Introduction to Web technology for GIS- Mapserver and Geoserver

Dr. Harish Karnatak
What is Distributed GIS?

Distributed GIS concerns itself with GI Systems that do not have all of the system components in the same physical location. This could be the processing, the database, the rendering or the user interface. Examples of distributed systems are web-based GIS, Mobile GIS, and GRID computing.
What is web service?

- “Web services” is an effort to build a distributed computing platform for the Web;
- Web services allows us to build a highly distributed infrastructure;
- Each web services can be dedicated to a specific task.
What is web service?

- A Service-Oriented Architecture (SOA) is a collection of services or software agents that communicate freely with each other.

- Web Services protocols and standards are the technology that promote the sharing and distribution of information and business data.

- A protocol is a standard method for transmitting data through a network. There are many different specialized protocols to accommodate the many kinds of data that might be transmitted.
Technically...

- **Web services**-
  - Identified by a **URL**
  - Interfaces defined using **XML**
  - Can be **discovered** by other systems
  - Interact using **XML** based messages conveyed by **Internet protocols**.
XML Messaging: SOAP

SOAP (Simple Object Access Protocol) is an XML-based messaging protocol. It defines a set of rules for structuring messages that can be used for simple one-way messaging but is particularly useful for performing RPC-style (Remote Procedure Call) request-response dialogues.
Web Services and GIS
Web Services and GIS

- The web service concept defines the relationships between the three major actions in geo-spatial data access:
  - Service providers who publish services;
  - Service requestor who search and use services;
  - The service registry that matches the request with the existing services.
GIS Web Services Provides Framework for Distributed System

Supporting Many Geospatial Communities

Over Time
- Expanded GIS Services
- More Synergy
- Easier Exploration Tools
- Pervasive Use

GeoWeb

Enterprise Integration
Consumer Mapping
Focused Applications
GIS Networks
Location Based Services
Sensor Networks
Situational Awareness

Data at different location can be accessed and visualized for better planning and decision making. This framework provides better utilization and value addition on data and information.
Services Oriented Architecture (SOA)

Provides a Framework for Integrating GIS and Enterprise Systems
Interoperability is Important in Web services

Focus Is on simple and practical approaches that work
Web Mapping Services

A web mapping service is a means of displaying and interacting with maps on the Web. The first web mapping service was the Xerox PARC Map Viewer built in 1993.

There have been 3 generations of web map service:

- The first generation was from 1993 onwards- consisted of simple image maps which had a single click function.

- The second generation was from 1996 onwards- and still used image maps the one click function. However, they also had zoom and pan capabilities (although slow) and could be customized through the use of the URL API.

- The third generation was from 1998 onwards- utilise AJAX technology which enables seamless panning and zooming. They are customisable using the URL API and can have extended functionality programmed in using the DOM (Document Object Model, a way to refer to XML or HTML elements as objects).
Standards

Dimensions of interoperability for earth science data

ISO / OGC
OGC Web Services (OWS) Initiatives

OGC has started a series of web-based interoperability initiatives since 1999.

The purpose of those initiatives is to develop a set of interoperability specifications for considering as official OGC specifications.

- Web map services specification (WMS);
- Web coverage services specification (WCS);
- Web feature services specification (WFS);
- Web registries services specification (WRS).
GML Overview

A standard encoding of spatial information (content) for encoding geographic information.

- GML allows us to leverage the world of XML;
- Graphical Drawing (2D and 3D);
- Querying and Element Selection;
- Meta-data & Relationships (Semantics);
- Transformations (spatial and non-spatial).
Geoserver
What is Geoserver?

