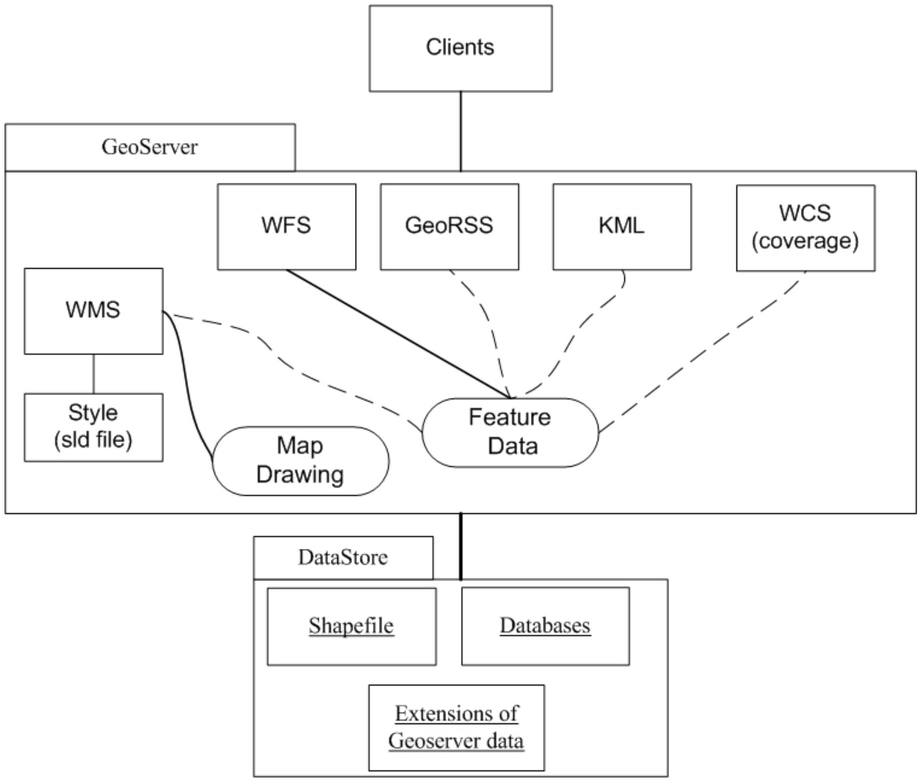


Figure 42: Architecture of Geoserver based on Geoserver version 2.2.1.

a) GeoWebCache

GeoWebCache in GeoServer based on a Java web application used to cache map tiles coming from a variety of sources such as OGC Web Map Service (WMS). It implements various service interfaces (such as WMS-C, WMTS, TMS, Google Maps KML, Virtual Earth) in order to increase process of the map and optimize map image delivery. It can also recombine tiles to work with regular WMS clients.

b) Image Pyramid

Clients render WMS data at every time they are queried for example client sends request to sever in order to zoom in, zoom out many time. They can result in unnecessary and increased waiting times. To solve the problem, image pyramid was one of the effective methods order to optimize maps.

An image pyramid builds multiple mosaics of image; each set at a different zoom level, each tile is stored in a different file. This comes with a composition of many images to bring back the tiles into a single image, it can speed up image. In order to build the pyramid user can use the gdal_retile.py utility, part of the GDAL command line utilities and available for various operating systems.

GeoServer allows loading image pyramid into GeoServer's Raster Data Sources by using ImagePyramid plugin (Fig. 43). User can simply install ImagePyramid extension inside GeoServer, it can efficiently deal with large TIFF with overviews, as long as the TIFF is below the 2GB size limit.

Figure 43: GeoServer's interface with Image Pyramid plug-in

5.3 Client Software

5.3.1 HTML and MapServer

A simple MapServer application consists of geographical data, HTML pages and a mapfile. It defines the map extent, the projection to be used, the data source, the layers that you want to display and how they have to be drawn. HTML pages are the interface between the user and MapServer and they normally sit in Web root. In its simplest form, MapServer can be called to place a static image on an html page. To make the map interactive, the image is placed in an html form on a page. Since CGI programs are 'stateless', every request they get is new. For this reason, every time the application sends a request to MapServer, it needs to pass context information (what layers are on, where you are on the map, etc.) in hidden form or URL variables. The simplest MapServer CGI configuration may include two html pages:

The initialization file uses a form with hidden variables to send an initial query to http server and MapServer;

The template file that controls how the maps and legends output by MapServer will appear in the browser. The template also determines how the user can interact with the MapServer application (browse, zoom, pan, and query).

5.3.2 OpenLayers

OpenLayers is Free and Open Source and it is a project of the Open Source Geospatial Foundation. OpenLayers is a map viewing library written in JavaScript library with no server-side dependencies, so it can be used both with MapServer and GeoServer (http://openlayers.org/). OpenLayers also implements a JavaScript API to incorporate maps from a variety of sources into a Web application. It can support WMS and WFS and other different data sources. Some JavaScript APIs make easy to interactive with user interface such as jQuery, ExtJS, etc.

5.3.3 Mapfish

An OpenLayers based client can be enriched with the use of MapFish (http://mapfish.org/). MapFish is a flexible and complete framework for building rich Web-mapping applications. MapFish is based on Pylons Python Web framework, extending it with geospatial-specific functionality. It also provides a complete JavaScript toolbox, a JavaScript testing environment and tools for compressing JavaScript code. The JavaScript toolbox is composed of the ExtJS, OpenLayers, GeoExt JavaScript toolkits and specific components for interacting with MapFish Web services.

5.4 istSOS (SOS software)

istSOS is acronymed from Investigation of the Sensor Observation Service. Basically, istSOS software is a Senser Observation Service software was developed by (Cannata and Antonovic (2012). It is developed in Python language and based on the Apache web server though the mod_python package (http://www.modpython.org), it relies on the PostgrSQL/PostGIS database for data storage (http://postgis.refractions.net/) and using advantage of GDAL library. Currently, istSOS implements the core profile including:

- The mandatory operations: GetCapabilities, GetObservation, DescribeSensor.
- The transactional profile is implemented by following operations: RegisterSensor, InsertObservation.
- Additionally, the following operations are implemented: getFeatureofInterrest, getObservationBBox, getObservationQualityIndex.

The istSOS insist sos.py file that is executed by Apache server every time a selected URL is accessed. The first sos.py load istSOS module and other requirement libralies and the configuration file (sosConfig file). The second step is using sosDatabase in order to open a connection with the configurated in istSOS data base. And then execute sosFactoryFilter to request and retrieve a filter object, execute sosFactoryResponse function to pass the filter, database connection and then retrieve a response object containing requested information. Finally, execute a sosFactoryRender function by passing the response object and getting the SOS response as XML format. And in the istSOS software packet also includes SQL file, the file has the schema shown as bellow (Fig 44):

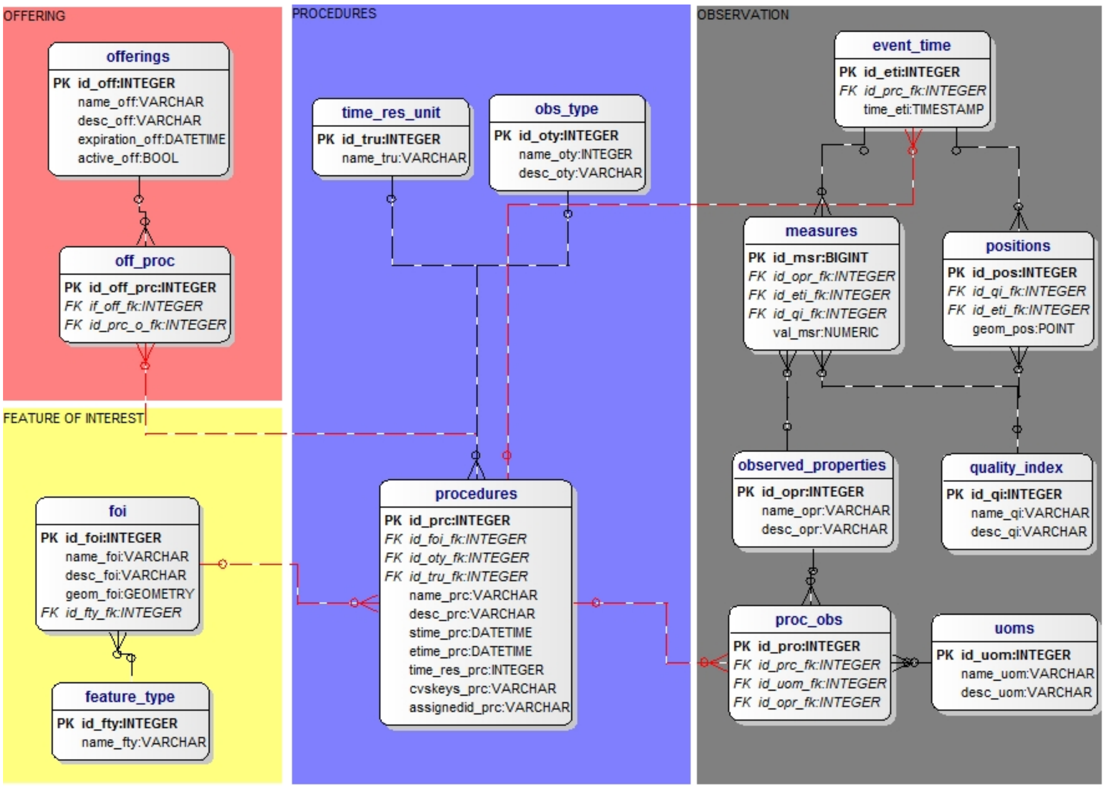


Figure 44: main SQL data table schema of istSOS software

Offering: is designed in case to use one or more sensors at the same station. Offering is analogous to a “layer” in Web Map Service. Example in the same station existing two sensors one is temperature and the other is wind speed. When client request for weather information at the station, they should probably be included in the same offering. If the client asks for wind speed only, air temperature only, or both, the time and location of the results should be the same.

Feature of interest: The Feature of Interest operation allows clients to retrieve certain features of interest of observation it includes: feature_type and FeatureOfInterest (foi) tables identifier of the feature type, list of feature types can be used when adding a new feature of interest which is associated with an observation inserted through InsertObservation. The foi includes important factors such as: name of feature of interest (e.g.: name of feature of interest), geometry of observation (e.g.: Location of point), identify of feature of interest (id_of_foi).

Procedure properties of one or more sensors including time register unit (time_res_unit) for register time unit of observation, observation type (obs_type) is to return particular phenomenon and finally procedures table, it allows users to insert and retrieve their information about one or more procedures, including sensor systems, instruments, simulators, etc.

Observation It contains 7 entity tables: event_time, measures, position, quality_index, observer_properties, proc_obs, uoms. The event_time presents exactly time of the observations; measures contains set of value of a quality; position for containing position of each time observation obtain its value of a quality (eg: in case remote measurement equitment); quality_index defines quality index of observations, observer_properties is designed in order to store properties of various of observations (e.g.: rainfall, water level, discharge, etc.); proc_obs manage reference index of properties such as procedures, observer_properties, uoms; uoms is built in order to defines unit of measurement such as: meter, millimeter, degree, etc.

5.5 ZOO-Project

A WPS software implementation used for processing our services is ZOO-Project. ZOO-Project (http://www.zoo-project.org/) includes ZOO Kernel and ZOO Service Provider (Fig. 45). ZOO Kernel is the main core of ZOO Project. It is a server side implemented in C language. Kernel makes possible to create, manage and chain WPS 1.0.0 compliant Web Services by loading dynamic libraries, handling them on-demand. Otherwise it can easily connect to geospatial libraries, scientific models, common cartographic engines and spatial databases. ZOO Service Provider is a couple of Service Shared Objects (SSO) and one metadata ZOO configuration file (.zcfg) per each provider service. The SSO is the file containing the function corresponding to the related process. The main difference among ZOO Project and earlier WPS implementations is the multi-language support. ZOO-Kernel is written in C, but Web Services can be programmed in C, FORTRAN, Python, Java, Perl, PHP and JavaScript.

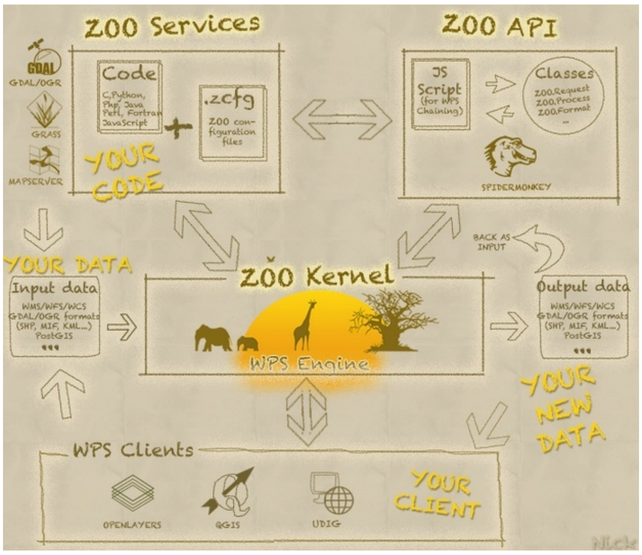


Figure 45: Architecture of ZOO-Project software

5.6 Open Source Libraries and frameworks

5.6.1 GDAL

GDAL (http://www.gdal.org/) is a translator library for raster geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats. It also comes with a variety of useful command line utilities for data translation and processing.

5.6.2 OGR

The OGR Simple Features Library is a C++ open source library (and commandline tools) providing read (and sometimes write) access to a variety of vector file formats including ESRI Shapefiles, S-57, SDTS, PostGIS, Oracle Spatial, and Mapinfo formats.

5.6.3 ExtJS

ExtJS is a pure JavaScript application framework for building interactive web applications using techniques such as Ajax, DHTML and DOM. Ext JS supports a rich set of form controls (or "widgets") for use within web applications such as: numeric fields, list box, combo boxes, grid control, tree control, toolbars, etc. Especially, ExtJS can be composed with Openlayers or Mapfish in order to enrich and support useful web applications for user/developer.