- Java based, platform independent, server side software
- What does it do:
  - take you data and publish it on a network with a variety of protocols
  - with a strong emphasis on OGC standards (WMS, WFS, WCS)
  - but with extensions for mass market ones too (Google Earth, Bing Map, GeoRSS, GeoJSON)
OGC protocols compliance

GeoServer is:

- WMS 1.1.1 compliant
- WFS 1.0 and 1.1 reference implementation
- WCS 1.0 compliant, and soon to become WCS 1.1 reference implementation as well
GeoServer reads a variety of data formats

- PostGIS
- Oracle Spatial
- ArcSDE
- DB2
- MySQL

- Shapefiles
- GeoTIFF
- GTOPO30
- ECW, MrSID
- JPEG2000

Through standard protocols it produces KML, GML, Shapefile, GeoRSS, PDF, GeoJSON, JPEG, GIF, SVG, PNG and more. In addition, one can edit data via the WFS transactional profile (WFS-T). GeoServer includes an integrated OpenLayers client for previewing data layers.
Styles

- **Styled Layer Descriptor (SLD)**
  - An XML document driving map generation,
  - allows for attribute and scale dependent styling

- **SLD file creation**
  - GeoServer has a very simple style generator
  - Use uDig or gvSig to generate SLD files

- No easy to use editor with full support for SLD styles
GeoServer Features

- Fully compliant to WMS 1.1.1, WFS (1.0 and 1.1, transactions and locking) and WCS (1.0 and 1.1) specifications, as tested by the CITE conformance tests. GeoServer additionally serves as Reference Implementation for WCS 1.1 and WFS 1.0 and 1.1.
- Easy to use web-based configuration tool - no need to touch long, complicated config files.
- Mature support for PostGIS, Shapefile, ArcSDE, DB2 and Oracle. MySQL, MapInfo, and Cascading WFS are also supported formats.
- Native Java support for GeoTIFF, GTOPO30, ArcGrid, WorldImages, ImageMosiacs and Image Pyramids.
GeoServer Features

- Support for MrSID, ECW, JPEG2000, DTED, Erdas Imagine, and NITF through GDAL ImageIO Extension. Any format that GDAL supports can be added with a bit of coding.

- On the fly re-projection, for WMS and WFS, with an embedded EPSG database supporting hundreds of projections by default.

- Excellent [Google Earth Support](#), ability to 'publish' data to Google's geo crawlers, so data from GeoServer can be exposed on Google Maps and Earth searches.

- Integration with [GeoWebCache](#), for accelerated tile mapping (like on Openlayers)

- Raw vector data available as GML (2.1.2 and 3.1.1), GeoJSON (JavaScript Object Notation), and zipped Shapefiles through the WFS.
GeoServer Features

- Integrated **OpenLayers** as a default AJAX viewer and preview engine.
- Standards compliant 'by default' - no need to figure out complex configuration options just to serve basic data.
- Performance, on par with the fastest open source mapping servers, sometime faster rendering than any major proprietary server.
- Support for atomic database transactions through the standard WFS-T protocol, available on all data formats.
- **Versioning WFS** to support wiki-style history, attribution, and rollbacks on geospatial data.
- **Integrated Security** to secure individual services and layers to groups of users.
GeoServer Features

- User Interface translated into French, German, Spanish, Portuguese, Dutch, Japanese, Chinese, Russian and more.
- Support for Component WMS / Feature Portrayal Service
- Streaming data readers: no memory bound limit to the amount of data that can be returned.
- Full SLD support, the open standard to define map styles, including both user defined (POST and GET) and used natively in styling configuration.
- Innovative validation Engine that checks inserted features against a configured set of topological and attributed based rules, to maintain backend integrity.
GeoServer Features

- Java (J2EE) servlet-based, can run in any servlet container. This means it easily support 64-bit architectures and advanced multi-threading capabilities.

- Easy to write new data formats with GeoTools DataStore interfaces and helper classes, making GeoServer the standards based interface to legacy data.

- Modularized Codebase for pluggable services.

- Documentation that details everything in easy to understand language.

- Active email lists for quick support.