5.6.4 GeoExt

GeoExt is a JavaScript library that provides a base for building up rich Web-GIS applications. It combines the Openlayers library with ExtJS. Across browser, Javascript library can create rich Web GIS application. GeoExt provides a suite of customizable widgets and data handling support. GeoExt is easy and useful tool to speed building applications for vewing, editing, styling geospatial data.

5.6.7 JQuery

jQuery is free open source software, licensed under the MIT License. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

Chapter 6: IMPLEMENTATION

6.1 Intelligent Geoportal

This section presents the Intelligent Geoportal named HDI4NVN, the aim is not only to improve WebGIS existing but also to extend functionality such as: supporting numerous of base maps, search and query, managing and visualization the data from gauge and sensor for long periods of time, presenting data in tables and charts. Otherwise WPS play important role into building statistical. All of the Web service integrated inside HDI4NVN system is built in standardized ways.

Free source remote sensing images available to download via internet, after image processing they are available to be a based map of WebGIS. Beside, Image pyramid and GeoWebCache are powerful tools for increasing a speed of rendering time.

Historical data of hydro-meteorological data can be managed by SOS service relying on PostGIS/PostgreSQL. The RegisterSensor request allows user to register new sensors into the SOS system. The InsertObservation facilitate the sensor in registering new observations for one or more sensors already registered in SOS system. The use of GetObservation request allows clients to obtain the desired observations and from these results of request they are possible to create a statistic chart or tables.

HDI4NVN system fulfill the requirements of an general Intelligent Geoportal adapting for rapid access hydrological data and handing flexibilities functionalities in study area in case north of Vietnam.

6.1.1 Map Generation with WMS

To generate the map we use Web Map Service OGC standard, it need at least one server side and client side for interacting between user and map. In the server side, GeoServer software is used as the Server side, all the web client is hosted by Apache Server Platform.

a) Image Pyramid

As the early work of [Sarawut. et al, 2004], the GIS data overlaid on satellite image over Web GIS interface, with many advantage of satellite image it is visible to represent the world as Digital Elevation Model, slope, aspect, or land use/cover. Data used from various scenes of satellite images for whole study area, the data amount of the mosaic image from 12 Landsat TM and 4 SRTM scenes they are quite large data, and time for rendering a map is limit, there is need to reduce the processing time for each time rendering map at different zoom level. In this case, we use pyramid image functionality in Geoserver version 2.1.2 to manage and gdal_retile library to build pyramid image, our tile per each image as show in Table 22.

Name	Scenes	Level	Resolution (meters/pixel)
DEM (SRTM)	40	4	~ 83-1300
Aspect (SRTM)	25	4	~75-1200
Landsat TM	96	8	~25-3000

Table 22: Pyramid Images.

b) GeoWebCache

As most Web mapping clients render WMS data at every time they are queried, they can result in unnecessary and increased wait times. To deal with this problem, we use GeoWebCache to optimize above experience by saving (caching) map images, or titles. As new maps and tiles are requested, GeoWebCache intercepts these calls and return pre-rendered tiles if stored, or call the server to render new tiles if it is necessary. In our Web GIS, we use GeoWebCache configuration tool of Geoserver software version 2.1.2.

c) Creating the Map Viewer

The OpenLayers Map is designed to display each of its layers at different levels of Scale or Resolution. In order to create the viewer, the first must create a map; the next step is to add a layer to the Map. The OpenLayers.Layer.WMS are used to display data from OGC Web Mapping Services.

First, the construction of the map should be in a custom projection. The map is constructed in a geographic projection “EPSG:4326” (EPSG is widely used database of Earth ellipsoids, geodetic datums, geographic and projected coordinate systems, units of measurement, etc.), with units in decimal degrees see at (http://www.epsg.org/). The imagery data/map is limited by boundary coordinate (100, 20, 110, 25). The final sets of custom options are related to map resolutions by the way set the maxResolution, minResolution property. Finally, zoom levels instead of the default Openlayers’s value needs to set.

GeoServer provides support for (OGC) Web Map Service (WMS) versions 1.1.1 and 1.3.0. This is the most widely used for generating maps on the web, and it is the primary interface to request map products from GeoServer. GeoServer’s WMS implementation fully supports the standard, it includes a wide variety of rendering and labeling options, and is one of the fastest WMS Servers for both raster and vector data.

Exmaple of scripts in client side in order to request WMS from Geoserver side:

```
var aspect = new OpenLayers.Layer.WMS( "Aspect ",
    "http://131.175.143.99:8085/geoserver/wms",
  {  srs: 'EPSG:4326',
     layers: "cite:aspect_pyramid",
    transparent: false
  } );
```

In our system all raster and vector map can be shown in HDI4NVN web side as the (Fig. 46), (Fig. 47) presents user's main menu to operate map layers for the list bellow:

- Raster : Hill shade, Google Map, Aspect, Composite of LansatTM data, Classification of DEM
- Vector

VN Map includes:

- Based layer (map of stations, bordering, road, point of interest, contour line, commune, province, river basin);
- Soil&Land (geo-mophology, vegetation cover)

VN Stations:

- Hydrology station
- Meteorology station
- Rain fall insoline
- Da river Station
- Lo river Station
- Thao river Station

River regions:

- Water body
- River branch
- Drainage system
- Main river
- Da Drainage System
- Da River
- Thao Drainage System
- Thao River
- Lo Drainage System
- Lo River

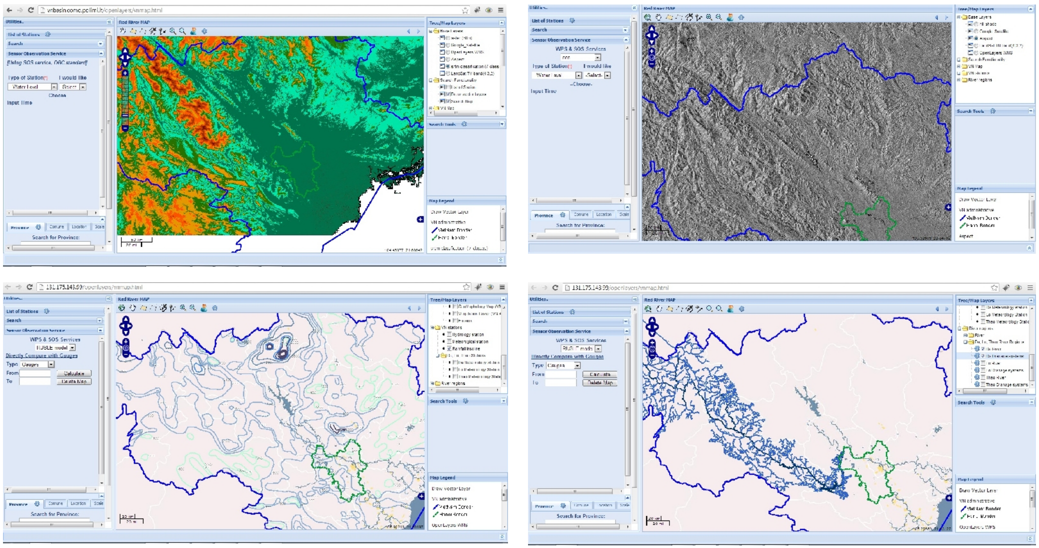


Figure 46: Showing results gained by using Image Pyramid and GeoWebCache using GeoServer in server side and Openlayer in client side. Top-Left side is classification of DEM in 7 classes; Top-Right side is aspect map, in the top layer of left and right maps is administrator boundary in. lower-left rainfall contour map, lower-right Drainage system and Da main river map.

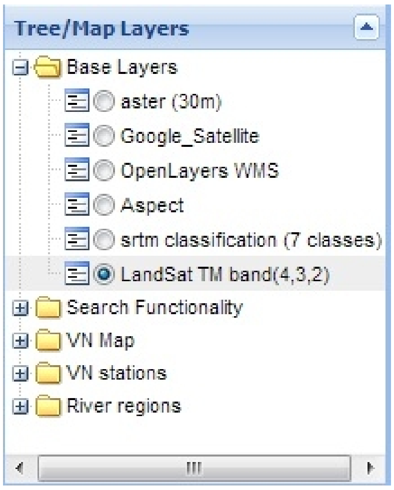


Figure 47: Main map layers interface of HDI4NVN Web-GIS system

d) Rendering vector and raster data using SLD

In order to visualize map or view a specific data with classification based on colour changes we use GeoServer with SLD OGC to render the dynamic location on the map such as contour line, difference between geomorphologies, elevation changes and so on. A SLD of OGC for describing the appearance of map layers in flexible ways, it is capable of describing the rendering of vector and raster data. A typical use of SLDs is to instruct a Web Map Service (WMS) of how to render a particular layer. SLD is a type of programming language, not unlike creating a web page or building a script. SLD language is written in tags e.g in the bellow quote from our SLD code, it is to classification Geomophology map in Fig. 48. Using The Style Editor (Fig. 49) of Geoserver's interface user can easily configuring a style's name and code.

```
 <Rule>
  <Name>Grey_Soil_Ferrallitic</Name>
  <Title>Grey Soil Ferrallitic</Title>
  <ogc:Filter>
   <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>DAT21C1_ID</ogc:PropertyName>
      <ogc:Literal>31</ogc:Literal>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
  <PolygonSymbolizer>
    <Fill>
      <CssParameter name="fill">#A0522D</CssParameter>
      <CssParameter name="fill-opacity">0.7</CssParameter>
    </Fill>
  </PolygonSymbolizer>
</Rule>
```

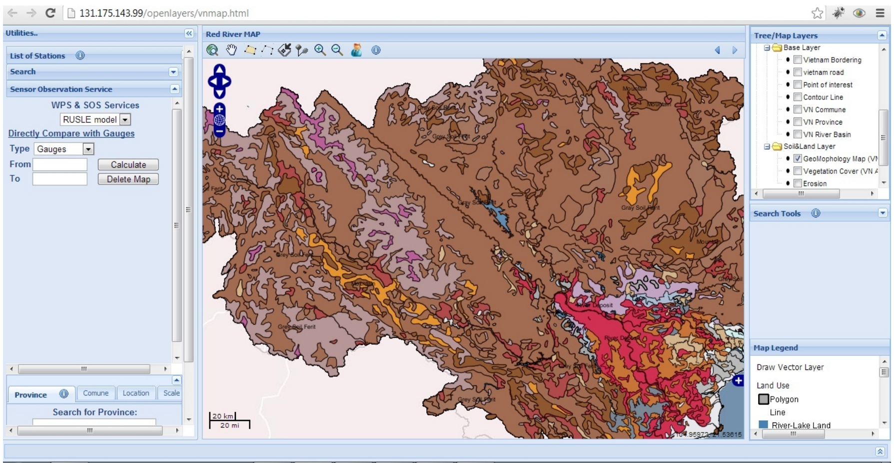


Figure 48: Geo-morphology map using SLD in order to classification different type of rocks

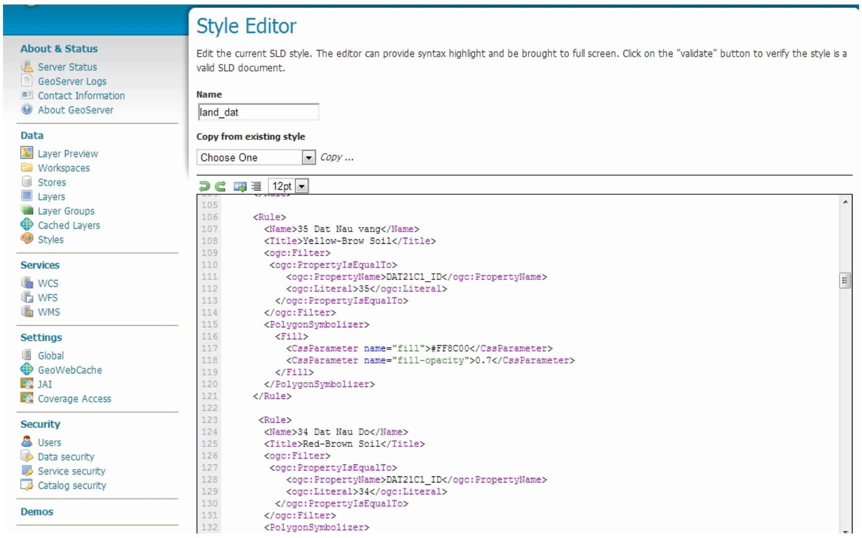


Figure 49: Style Editor's interface of GeoServer

6.1.2 Query and Search

a) Query a map

Currently, to query map features, WMS provides an easy tool to query feature info on a map by using GetFeatureInfo. WMS GetFeatureInfo control of Openlayers library allowing to use a WMS query to get information about a point on the map. The information may be in a display friendly format such as HTML or GML, depending on the server's capabilities and the client's configuration. The control handles click or hover events, attempts to parse the results using an OpenLayers.Format, and triggers a 'getfeatureinfo' event with the click position (Fig. 50).

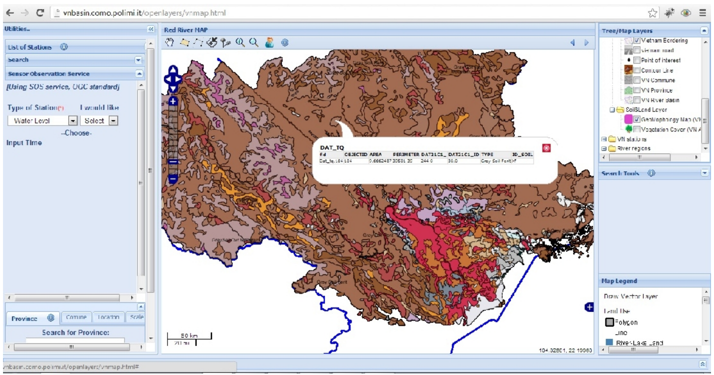


Figure 50: Get feature information at click position

b) Search

In order to build searching functionality for whole 25 ESRI shapefile, it is quite simple way to use WFS GetFeature; because WFS GetFeature can accept query of feature within a bounding box and return the query results in GML format. Most of client software such as P.mapper, Chameleon, OpenLayers primarily support WMS and WFS. In this case, using Openlayers is quite easy to approach GetFeature. Base on Openlayers and Geoext library such as class Geoext.form.Panel to create searching form and protocol in order to use a form to trigger search requests through an OpenLayers.Protocol. Then run action builds an OpenLayers.Filter from and passes this filter to its protocol's read search request. The form fields must be named after a specific convention, so that an appropriate OpenLayers.Filter.Comparison filter is created for each time request. The comparision filter such as: OpenLayers.Filter.Comparison.EQUAL_TO,OpenLayers.Filter.Comparison.LESS_THAN, etc.