- Commercial grade installation, support, customization and improvements available from an eco-system of companies, non-profits, and consulting individuals.
Support

- **User Manual:**

- **Online documentation (wiki)**
  - http://docs.codehaus.org/display/GEOS/Home
  - Mailing lists (subscribe, archives):
    - http://sourceforge.net/mail/?group_id=25086

- **Mailing lists, as forums on Nabble:**

- **IRC channel**
  - #geoserver on irc.freenode.net
Mapserver
(Formerly UMN Map Server)
MapServer Origins

- Developed at UMN Remote Sensing Lab
- Funded by NASA “Mission to Planet Earth”
- Started as Arc/INFO AML generation script
- Built on top of standard OpenSource projects like GD, Flex, Bison, FreeType, Proj.4 and libTIFF
System Characteristics

- OpenSource software
- Implemented as a CGI program
- Written in ANSI C/C++
- Source distribution
  - UNIX: autoconf and automake
  - 95/98/NT: VC++ makefiles
Basic Functionality

- Map creation
- Map component (e.g., legend, scalebar and reference map) automation
- Web application building
- Simple feature query

Basic Demonstration Application
Advanced Functionality ...

- On-the-fly projection (w/Proj.4)
- Feature annotation including scaling, rotation, outlines and drop shadows
- TrueType labels and symbols (w/FreeType)
- Feature classification- string comparision, regex and logical expressions
- Scale dependent display and query
... Advanced Functionality...

- Query by point, area or feature across multiple layers
- Quad-tree spatial indexing for shapefiles
- Support for tiled datasets (raster or vector)
- Label collision removal
- URL based configuration
... Advanced Functionality

- OpenGIS specification support for WMS, Context and WFS (partial)
- Programmatic access to underlying C API-MapScript (available for Perl, Python, Tk/Tcl, PHP and Java)
- 24-bit output support, including GDAL-based output (e.g. GeoTIFF)
Supported File Formats

- **Vector Formats**
  - ESRI Shapefiles and SDE layers
  - Inline vector features
  - Database sources- SDE, Oracle Spatial, PostGIS

- **Raster Formats (8-bit only)**
  - TIFF*/GeoTIFF
  - EPPL7
  - WMS

- **Numerous additional formats via GDAL/OGR**

* indicates files that must be accompanied by an ESRI world file for georeferencing
MapServer Output Formats

- varies based upon software build configuration
- image formats include GIF, PNG (8/24 bit), JPEG, WBMP, and GDAL
- vector formats include GML (via WMS/WFS or templates), PDF, SWF (Flash) and SVG (via templates)
- templates can be any text-based format (default is HTML)
So, how does it work?

- Each application is configured using a text file called a “map” file.
- A user adds to the configuration using an HTML form (e.g. layers, area of interest).
- Program results are run through a series of templates depending on the application (e.g. data browse, feature query).
- And the process begins again…
How MapServer Works...

- MapServer application
  - template files
  - config files
  - spatial data

MapServer CGI

web server

HTTP
... How MapServer Works

MapServer CGI

config files

augment config with user input via request (e.g. what layers and where)

spatial data

template files

create map components and/or execute a query, write components to temporary web space

request (HTTP)

response (HTTP)
Typical Application Layout

- **test** (whole application stored in one place)
  - test.map file
  - **graphics** (images used in application)
  - **symbols** (symbol files and icons)
  - **fonts** (font files and index list)
  - **data** (GIS data)
  - *.html (supporting HTML and templates)

Map file is referred to using its full path relative to the location of the MapServer CGI binary.
MapServer Map Files

- Text based, hierarchical
- Control all aspects of an application
  - legends, scalebars, reference maps
  - layer definitions
  - web template definitions

simple one layer example
Map Object

MAP

NAME application name
STATUS ON|OFF
IMAGECOLOR r g b
UNITS METERS|FEET|INCHES|...
FONTSET filename
SYMBOLSET filename
SIZE x y

...Layers, Scalebar, Legend...