User can define input parameter (Fig. 51) as filter criteria on web's interface and request filter service to find specific place such as: filter for name of provinces, population, or river names, etc.

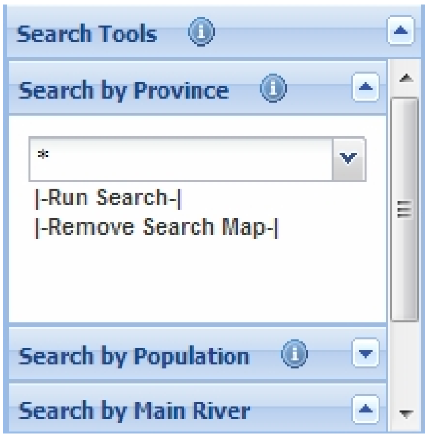


Figure 51: GeoExt form panel applying for search functionality in case of Search for Province

The result is present in a new map and the deferent color corresponding to specific place where client need to show in the client interface (Fig. 52).

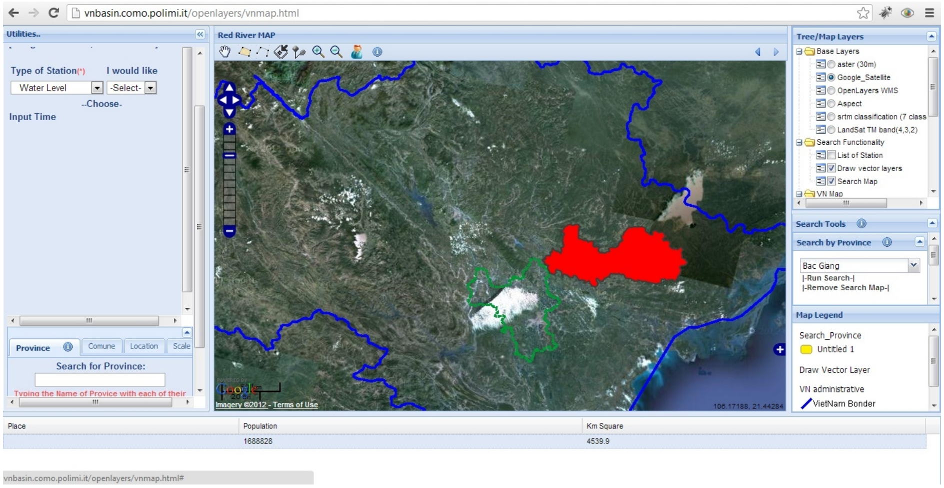


Figure 52: Search map by using Openlayers and GeoExt

c) Creating a Synchronized Grid and Map View of WFS Features

GeoExt uses most of its WFS support from OpenLayers, it does provide though is the GeoExt.data.FeatureStore, it is a store containing GeoExt.data.FeatureRecord entries that optionally synchronizes with an OpenLayers.Layer.Vector. If user just wants to display features in a grid. So, developer can invoke this library to show feature attributes in a grid in a very easy task (see Fig. 53).

User can use OpenLayers.Protocol.WFS or GeoExt.data.ProtocolProxy. To use these, user is allowed to select a layer in the tree, and the grid would automatically display its feature attributes, and the features could be selected in the grid and the map. The code below is example of OpenLayers.Protocol.WFS in case Server-side is Geoserver and GeoExt.data.ProtocolProxy code what we have implemented:

```
var protocol=new OpenLayers.Protocol.WFS({
            srsName: "EPSG:4326"
            ,url: "http://131.175.143.99:8085/geoserver/wfs"

            ,featureType: "rain_station"
            ,featureNS: "http://www.openplans.org/topp"
            });

var proxy_rain_station = new GeoExt.data.ProtocolProxy({
            protocol: new OpenLayers.Protocol.HTTP({
            url: "data/rain_stations.json",
            format: new OpenLayers.Format.GeoJSON()})});
```

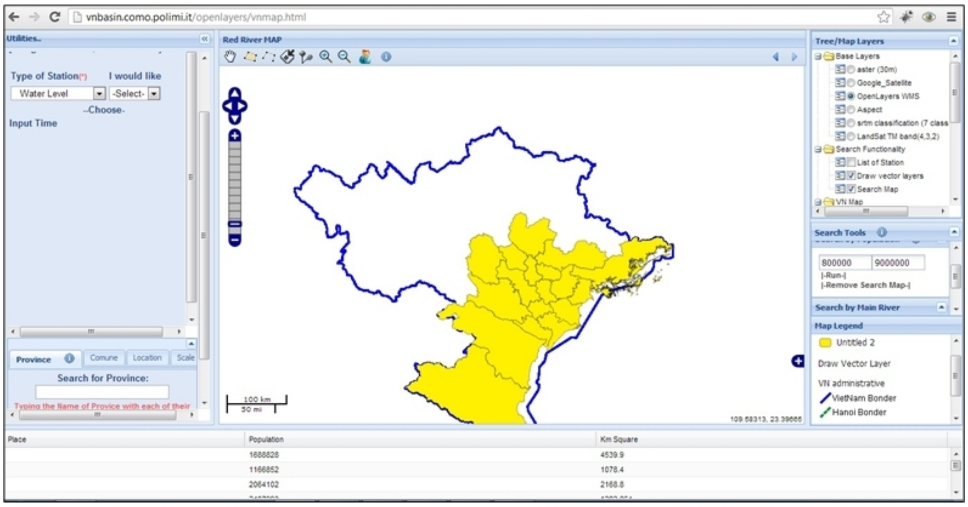


Figure 53: The result of search for population larger 800.000 and smaller 9.000.000.00 all feature attribute is presented in a grid (see at the bottom of the web interface).

6.1.3 Sensor Observation Service

In recent years, SWE standards have been applied in various projects, as (Chung. et al, 2009) presented standard way to access to observation form sensors, (K. Walter, 2009) used SWE as the method for data access ,communication and visualization in order to procedure a low-cost, interoperable and performant early warning system. (A. Broering, 2010) presented approach of SIDs for realizing a vision of sensor plug & play and making sensors on-the-fly available on the Sensor Web. To visualize and analyze sensor data both with respect to time and space to build up Web Geoservices for monitoring the environmental impact of large structures (M. A. Brovelli, 2010). New software supports for implementing SOS in the standard way is developed by (M. Cannata and M. Antonovic, 2010) and also presented its application in real case.

Base on our data available, to mange 28 stations including their sensors/gauges as (Table 23). Thanks to WRP to support many data to us. To build up Web GIS adopting SOS as a synchronic system, it should satisfy the list bellows:

- Manual registration or on-line sensors and their observation by various time intervals;
- Distribution of observed data;
- Mapping of sensor location;
- Report of sensor properties;

- Extracting measured data in time series with frequency day, month, year (shows in text or plots);
- Multi selections of various sensors and time intervals (Client Interface);

	Daily	Monthly	Yearly
Rainfall	19	19	19
Discharge	9	9	9
Evaporation	15	15	15
Inflow/Reservoir	1	1	1
Water Level	6	6	6
Total	50 for 28 stations 150 sensors	50 for 28 stations	50 for 28 stations

Table 23: Table presenting number of sensors need to be insert in SOS system

Our system implements not only GetCapabilities, DescribeSensor, GetObservation but also optional transactional profile. Because we need to register new sensors and their observation data also, transactional profile provides an access point for data uploading (M. Cannata and M. Antonovic, 2010). The RegisterSensor request allows to add into the SOS system a new sensor by supplying the sensor description (SensorML). The InsertObservation facilitate the sensor in registering new observations for one or more sensors already registered in SOS system and the O&M encode elements containing and presenting the measured values. To get result from a specific sensor in a period of time such as day, month, year, GetObservation request allows obtaining desired observation value. Our system architecture is outlined schematically in (Fig. 54):

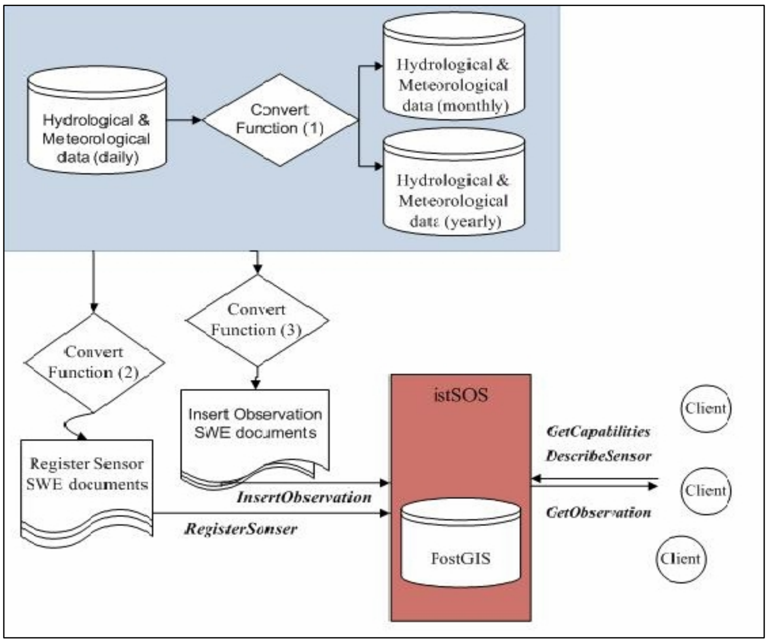


Figure 54: Our implementation flow chart. Convert Function (1), (2) and (3) are small code written in Python in order to convert data to specific format.

To enrich data, form daily data (original data), we convert daily data to become monthly and yearly data. Improving processing time of data query and web service process is very important point. In this case we sub-divide (see convert function [1] in Fig. 54) daily data to become monthly and yearly data set.

a) Register a new Sensor

The RegisterSensor operation allows sensor data providers to register new sensors via the service interface by sending a SensorML sensor description to the SOS system. Historical series of rainfall, discharge, evaporation, water level, inflow data from 1956 to 2004/2008 at daily, monthly and yearly measured by gauges/sensors at hydrological or hydro-meteorological stations (Table 23) are registered as a sensor into PostgreSQL/PostGIS by using RegisterSensor operation based on istSOS software (Fig. 55).

SOS system understands SensorML language; in order to automatically register large number sensors into SOS system we have written a function by using python language named register.py to convert all sensor properties to be SensorML language and then insert them SOS system, see the Pseudo code bellow:

```
Begin
For i=1 to 150 do {  //number of sensors without PERSIANN data
            Sensor[i]=sensors[i]; //read sensor_id, name
            Foi[i]= fois[i] ; //read [sensor name, (lat,long)] from file

            //read [responsibility of sensor, unit of measuring] from file,
            feature_of_interest_id, phenomenon_id,
            starting_date_of_measure, interval_unit
            Phenom[i]= Phenoms[i] ;
            //read sos description, epsg code, etc.
            sos[i]= read_sos_properties[i] ;
            Create_sensor_id[i]; //initiate sensor_id for each specific sensor
            Write_sensor_id[i]; //write sensor_id to the file
            //generate_SensorXML document
            xml =  """<?xml version="1.0" encoding="UTF-8"?>
                        <sos:RegisterSensor
                        ...
             <gml:description>%s</gml:description>""" %
            (sos[description]) + """
            <gml:name>%s</gml:name>""" % (sensor[i][name]) + """
                       ...
            <input name="%s">""" % (phenom[i]) + """
                       ....
            <swe:uom code="%s"/>""" % (phenom[i]['units']) + """
                       ....
            <gml:Point srsName="EPSG:%s">""" % (sos['epsg']) + """
             <gml:coordinates>%s,%s</gml:coordinates>""" %
             (sos[i]['lat'], sos[i]['lon']) + """
                       ....
            xml += "</sos:RegisterSensor>"
            return xml

}

register()
Begin
            Connect_to_istSOS_system();
            Send_request_to_SOS_system(xml);
End;
```

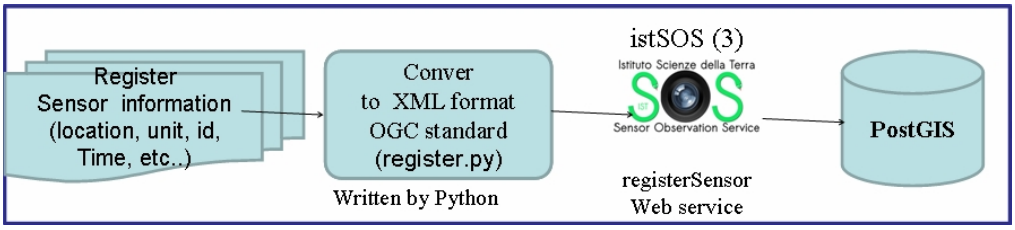


Figure 55: Architecture of RegisterSensor in SOS system

User can send GET- GetCapabilities request as following request parameters, server respond to client all information of sensors already registered as in (Fig. 56):

http://localhost:8099/sos/sos.py?request=getCapabilities&
section=serviceidentification,serviceprovider,operationsmetadata,contents&
service=SOS&version=1.0.0

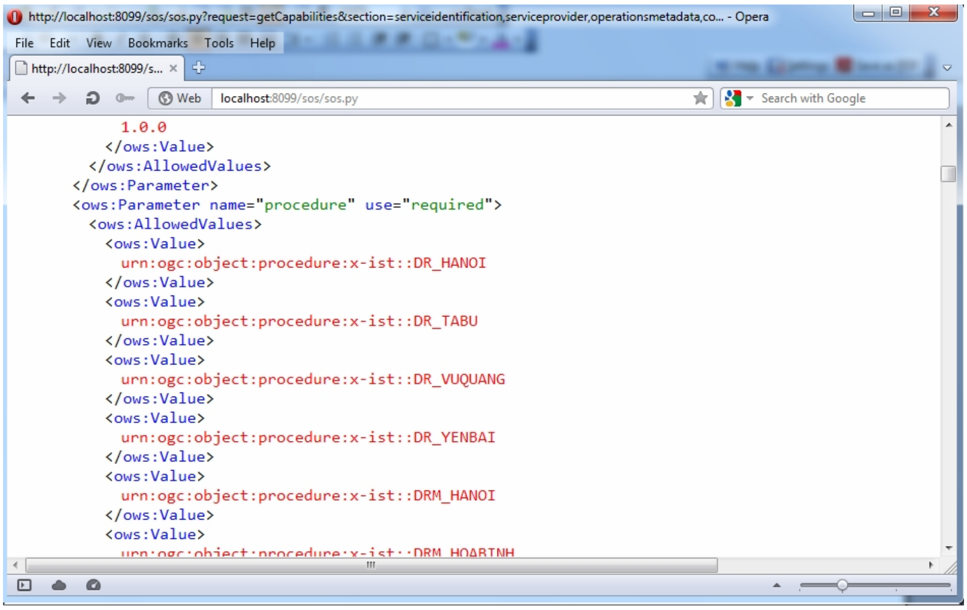


Figure 56: Respond from server of getCapabilities request

b) Insert Observation

The InsertObservation operation allows the client to insert new observations for the system. This is a request to an SOS to perform the Insert operation. The request includes the sensor ID that was obtained from the

RegisterSensor operation. The observation must be encoded in XML following the O&M specification, our system architecture is outlined schematically in (Fig. 57). As the above section “Register a new Sensor”, we also wrote a function in python language named insert_data.py in order to insert more than 40 years of daily, monthly, yearly intervals of all 150 sensors (Table 23) into SOS system by using istSOS software. See the Pseudo code bellow:

```
InsertObservation_XML(sensorID, values)
Begin
// function to connect to database management system (DBMS)
        Connect_to_DBMS();
// get sensor_id, units, epgs, lat, lon, etc from /DBMS
        Get_information(sensor, foi, phenom)

xml = """<?xml version="1.0" encoding="UTF-8"?>
<sos:InsertObservation
xml += "<AssignedSensorId>%s</AssignedSensorId>" % (sensor_ID)
        ....
<swe:values  %s """ % (values) + """</swe:values>
xml += "</sos:InsertObservation>"

return xml;
End
//Read file
Insert_data ()
Begin
        Open_folder();
        For sensor_ID in [file_name_1 to file_name_n]
                Read_file(sensor_ID.xls);
                for line in lines:
                        time=get_dat_time();
                        value=get_data();
                        values+=time+','+value+'@';

                InsertObservation_XML(sensorID, values);
                Write_to_file(xml);
                Send_request_to_SOS_system();
End
```

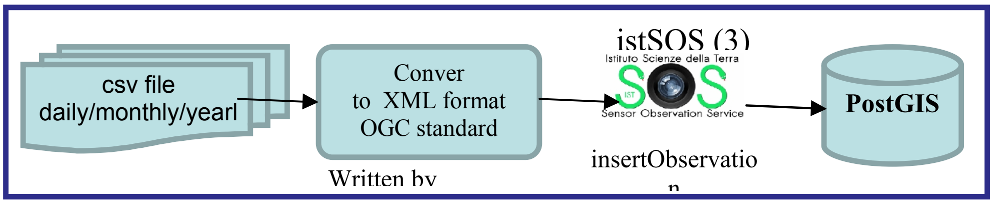


Figure 57: Architecture of InserObservation in SOS system

After run the request, the result of request is to create series of XML files corresponding to each sensor with their observation's values see an example in (Fig. 58).

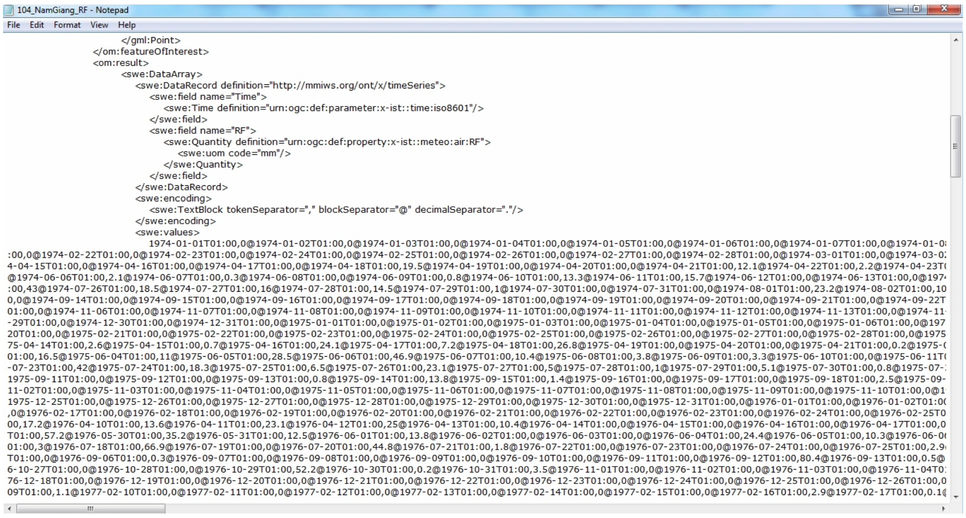


Figure 58: XML file contains observation values of sensors corresponding to InsertObservation service

c) Get Observation

The GetObservation operation is designed to retrieve observation data structure according to the O&M specification and XML encoding, observation data includes time-series data from one or more sensors on one or more platforms. Bellow is one example of GetObservation's aspect:

```
http://localhost:8099/sos/sos.py?
service=SOS&
version=1.0.0&
request=getObservation&
responseFormat= text/xml;subtype='sensorML/1.0.0'&
```

offering=temporary&
observedProperty=RFM&
eventTime=1999-04-15T01:00:00+00/2000-12-15T01:00:00+00

In the example observedProperty=RFM is rain fall data in monthly, eventTime=1999-04-15T01:00:00+00/2000-12-15T01:00:00+00, requesting time of data start from 1AM on 15-Apr-1999 to 1 AM on 15-Dec-2000. Client sent GET or POST request to the SOS system, SOS server responds to client O&M specification and XML encoding document (Fig. 59).

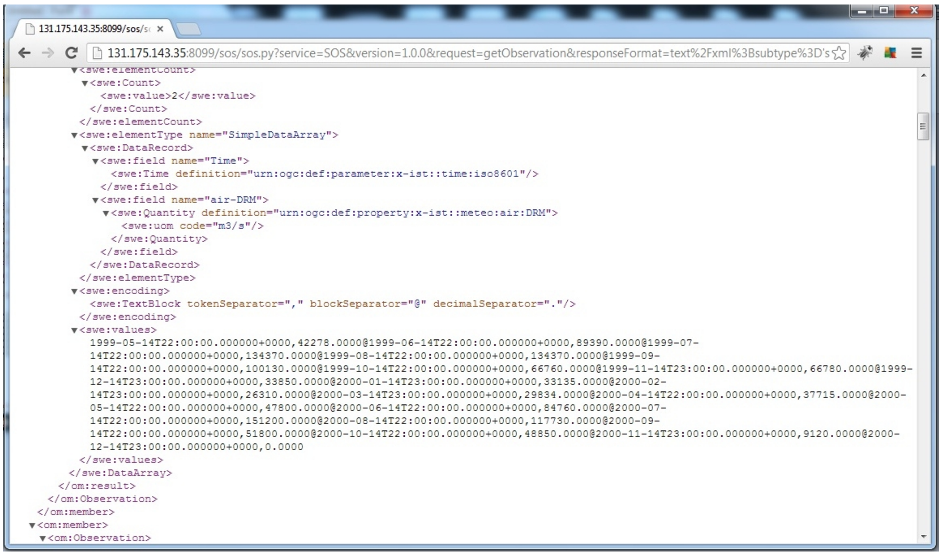


Figure 59: O&M specification and XML encoding of GetObservation's respond.

To represent the time-series of data over the user interface, client-side needs to be configured as the filter to parse XML encoding in order to present data in flexible way. The tool for developing in client-side is Openlayers, Openlayers supports OpenLayers.Request.GET method to send an HTTP GET request. Additional configuration properties are document in the issue method, with the method property set to GET. An issue is to create a new XMLHttpRequest object, open it, set header, bind a callback to done state and send any data. In client side we have already implemented following system architecture is outlined in the (Fig. 60).

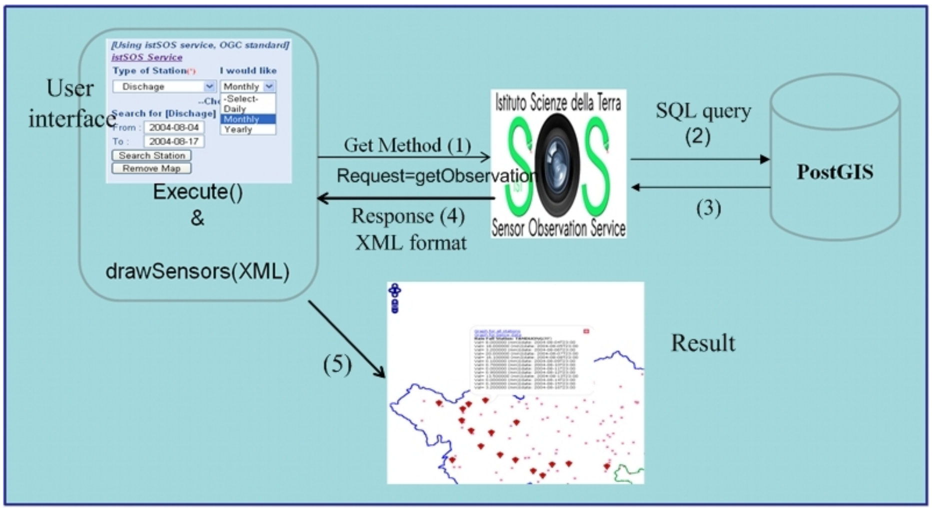


Figure 60: System architecture of GetObservation in both client and server sides.

Main menu of SOS service was mainly designed by using JQuery (Fig. 61). User can freely select their own specific parameters such as: type of sensors (discharge, rainfall, water level, etc.) and time intervals (daily, monthly and yearly). Finally user click to button "Search Station" it means: user send a GetObservation request to SOS system. SOS system represented by istSOS software, the main function of istSOS is as a filter in order to transform parameters from GetObservation request to be come necessary information in client side. At fist, istSOS check all syntax of request and number of parameters, then create a connection to DBMS, retrieve needed information from DBMS corresponding to requirements from parameters, finally server responds to client a result of request in XML encoding.

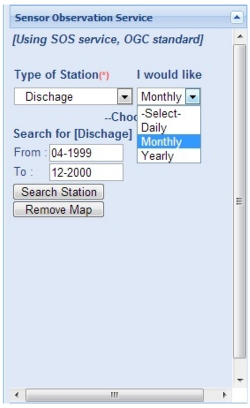


Figure 61: SOS service main menu

The aim of drawSensor(XML) function in (Fig. 60) is to parse of XML responding from SOS system and restructure them into Openlayers syntax and JavaScript code. OpenLayers supports the OpenLayers.Feature class specifically combines a marker and a lon/lat. Features are combinations of geography and attributes. Marker is shown for the given location (sensor locations) and a popup window will be displayed when the marker is clicked. An HTML div is used for the map display. A map object is created and the controls and layers are added to the map. Every time user needs to know series of data corresponding to client request, it means client can click on one or more markers in order to see popup windows overlay on the map. Popup windows may present series of data, unit of measures, data-time, etc. To solve said problem we use OpenLayers popup features parameter (popupContentHTML) to create popup (Fig. 62).

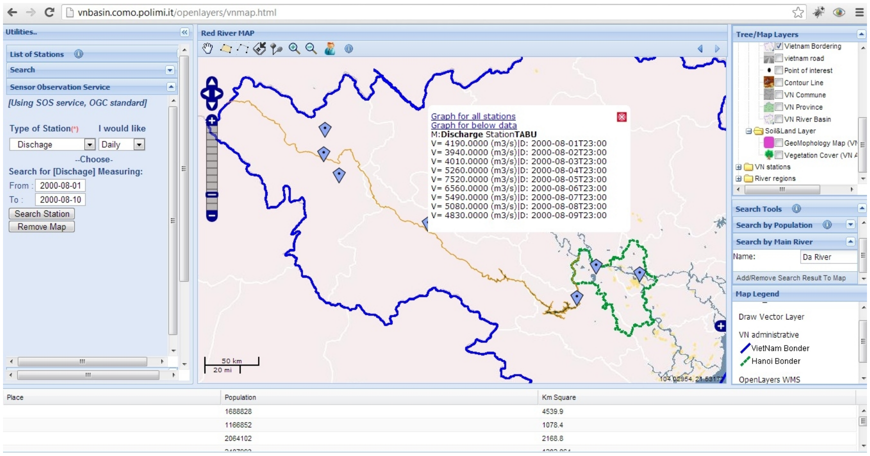


Figure 62: Popup presents time-series of data for Discharge Observation along Da River at TaBu station

d) Chart

Charts are very useful for reporting geospatial data. It often used to ease understanding of large quantities of data and the relationships between parts of the data. Charts can usually be read more quickly than the raw data that they are produced from and they also look more attractive than raw data. They are used in a wide variety of fields. Statistic charts based on Web-GIS technology, users can quickly access the needed data through a browser

Principal of charts applied in our Web-GIS is used for supporting users requesting information about hydrological and environmental data which is invoked by using GetObservation requests in difference ways. Our SOS service functionality gives client the ability to export the map of sensor networks and their statistic chart of the indicator in various types. Web-GIS provides various chart options, including line chart and bar chart Through the display of the charts, users can not only read the statistic data more efficiently but also apply the chart to have general ideas about data to enhance the usability of the data.