END
Basic Layer Object

```
LAYER

   NAME name
   GROUP name
   DATA filename
   STATUS ON|OFF|DEFAULT
   TYPE ANNOTATION|POINT|LINE|POLYGON|RASTER|QUERY
   MINSCALE n (MAXSCALE)
   CLASSITEM column name
   CLASS ...parameters... END (up to 50)
   LABELITEM column name

END
```
Basic Class Object

CLASS

    NAME full name for legend
    EXPRESSION string
    STYLE ...parameters... END
    LABEL ...parameters... END
    TEXT string

END
Basic Style Object

STYLE

ANTIALIAS TRUE|FALSE
BACKGROUND COLOR r g b
COLOR r g b
OFFSET dx dy
OUTLINE COLOR r g b
SIZE n
SYMBOL n|name

END
Basic Label Object

LABEL

TYPE BITMAP | TRUETYPE
FONT name
COLOR r g b
OUTLINECOLOR r g b
SIZE n | TINY | SMALL | MEDIUM | LARGE | GIANT
POSITION UL | CC | LR | ... | AUTO
ANGLE n | AUTO
BUFFER n
ANTIALIAS TRUE | FALSE

END
LEGEND

STATUS ON|OFF
KEYSIZE x y
KEYSPACING x y
LABEL ...parameters... END
IMAGECOLOR r g b

END
Scalebar Object

SCALEBAR
  STATUS ON|OFF
  STYLE 0|1
  INTERVALS n
  COLOR r g b
  BACKGROUND COLOR r g b
  OUTLINE COLOR r g b
  UNITS KILOMETERS|MILES|METERS|...
  LABEL ...parameters... END
  IMAGE COLOR r g b

END
Reference Map Object

REFERENCE
    STATUS ON|OFF
    SIZE x y
    EXTENT minx miny maxx maxy
    COLOR r g b
    OUTLINECOLOR r g b
    IMAGE filename

END
Thank You
CGI vs Servlets
CGI

- Was designed in the early days of the web
- has evolved into a powerful and a useful model for server-side programming
- The **Common Gateway Interface**, or CGI, is a standard for external gateway programs to interface with information servers such as HTTP servers
CGI

- recent developments include use of Perl on the client-side too
- when a web server receives a CGI request, it needs to start a script, allow it to run, get back the results and pass them back to the web client.
- Dynamic environment
Servlets

- Servlets is the opposite to the applets
- it’s a server-side applet
- run inside the environment of the web server
- the browser can submit a request to the servlet for execution
- developed by Javasoft to support server-side programming
- a parallel to the CGI model
- The servlets execute in the JVM loaded by the Web Server. Its loaded only once, and live until it is explicitly unloaded.
- Results in better performance, higher speed.
- Brings in platform-independence, power of a full-blown language - Java, and compatibility with other popular environments like COM, CORBA and C++.
- The web server basically supports the Java Servlet API. It defines how the web server communicates with the servlet engine.
The Servlet Engine is responsible to actually load and unload the servlets and communicate with the web server. The engine in principal is a set of servlets which together do the work.
Comparison

- Servlets use the Java environment. All the positive features of Java are a strong motivation for a secure, scalable and robust deployment over the Internet/Intranet.
- Servlets has all benefits of a CGI-type language
- Servlets are an easier and friendlier development environment
- Servlets has a increasingly strong developer base and expertise. CGI has a better base, and larger support, as of now.

- In servlets, the developer need not worry about inner workings of the web server. Faster design and development. In contrast, it looses the tremendous control and customization which the CGI model brings along.

- This control is especially important when you are working in multiple different environments with different security and administration needs.
CGI-Perl combo is the most popular. Perl is interpreted. So, it brings in all the positive and negatives of an interpreted language. For example, simplicity, customization - +ve, and slower speed, a newer and a semi-language, not memory-resident.

Servlets have a powerful named - servlet chaining. Do not have a similar philosophy in CGI-Perl.

CGI-Perl has a strong backing of most of industry and educational institutions. Servlets are newer, but gaining acceptance.
- CGI-Perl model is proven, unlike Servlets (although reports are coming in on the awesome speed and efficiency of the servlets).
- CGI-Perl is a big security risk. Servlets can take advantage of the Java Security Model.
- CGI is going to stay. Servlets are however gaining popularity.
- An interface between CGI-Perl and the Servlets is a good direction to move to. This solution is currently being worked on (alpha version is available).