Software used to create charts is JS Charts, is a JavaScript based chart generator that requires little or no coding. With JS Charts drawing charts is a simple and easy task, since you only have to use client-side. No additional plug-in or server modules are required. Just include scripts, prepare chart data in XML, JSON or JavaScript Array (Fig. 63).

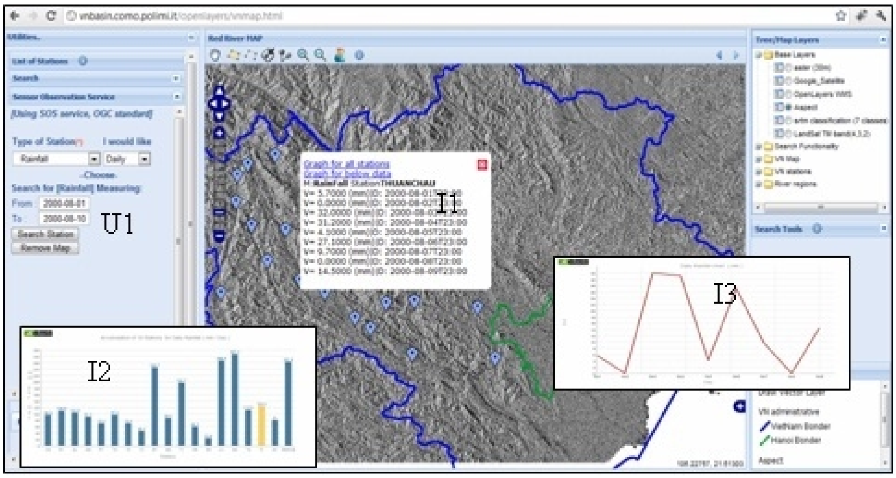


Figure 63: WebGIS with SOS's interface to query data and their results, I1 is popup showing value of interval time corresponding to selection from user's interface (U1). I2 is comparison graph for all sensors have a same measure's properties. I3 is graph for showing different values of observation time in incident station when user clicks on.

e) Potential of PERSIANN data inside Intelligent Geo-Portal

PERSIANN data can be extracted as the detail in the section 2.1.4.2, afterward by using SOS's transaction (RegisterSensor and InsertObservation) user can register and insert sensors into PostgreSQL/PostGIS (section 6.1.3a and 6.1.3b), and it is possible to use those data as the input data to create new specific SOS services in order to do so. Otherwise PERSIANN data in The ESRI ASCII raster format are also available to use as the input data for map overlay. PostGIS is able to manage large number of observation and long time periods.

f) Statistics functionalities by using Web Processing Service

The WPS Interface can request geospatial Web Service, such as WFS, WMS, etc., by calling GET operations of these services. Two different implementations can be considered. The former corresponds to direct transfer of data from client to WPS server: user through the web interface makes his/her request, the client sends the request to SOS, collects the results and then send them to WPS to be processed. This implementation is easier at server side but it has at least two drawbacks: it needs to invoke SOS service then service response and transfer all data following service's data structure, in this case there is no way to avoid redundant data base in some case e.g. we need to know only max/min or average of all data set; it can cause delay in case user request large of data set. In order to build statistical functionality we implement a new WPS service. (Fig. 64) shows architecture of statistical functionality dealing with WPS GET operation, (Fig. 65) shows flow diagram of WPS service.

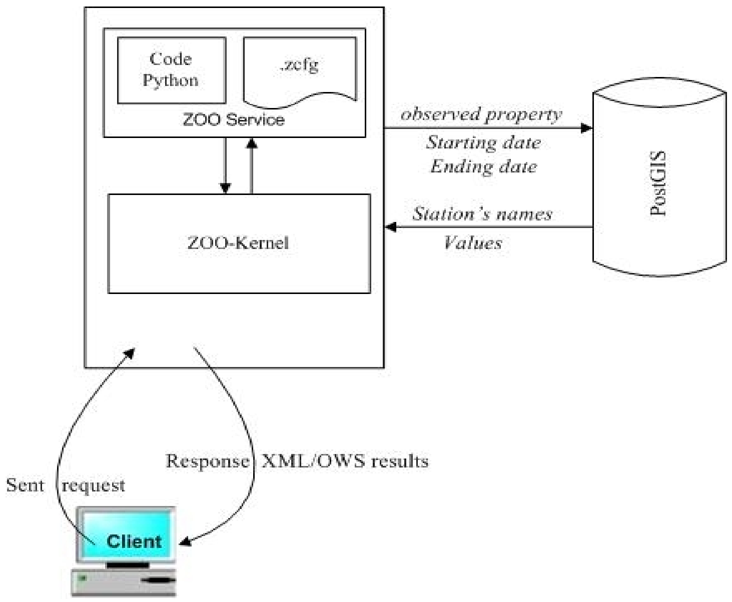


Figure 64: architecture of statistical functionality dealing with ZOO-Project software implementing WPS.

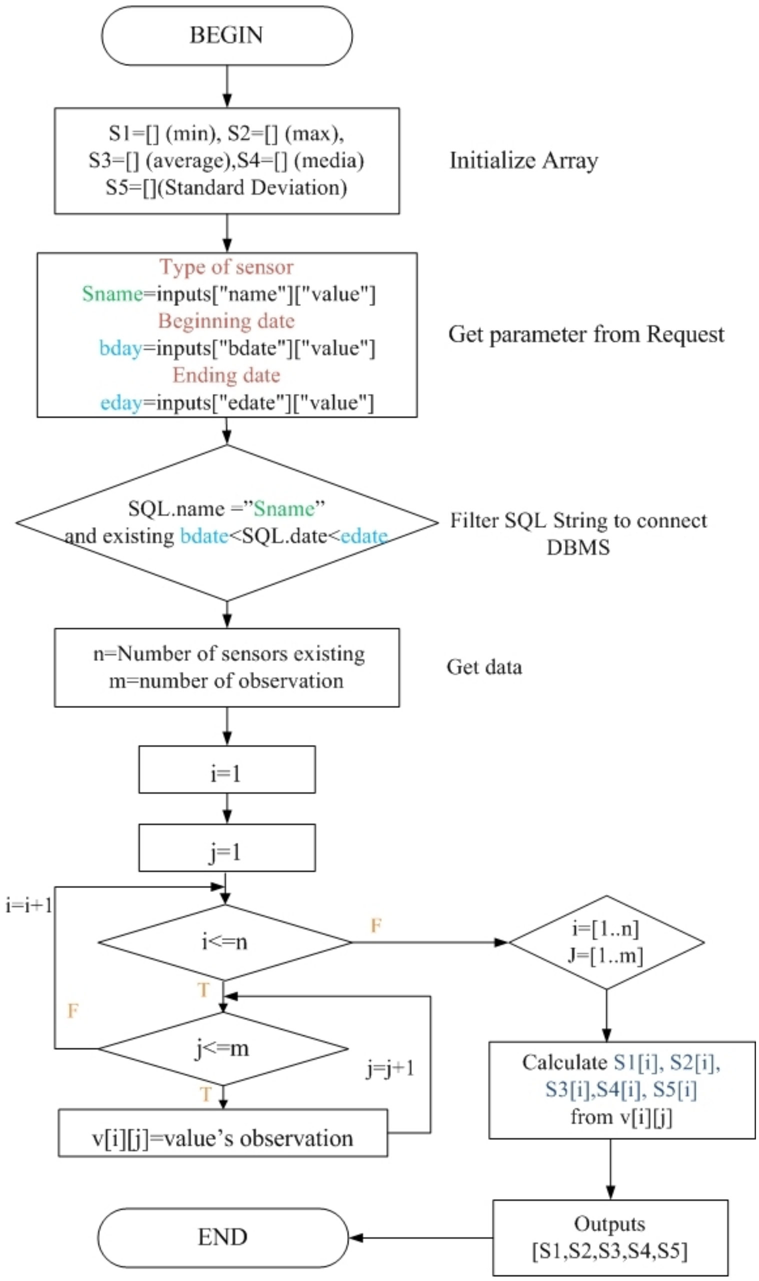


Figure 65: Flow diagram of wps_statistics.py in python code

ZOO configuration file of wps_statistics (wps_statistics.zcfg) service is presented in example of Pseudo code below:

```
[wps_statistics]
….
serviceProvider = wps_statistics
serviceType = Python
….
<DataInputs>
  [name]
….
<LiteralData>
    DataType = string
    <Default>
    </Default>
   </LiteralData>
[bdate]
….
[edate]
….
<DataOutputs>
  [points]
…
[S1]
..
[S5]
  </ComplexData>
 </DataOutputs>
```

The aggregated feature can be requested as following:

```
http://localhost/cgi-bin/zoo_loader.cgi?
request=execute&service=WPS&version=1.0.0&Identifier=wps_statistics&
datainputs=name=RFM;bdate=1999-01-01;edate=2000-01-01
```

Identifier is the name of the WPS service, RFM is observed property represents for monthly rainfall gauges, bdate and edate represent beginning date and ending date of request.

In the client side, client was developed in Openlayers framework in order to parse of XML document respond from server side, using OpenLayers.Request.GET as the section 6.1.3.c, Openlayers allows to create new map of marker and supports client not only to operate the map but client also can click on the map to see popup windows containing minx, max, media, average, standard deviation of series data. The user's interface of WebGIS shows in (Fig. 66).

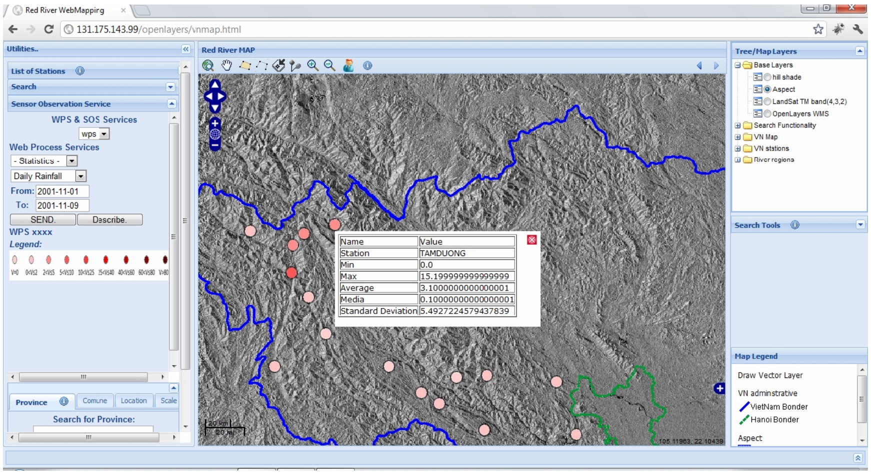


Figure 66: WebGIS interface of statistical functionality, left side is menu for user to operate and select parameters of GET request. Map in the middle shows the various colors of stations based on classification among average values; users can click on the station to see a name of station, sensors, max, min, average, media, and standard deviation of data series.

6.2 Decision support system (DSS)

We not only used historical rain gauges/environmental sensors (RG) but also precipitation data from satellite data (PERSIANN), however PERSIANN data need to be monitored before using because in some case rainfall is effected by topology of terrain. Rain Gauges (RG) data are accurately but qualities of station in this project are spare and maps of stations do not present spatial attribute of precipitation data. On the opposite PERSIANN data can completely present spatial attribute of precipitation. To investigate PERSIANN, we present Decision Support System based on Web Service by means of a direct comparison and of the Revised Universal Soil Loss Equation (RUSLE) model.

a) Direct comparison:

- Compare directly two data sources is interpolate RG and generate classification of PERSIANN data afterward estimating correlation between them;
- Eestimate a correlation and examined a percentage of simultaneous rain r no-rain between PERSIANN and RG.

b) The RUSLE model: requiring huge amount of information and data was handled for both PERSIANN product and RG data to estimate the difference between the two data sources.

6.2.1 Direct comparisons

In this section in order create user's main menu which it supports for user to select not only estimate correlation between PERSIAAN and RG data but also to operate simultaneous rain and no-rain of two data sources, we design "comparison functionality menu" include Direct Compare menu and simultaneous between rain and no-rain main menu as the Fig 67(a). Otherwise client also can select parameter for GET request of interpolation map which is interpolated from point map of RG station see at Fig 67(b)

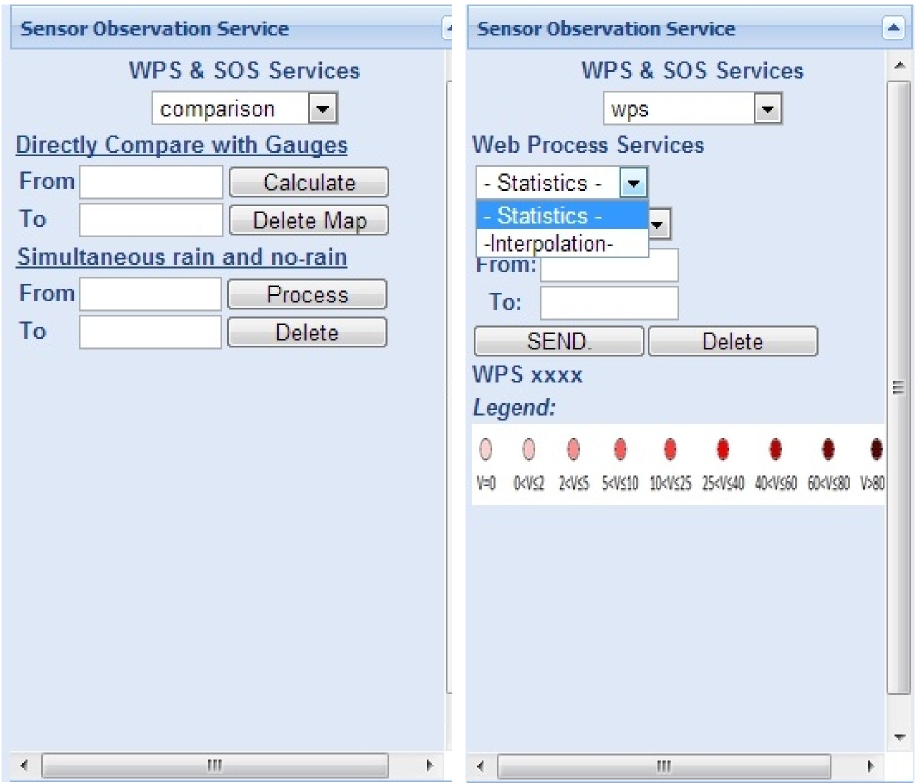


Figure 67: (a) Client's main menu for operating GET request's parameter of Direct comparison functionality. (b) Main menu for operating GET request's parameter of Interpolation functionality

6.2.1.1 Compare directly two data sources

a) Interpolation

WPS can access data by connecting to the PostgreSQL database. From 19 rainfall gauges data set, it is possible to estimate rainfall spatial distribution within an area. Instead of implementing an interpolation algorithm from scratch, we use the GRASS package, a well known and mature GIS. We apply GRASS in order to avoid rewriting a same interpolation function instead of it needs only calling GRASS commands. The specific GRASS command to build interpolation map is Inverse distance (v.surf.idw) command (http://grass.fbk.eu/gdp/html_grass64/v.surf.idw.html). The boundary box of interpolation map is within 20.612° and 22°82 N, 102.65° E and 105.67°10E. ZOO Service can be implemented using the Python language and Python code invokes GRASS functionalities, so we are able to access those functions also from the ZOO service's code. We decided to use the inverse distance interpolation but in principle we can leave the users the freedom of choosing the algorithm they prefer among those available in GRASS. The (Fig. 61) shows architecture of interpolation of RG.

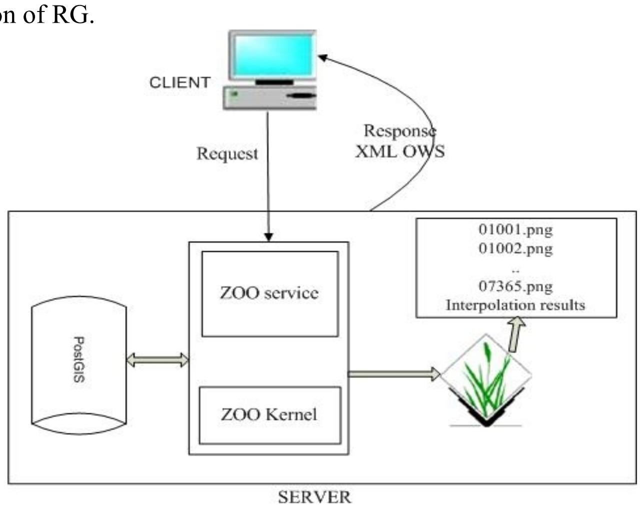


Figure 68: System architecture of interpolation of RG

The interpolation map can request by GET method as following:

```
http://localhost/cgi-bin/zoo_loader.cgi?
request=Execute&service=WPS&version=1.0.0&
identifier= interpolation&
DataInputs=name=RF;bdate=2002-01-25
```

Name=RF is name of the sensor in this case sensor name is Rainfall, dbate is the data needs to be interpolated

Pseudo code of interpolation.py presened is bellow:

```
BEGIN
    Input GET parameters [name of sensor, date-time to interpolate]
    Initiate file_date_time.txt
    Connect to database and extract values correspond to the input date-time
    Write date-time values into file_date_time.txt
    Generate GRASS environment, location, mapset of GRASS
    //GRASS function

    1. run_command('v.in.ascii', input= file_date_time, output='%s' %(fileO), columns='x double precision, y double precision, z double precision, stations varchar(20)', z=3)
    2. run_command('g.region', vect='%s' %( fileO), n='22.82', e='105.67', w='102.65', s='20.612')
    3.run_command('v.surf.idw', column='z', input='%s' %( fileO), output='%s' %(tmpR), npoints='12')
    4.run_command('r.colors', map='%s' %(tmpR), color='gyr')
    5. run_command('r.out.png', input='%s' %(tmpR), output='%s' %(img_date_time))
    // End of Grass's command
    Generate Output
END
```

As the above pseudo code, command 1 is to create a vector map from ASCII points file or ASCII vector file, command 2 manages the boundary definitions for the geographic region. In command 3 is surface interpolation from vector point data by Inverse Distance Squared Weighting. Command 4 creates/modifies the color table associated with a raster map layer, command 5 is to Export GRASS raster as non-georeferenced PNG image.

In the client side, client was implemented in Openlayers, it needs only refer to the image file named img_date_time corresponding date-time's input as the client request. The image file exists inside the server's data store. Client invokes OpenLayers.Layer.Image in order to overlay image which is result of interpolation (img_date_time.PNG) over the Web-GIS interface as a new map. The (Fig. 62) shows result of the interpolation precipitation in day 25-Jan-2002.

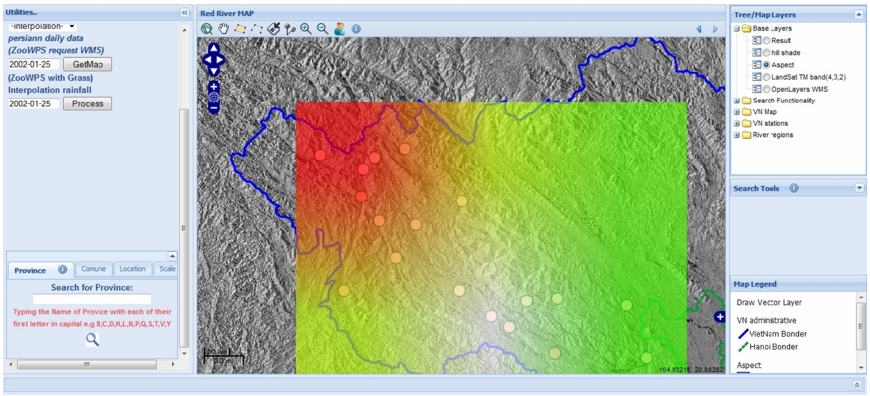


Figure 69: Map of precipitation interpolation using inverse distance method

b) PERSIANN Visualization

Daily precipitation data for whole Vietnam can be downloaded though internet at the web site of PERSIANN product (see at http://chrs.web.uci.edu/persiann/data.html). To get precipitation we used “Country tool”; users can receive their data through their personal email address by using this tool. The data is received in the ESRI ASCII raster format, 0.25° x 0.25° resolution, data available from 2000 to 2007. The name of each file representing each day’s precipitation was defined as YYDOY.asc, where YY stands for year and DOY for the day of the year. To be able to provide visualization of the PERSIANN data, we used the new enhancement of ZOO Project: its support for Standard Web Services Outputs (SWSO see at http://www.zoo-project.org/trac/ticket/34). The idea of SWSO is based on the fact that the WPS main goal is to deal with GIS data; by consequence, execution of many services will result in the production of new GIS data. Or, with WPS 1.0.0, clients may ask the server, by adding a asReference property, to store the result and to provide an URL which can be used to access it later. So, for each services which is producing GIS data as output, ZOO-Kernel handle the WMS, WFS and WCS (when applicable) publication automatically by using GDAL/OGR library for accessing raster and vector properties and the MapServer library for publishing Standard Web Services. Note, the user of the service do not have to modify its code but simply specify in the corresponding .zcfg file that it outputs GIS data and it require that it be accessible as OGC Standard Web Services. By settings, user can ask for automatic data classification of both vector and raster data. The Web Service selection will be made automatically by the ZOO-Kernel depending on the kind of output resulting of the service execution, parameters set in the .zcfg file and those provided by the client when requesting the server. The client can then access the output data by requesting MapServer as a WMS, a WFS or a WCS server. Hence, to publish and classify PRESIANN data using WMS, we created a service which returns the input data as provided by the client and defined a .zcfg file to get the WMS server available. Then we requested the WPS server to execute our service by providing

URL for the data to be published and adding the asReference attribute to the ResponseDocument value requested. The execution for each files results in a set of Mapfiles which are used by MapServer to access locations and properties of the data to be displayed that we can use to access our classified raster data through WMS. Users can send a request to WPS server as following url:

```
http://localhost/cgi-bin/zoo_loader.cgi?
request=Execute&service=WPS&version=1.0.0&Identifier=visualization&
DataInputs=a=Reference@xlink:href=http://localhost/02014.asc&
ResponseDocument=Result@asReference=true@mimetype=image/png
```

The result response from server-side looks similar to following example:

```
<wps:ExecuteResponse….
….
<wps:Reference href="http://localhost/cgi-bin/mapserv1?
map=/var/www/data//Result_2219.map&request=GetMap&service=WMS
&version=1.3.0&layers=Result&width=240.000&height=480.000&format=i
mage/png&bbox=8.625000,102.125000,23.625000,109.625000&crs=EPSG:
4326" mimetype="image/png"/>
</wps:Output>
</wps:ProcessOutputs>
</wps:ExecuteResponse>
</wps:ExecuteResponse>
```

As Result response in <wps: Reference> tag presenting GetMap request is automatically created by using ZOO Services as the detail above. In client side, client needs only to extract GetMap request which is responded from server side and then using OpenLayers.Layer.MapServer library to invoke GetMap request link as a mapserver online resourse. OpenLayers.Layer.MapServer is used to display data from a MapServer. The (Fig. 70) shows system architecture of PERSIANN visualization and (Fig. 71) shows result of the PERSIANN visualization in day 11-01-2002.

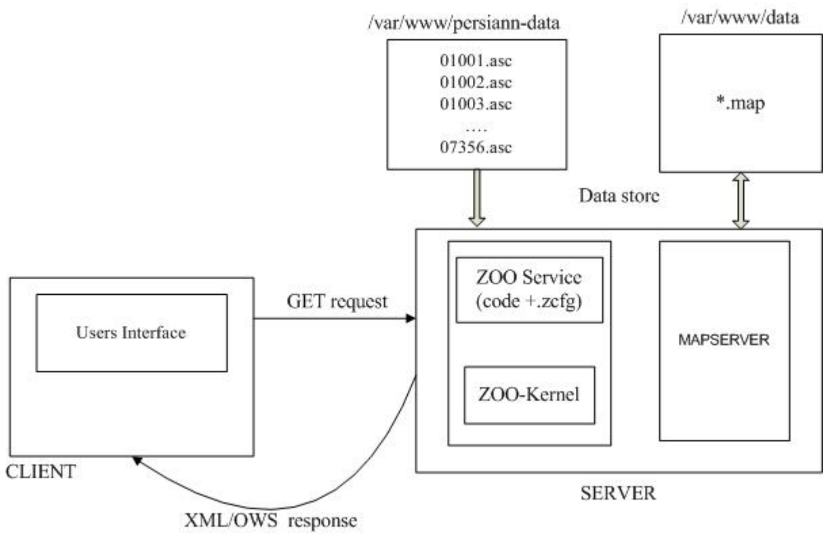


Figure 70: Architecture of visualization PERSIANN service over to the WebGIS

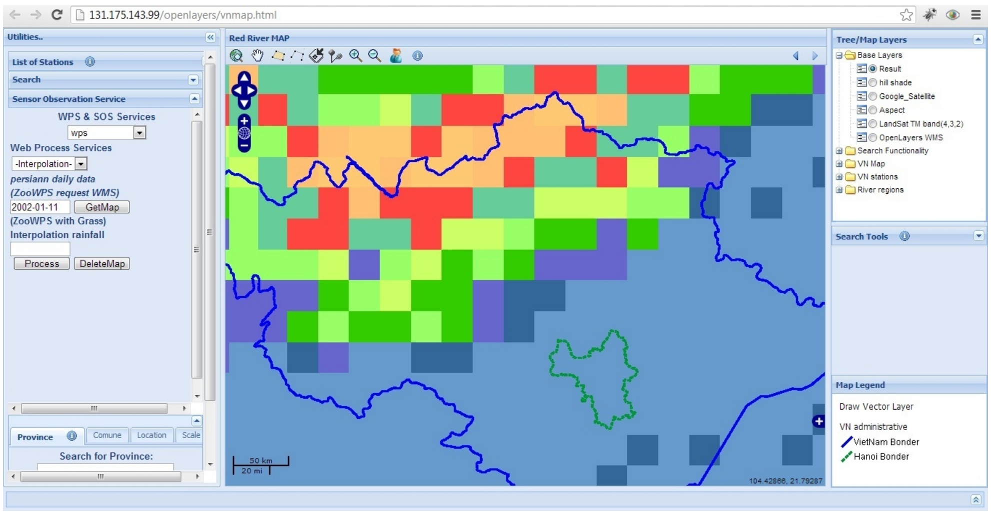


Figure 71: Map of PERSIANN Visualiztion

6.2.1.2 Estimate a correlation and examined a percentage of simultaneous

At this point, in addition to rain gauges data, the data at the locations of the 19 stations can be extracted from the PERSIANN product. Then, they are registered by mean of the istSOS RegisterSensor and inserted by using the

InsertObservation request. This implementation is processed analogously to the details provided in section 6.1.3(a) and 6.1.3(b). In our system two database sources exists: one for rain gauge RG data and the other one for the PERSIANN one. By accessing these two data sets through database system, a WPS service was created to analyze the correlation and percentage of simultaneous rain or no-rain between them. The result of this section show in the (Fig. 72) and the Pseudo code as bellows:

BEGIN
- Input GET parameter [name of sensor, date-time to interpolate]
- Connect to PERSIANN database and RG database in order to extract theirs values start from starting-date to ending-date
- Assign array PN[]'s values to PERSIANN data extracting form DBMS
- Assign array RG[]'s values to PERSIANN data extracting form DBMS
- Calculate Correlation between PN[] and RG[]
- Calculate percentage of simultaneous between PN[] and RG[]
- Generate Output

END

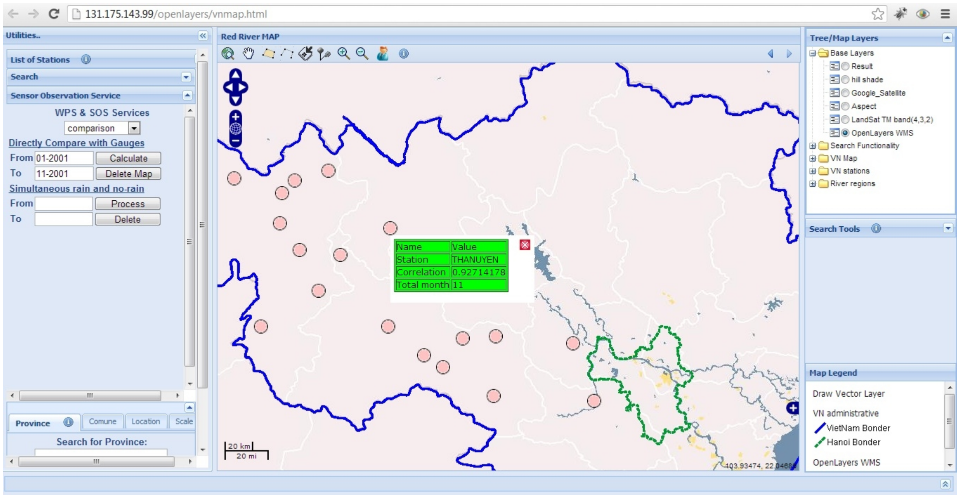


Figure 72: Map of estimate a correlation and examined a percentage of simultaneous between RG and PERSIANN data

6.2.2 Revised Universal Soil Loss Equation computation

The Revised Universal Soil Loss Equation (RUSLE) (Renard et al. 1991) is an empirical equation designed for the computation of average soil loss in total area. Many researches have implemented RUSLE model based on off-line Geographic Information System (GIS) and Remote Sensing (RS) such as (Ashaq, 2011), (Liu Yaolin. 2005). The RUSLE is written as:

$$A = R \times K \times L \times S \times C \times P \quad (1)$$

where A is the estimation of average soil loss in t/ha over a period selected for R usually a yearly basis; R is the rainfall-runoff erosivity factor in MJ mm/(ha h yr). K is the soil erodibility factor [t ha hr/(ha MJ mm)] which is a measure of the susceptibility of soil to erode under a standard condition and is adjusted to accommodate variation in soil moisture content (Millward and Mersey, 1999), it is determined by the cohesive force between the soil particles, and may vary depending on the presence or absence of plant cover; L is the slope length factor; S is slope steepness factor; C is cover and management factor which estimates the soil loss ratio, it may understand as a ratio comparing the soil loss from land under a specific crop and management system to the corresponding loss from continuously fallow and tilled land; and P is the ratio of soil loss with a support practice like contouring, strip-cropping, or terracing to that with straight-row farming up and down the slope (AH. 1978). LS is the combination of the slope steepness (S) and slope length (L) measurements.

LS factor was determined using the equation presented in (Mitasova, 2012):

$$LS = (m+1)\,[A(r) / 22.1]^{m}\,[\sin b(r) / b_0]^{n} \quad (2)$$

Where A(r): is upslope contributing area per unit contour width, b is the slope, m is 0.4-0.6 and n=1-1.4.

The land cover factor was determined by using the Vietnam Atlas in 2004. The C factor includes: water body, rivers or deep zones, forest, dense vegetation, city, etc.; their values vary from 0 to 0.85.

The soil erodibility factor (K) was obtained from Vietnam Atlas in 2004. Soil profile in study area includes: fine sand, very fine sand, loamy, clay, etc; values range from 0.10 to 0.42. The P factor was calculated based on the land use and slope map in percent. Slope classes vary from 0-5% to 50-100% and the corresponding values are from 0.10 to 0.33 in case of land use is agriculture (Wischmeier and Smith, 1978). For the determination of the R factor, the (Renard, 1991) equation was used (equation 3). Average of rainfall annually over 40 years ranges from ~ 1400 to 3000 mm.

$$R = 587.8 - 1279P + 0.004105P^{2} \quad (3)$$

In our study, LS, C, K, P factors were derived from Atlas-Vietnam maps in the years 2000/2004 and ASTER DEM. Finally, according to (2), we build a dedicated Web Service able to query the two database sources (PERSIANN and RG) and invoke GRASS functions in order to use r.mapcalc to convert rainfall annual to R factors by using (3). At last, we adding the new R factor to (1) to calculate soil erosion hazard map (see Figure 66) and example of Web-GIS interface (Fig 67). To report the results of calculation, GRASS's statistical report functionality was used for each area affected by soil erosion problems over the whole area. In client's main menu supports to client in order to select GET request's parameters to measure erosion hazard map and overlay map to Web-GIS (see Figure 68).

All processes were made available as GRASS-based WPS.

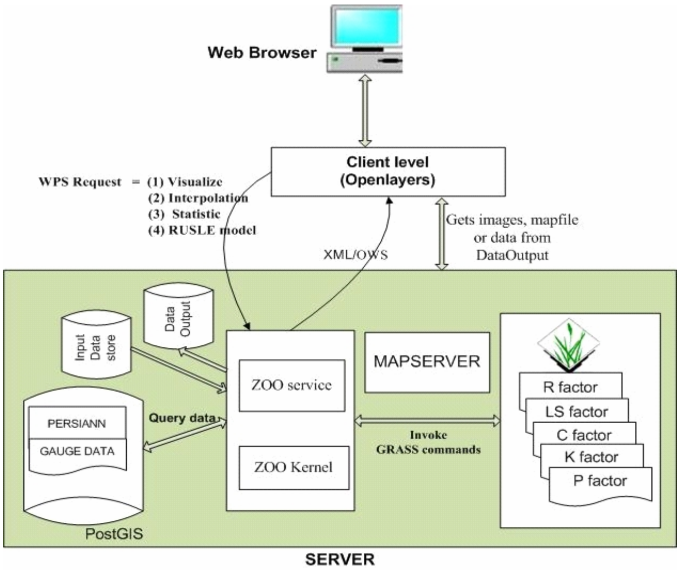


Figure 73: Showing Architecture of main component of DSS

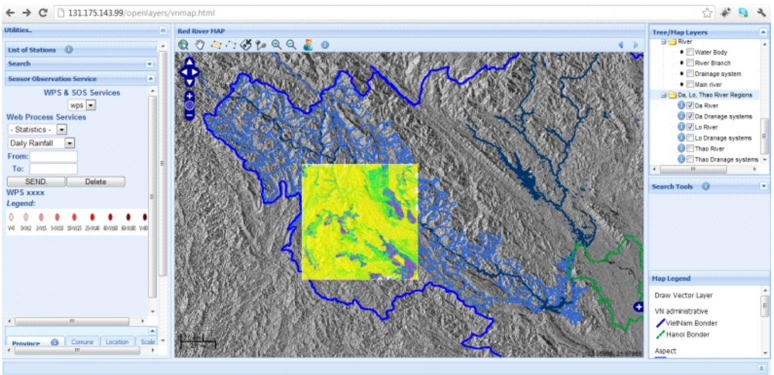


Figure 74: Showing erosion hazard map of sub-area (A4) based on PERSIANN in year 2000

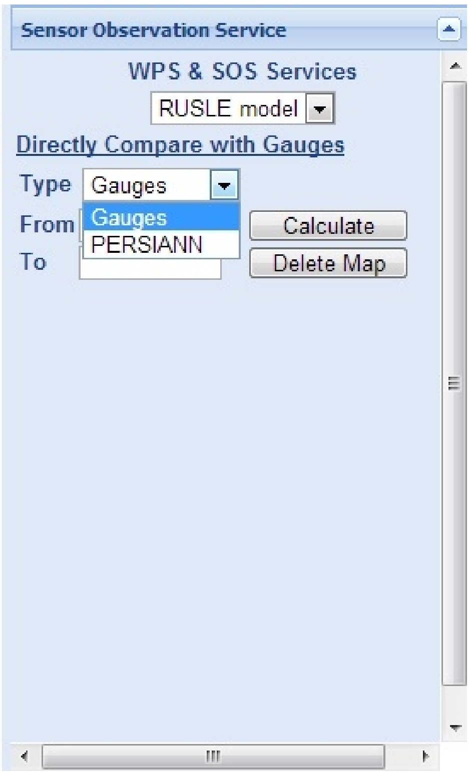


Figure 75: User's main menu for operating RUSLE model and map

6.2.3 Results of comparison

The table 24 presents results of direct comparison between PERSIANN and RG

Station's		2000				2001				2002		
name	Corr	P (%)	Ave	Std	Corr	P (%)	Ave	Std	Corr	P (%)	Ave	Std
Dien Bien	0.56	64.1	1.7	3.3	74.5	75.25	0.96	1.2	0.94	72.5	1.2	3.26
Nam Muc	0.73	72.3	1.8	4.2	0.87	75.6	1.08	3.87	0.79	75.8	3.21	7.36
Than Uyen	0.91	73.6	1.5	4.5	0.90	78.75	1.4	2.26	0.93	69.9	2.54	7.24
Nam Giang	0.92	74.1	4.1	9.6	0.85	76.5	2.98	5.84	0.72	76	4.0	9.2
Tuan Giao	0.67	67.2	1.9	4.6	0.75	76.2	1.56	4.1	0.95	71.4	1.8	3.3
Hoa Binh	0.59	66.1	-0.6	6.3	0.81	64.5	3.67	9.77	0.71	71.4	-1.9	-2.85
Lai Chau	0.81	72	4.1	9.5	0.88	78.4	3.5	7.4	0.93	75	7.24	13.5
Muong Te	0.81	72.8	5.4	10.7	0.98	75.5	-2.1	-6.08	0.92	71.8	4.88	9.6
Sin Ho	0.83	71.2	6.5	10.3	0.87	75.4	4.9	8.5	0.77	73.1	5.2	8.8
Quynh Nhai	0.81	68.5	2.2	7.4	0.70	75.1	2.09	8.23	0.91	72.58	1.31	3.94
Son La	0.81	70.6	0.16	-0.14	0.75	77.25	-0.87	-2.14	0.82	75.4	-0.48	-1.16
Tam Duong	0.85	73.8	4.1	6.5	0.90	76.33	2.38	3.58	0.80	71.7	4.14	6.6
Phu Yen	0.78	69.3	-0.26	1.2	0.93	73.66	-0.51	-0.91	0.69	70.5	-0.71	-1.2
Bac Yen	0.87	69.4	0.9	2.56	0.91	73.8	-0.78	-2.28	0.84	71.16	0.78	1.95
Co Noi	0.75	74.5	0.35	0.95	0.76	77.25	-2.37	-6.9	0.88	76	-0.69	-1.96
Moc Chau	0.70	65.7	-1.04	-2.8	0.76	62.8	-1.09	-3.65	0.69	70.6	-0.01	-1.4
Thuan Chau	0.53	69.2	1.53	4.18	0.73	74.25	-0.44	-1.6	0.83	74.6	0.74	-0.74
Yen Chau	0.86	71.6	-0.67	-1.46	0.87	76.58	-2.23	-6.25	0.94	77.6	-0.003	0.267
Thanh Son	0.82	67.3	-1.49	-0.27	0.87	67.5	-0.49	-0.50	0.39	65	-2.44	-2.37

Table 24: Example of three years data (2000, 2001, 2002) of difference between monthly rain gauge (RG) and PERSIANN data (PN); Corr: Correlation between PN and RG; P: percent of sum of no-rain or rain they are simultaneous between PN and RG divide to number of observations; Ave: average difference between RG and PN; Std: difference of standard deviation between PN and RG.

In the Table 24 showing the difference between PERSIANN and RG based on RUSLE model

Year	2000		2001		2002	
Name	PERSIANN	RG	PERSIANN	RG	PERSIANN	RG
hectares	886240	849013	89332	837263	858301	840011
Area (ha)	1,147,943					
P (%)	0.76	0.74	0.78	0.73	0.75	0.73
Def (ha)	37227		56059		18290	
Def (%)	0.02		0.05		0.02	

Table 25 . RUSLE model for A4 sub-area. P (%) is the percentage of the total effective area and the total land. Def (ha) and Def (%) are difference between PERSIANN and RG in hectare and in percentage.

Chapter 7: CONCLUSIONS

Free source remote sensing images available to download via internet, after image processing they are available to be a based map of WebGIS. Beside, Image pyramid and GeoWebCache are powerful tools for increasing a speed of rendering time and those extend functionality can be managed inside GeoServer.

Historical data of hydro-meteorological data (rainfall, water level, discharge, evaporation, etc) can be managed by SOS service relying on PostgreSQL/PostGIS. The use of GetObservation request allows clients to obtain the desired observations and from these results of request they are possible to create a statistic chart or tables. New sensors can be registered as new sensors in SOS system by using RegisterSensor in both way online or off-line, otherwise InsertObservation can be satisfied used to insert observations corresponding to the sensors already registered into SOS system.

User can request his/her statistical results and charts of time-series in short-term (daily) or long-term (monthly, yearly) interval of various sensors from HDI4NVN system for about 50 year of data.

User can request his/her interpolation map from our system for RG data including 50 years of data; and for PERSIANN visualization containing about 7 years of data. But interpolation results still have several drawbacks such as number of point is less and study area is large, and PERSIANN visualization's result doesn't match perfectly with interpolation map of RG data.

Intelligent Geoportal's aim is not only to improve WebGIS existing but also to extend functionality such as: supporting numerous of base maps, search and query, managing and visualization the data from gauge and sensor for long periods of time, presenting data in tables and charts. Otherwise WPS play important role into building statistical functionality, presenting series of daily rainfall from remote-sensed based for around over 10 years over WebGIS, making processing tool to interpolate rainfall/evaporation for around 50 years of data.

Beside Intelligent Geoportal, our DSS system based on two database sources one from rain gauge data (RG) and the other from PERSIANN data (PN). From two data sources, there are able to create WPS service to analyze correlation between them; the correlations in short time (daily interval) are extremely low. However, the correlation of month accumulation between gauge and PERSIANN data almost indicates strong linear relationship (the correlation coefficients of the 19 stations are larger than 0.7). The best correlation corresponds to year 2002, at Tuan Giao station, and it is about 0.95 (table 24).

Differences in the estimate of the precipitation influence the computation of the derived maps. As an example we can consider what we obtain applying the RUSLE model to the interpolated rain gauge map or to the PERSIANN one. Applying RUSLE in the study area with the two datasets, the difference in percentage varies from 0.02% to 0.04% bringing to a soil loss rate for the area smaller than 100 t.ha-1 year-1, see (table 25).

Beside, Intelligent Geoportal our DSS system can support user a general overview about accuracy of precipitation products obtained from satellite based, especially PERSIANN data. Our DSS system also supports to user/developer a potential of applying SOS, WPS OGC in field of managing, monitoring and eliminating the effecting of natural disaster and climate change.

The backbone of the system has been implemented. Now it is just a matter of adding new data and functionalities for enriching it, orchestrating the processes for obtaining information from the raw data collected by sensors (as the RG) or estimated from models (PERSIANN).

In the future work we would like to select and apply some of hydraulic models to enrich functionality of DSS of HDI4NVN's Web-GIS. and, at the same time, to continue to investigate the PERSIANN dataset.

References

Agriculture Handbooks [AH] (1978) *Predicting rainfall erosion losses: a guide to conservation planning, (USA), No. 537.*

Alesheikh, A., Helali, H. and Behroz, H., *2002. Web GIS: Technologies and Its Applications. Symposium on Geospatial Theory, Processing and Applications, 15.*

Alesheikh, A.A., Helali, H., Behroz, H.A., "Web GIS: Technologies and Its Applications", *International Symposium and Exhibition on 'Geospatial Theory, Processing, and Applications', 95th Annual Geomatics Conference of the Canadian Institute of Geomatics (CIG), Ottawa, Canada, 2002.*

Bernard, L., Kanellopoulos, L., Annoni, A. and P. Smits (2005). The European geoportal one step towards the establishment of a European Spatial Data Infrastructure*, Computers, Environment and Urban Systems, 29(1): 15-31.*

Broering, A., Below, S. and Foerster, T., Declarative Sensor Interface Descriptors for the Sensor Web. *WebMGS 2010: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services. 26.-27. August 2010. Como, Italy.*

Brovelli, M. A., Giorib, G., Mussinc, M., Negretti, M., Web geoservices for monitoring the environmental impact of large structure, *International Society for Photogrammetry and Remote Sensing (ISPRS), 2010.*

Brovelli, M.A., Lecture notes in Internet GIS e UMN Mapserver, last update 2011. *http://geomatica.como.polimi.it/corsi/geog_info_system/*

Cannata, M. and Antonovic, M., ISTSOS: Investigation of The Sensor Observation Service. *WebMGS 2010: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services. 26.-27. August 2010. Como, Italy.*

Choi, J.Y., Bernard, A., Engel and Richard L. Farnsworth., *Web-based GIS and spatial decision support system for watershed management. QIWA Publishing 2005 Journal of Hydroinformatics (2005)*

CHRS, Center for Hydrometeorology & Remote Sensing, University of California, Irvine. *http://chrs.web.uci.edu*

Chung, L.K., Fang, Y.M., Chang, Y.H., Chou, T.Y., Lee, B.J., and Baranski, B. (2009). A SOA based debris flow monitoring system - Architecture and proof-of-concept implementation. *Presented at The first symposium on Enterprise Architecture and Information Technology, Taipai, Taiwan*

Craglia, M., be Bie K., Jackson, D., Pesaresi, M., G-Remetey, Wang C., A Annoni, Bian, L., Campbell, F., Ehlers, M., Genderen, J.V., Goodchildi, M., Guo, H., Lewis, A., Simpson, R., Skidmore, A., and Woodgate, P., 2012, Digital Earth 2020: towards the vision for the next decade*, International Journal of Digital Earth, Volume 5, Issue 1, 2012, 4-21.*

Crompvoets, J., & Bregt, A. (2007). National spatial data clearinghouses, 2000–2005. In H. Onsrud (Ed.), *Research and theory in advancing spatial data infrastructure* (pp. 133–146). Redlands: ESRI Press.

Demirkesen. A.C., Evrendilek, F., Berberoglu, S.., Coastal Flood Risk Analysis Using Landsat-7 ETM+ Imagery and SRTM DEM: A Case Study of Izmir, Turkey. *Published online 14 December 2006, Environ Monit Assess 131:293-300 Springer Science.*

ERSDAC, 2010 Earth Remote Sensing Data Analysis Center. *www.ersdac.or.jp*

ERSDAC, 2011 ASTER Global Digital Elevation Model (GDEM). *www.ersdac.or.jp/GDEM/E/2.html*

Geoserver, Available at: *http://geoserver.org/display/GEOS/Welcome, (October, 2012)*

Groot, R., & McLaughlin, J. (2000). Introduction. In R. Groot & J. McLaughlin (Eds.), *Geospatial data infrastructure: Concepts, cases, and good practice (pp. 1–12). UK: Cambridge University Press.*

H. Mitasova, et al., *Geographic Modeling Systems Lab Website (accessed Aug, 2012)*

Ho , L. T. K., Umitsu, M., and Yamaguchi, Y., Flood hazard mapping by statellite images and SRTM DEM in the Vu Gia-Thu Bon alluvial plain, central VietNam. *International Archives of the Photogrammetry, Remote sensing and spatail Science, Vol. XXXVIII, Part 8, Kyto Japan 2010.*

Iwaniak, A., Kaczmarek, I., Kubik, T., Lukowicz, J., Paluszyński,W., Kourie, D., Cooper, A., & Coetzee, S. (2011). *An Intelligent Geoportal for Spatial Planning. 25th International Cartographic Conference, Paris, July 4-8, 2011.*

Jin, B., Tao, X,., Li, P., Wang, Y., 2009. WebGIS Based Information and Decision-marking Support System for Earth Disaster Reduction. *FSKD'09 Proceedings of the 6th* *international conference on Fuzzy systems and knowledge discovery - Volume 7, Pages 397-401 IEEE Press Piscataway, NJ, USA.*

Liu Yaolin, Luo Zhijun, A study on estimation of the amount of soil erosion in small watershed based on GIS: A case study in the three gorge area of China, *Proceedings of Geospatial Information, Data Mining, and Applications (2005)*

Millward, A.A. and J.E. Mersey,. 1999 Adapting the RUSLE to model soil erosion potential in a mountainous tropical watershed. *Catena. 38(2): 109-129.*

Mioc. D., B. Nickerson., E. MacGillivray., A. Morton., F. Anton., D. Fraser., P. Tang., G. Liang., (2010) Early Warning and Mapping for Flood Disasters. *WebMGS: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services. 26.-27. August 2010. Como, Italy.*

Maguire, D.J. and Longley, P.A,. The emergence of geoportals and their role in spatial data infrastructures, *Computers, Environment and Urban Systems, 29(1): 3-14 (2005).*

Masser I. 2005. K.U. Leuven.. GIS Worlds: creating spatial data infrastructures. *Redlands: ESRI Press. http://www.ec-gis.org/inspire/reports/stateofplay2005/rpact05v42.pdf*

Masser, I. (2007). Building European Spatial Data Infrastructures, Redlands, *California: ESRI Press.*

Masser I 2009, Changing notions of a spatial data infrastructure. In Book SDI converegence: research, emerging trends and critical assessment, *Publisher: Nederlands Commissie voor Geodesie, 06/2009; ISBN: 978-90-6132-310-5*

Manonmani, R., Mary Suganya, G., Remote Sensing and GIS Application In Change Detection Study In Urban Zone Using Multi Temporal Satellite, *International journal of Geomatics and Geosciences*, Vol 1, No 1, 2010.

McLaughlin, J.D. 1991. Towards national spatial data infrastructure*, Proceedings of the Canadian Conference on Geographic Information Systems, Ottawa, pp. 1-5.*

Nebert, D.D., ed., 2004. Developing Spatial Data Infrastructures: *The SDI Cookbook. Global Spatial Data Infrastructure Association, ver. 2.0, 25th January 2004, http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf, (accessed, Dec 2012).*

NCHMF, The National Center for Hydro-Meteorological Forecasting, *http://www.nchmf.gov.vn/web/en-US/105/92/Default.aspx*

NASA, 2011 Landsat Compositor *http://landsat.gsfc.nasa.gov/education/compositor/*

Oxfam, 2008, Viet Nam Climate Change, Adaptation and Poor People. *Oxfam International, ISBN 978-1-84814-055-4, www.oxfam.org.uk/publications.*

PW PreventionWeb, 2012. International Disaster Database. PreventionWeb. *http://www.preventionweb.net/english/countries/statistics/?cid=190*. (Access date 06 june 2012).

Peng, Z-R ., An assessment framework for the development of Internet GIS, Environment and Planning *B: Planning and Design 1999, volume 25 pages 117-132.*
http://geography.sdsu.edu/People/Pages/jankowski/public_html/web780/Peng_Internet_Assess_1999.pdf

Pandey, S., Harbor , J. & Engel, B. A. 2001 Internet Based Geographic Information Systems and Decision Support Tools. Quick Study Series. *Urban and Regional Informatio n Systems Association, Park Ridge, IL.*

Raghavan, V., Santitamnont, P., Masumoto, S., Honda, K., : Implementing Web GIS Application using Open Source Software, MapASIA 2002, *by Center for Spatial Data Management and Asian Institute of Technology, at Changri-la Hotel, Bangkok Thailand, 11-13 Sept. (2002).*

Rautenbacha, V., Coetzee, S., Strzelecki, M., Iwaniak, A., Result of an evaluation of the orchestration capabilities of the ZOO Project and 52o North freamework for an Intelligent Geoportal, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume I-4, 2012 XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia*.

Renard, G.K., Foster, G.R., Weesies, G.A., Porter, J.P. 1991 RUSLE - Revised universal soil loss equation. *Journal of Soil and Water Conservation, v. 46, p.30-33.*

Sheikh, A. H., Sarvesh Palria and Akhtar Alam., Integration of GIS and Universal Soil Loss Equation (USLE) for Soil Loss Estimation in a Himalayan Watershed. *Recent Research in Science and Technology, Vol 3, No 3. (2011)*

Sadeghi-Niaraki A., Rajabifard A., Kim K., Seo J., 2010, Ontology Based SDI to Facilitate Spatially Enabled Society, *Proceedings of GSDI 12 World Conference. 19 – 22 October 2010. Singapour.*

Decision Support Systems, *ISBN 978-953-7619-64-0, pp. 406, January 2010, INTECH, Croatia.*

Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R. & Carlsson, C., Past, present, and future of decision support technology. *Dec. Support Syst. 33 , 111 – 126 (2002).*

SLD, Style Layer Descriptor Profile of the Web Map Service Implementation Specification. OGC Implementation Specification *(http://www.opengeospatial.org/standards/sld), (2007)*

SensorML *OGC http://www.opengeospatial.org/standards/sensorml (2012)*

Sarawut Ninsawat, Kiyoshi Honda, Development of NOAA and Landsat image server using FOSS. *Proceedings of the FOSS/GRASS Users Conference - Bangkok, Thailand, 12-14 September 2004*

Tait, M.G., Implementing geoportals: applications of distributed GIS*, Computers, Environment and Urban Systems, 29(1): 33-47, (2005).*

Toby Kurk., David G.Adams., Simon D. Connell., Neil H. Thomson., Three-channel false color AFM for improved of complex surfaces: A study of filamentous cyanobacteria. Ultramicroscopy. *Volume 110, Issue 6, Pages 573-740, 11th International Scanning Probe Miscroscopy Conference, (May 2010.*

Tsou, M-H, Recent developments in Internet GIS
http://www.gisdevelopment.net/technology/gis/techgis_002pf.htm (Accessed Oct 2012)

Ursa, K., Tatjana, V., Kristof, O., Change detection of urban areas – the Ljubljana Slovenia case study. *IEEE. on Joint Urban Remote Sensing Event*. April 11-13 (2011)

Ulrich. K., Tobias. B., Jeffrey. O., DEM generation from ASTER Data for geomorphometric analysis of cerro Sillajhuay, Chile/Bolivia*, (accessed on 30 May 2011).*

Vandenbroucke D. 2005. Spatial Data Infrastructures in Europe*: State of Play, Spring 2005.*

World Disaster Report 2011; *International Federation of Red Cross and Red Crescent Societies: Geneva, Switzerland, 2011. Available online: http://www.ifrc.org/ (accessed on 30 May 2012).*

Williamson I., A. Rajabifard A. and M.E F. Feeney (Eds.), 2003. Developing spatial data infrastructures: from concept to reality. *Boca Raton, FL.: CRC Press*.

Wang, Y., Colby J. D. and Mulcahy K. A., 2002. An efficient method for mapping flood extent in a coastal floodplain using Landsat TM and DEM data*. International Jornal of Remote sensing, 2002 , vol. 23, no 18, 3681-3696, Taylor & Francis Ltd.*

WMS, OpenGIS® Web Map Server Implementation Specification, *http://www.opengeospatial.org/standards/wms (2006)*

Web Processing Service OGC WPS *http://www.opengeospatial.org/standards/wps (2007)*

Walter, K. and Nash, E., Coupling Wireless Sensor Networks and the Sensor Observation Service – Bridging the Interoperability Gap. *12th AGILE International Conference on Geographic Information Science 2009 Leibniz Universität Hannover, Germany*

Wischmeier, W.H. and D.D. Smith., 1978, Predicting rainfall erosion losses, a guide to conservation planning, *Agriculture, Washington D.C., pp: 55-57.*

Zhang. C, Develop a Spatial Decision Support System based on Service-Oriented Architecture