Open Geospatial Consortium (OGC) and Web Services (WMS, WFS)

Responsible persons:

Regula Stopper

(Overall)

Ionut Iosifescu Enescu

(Content)

Samuel Wiesmann

(Revision)

Olaf Schnabel

(Revision)

Table Of Content

1. Open Geospatial Consortium (OGC) and Web Services (WMS, WFS)

Learning Objectives

You will be able...

- ...to explain why OGC developed publicly available specifications.
- ...to explain why it is recommended to use OGC standards.
- ...to understand the content of a GetCapabilities file.
- ...to make a WFS or WMS request on a Web Map Server.

Introduction

Every day, more people use the World Wide Web to get geospatial information.
According to McKee (2001) what is happening in the world of the "Spatial Web" is the following:

- As the Spatial Web grows, there are more online spatial resources available. And while these resources are getting simpler to use, there is increasing potential for extended capability and complexity in Web mapping applications.
- As Web sites become richer in processing resources, users will need to own less software and their sessions on the Web will become more interactive.

Until now, data sharing between institutions that use software from different vendors was difficult, because of the various formats.
That is why Open Geospatial Consortium (OGC) began in 1994 to enlist vendors of GIS software (and other organisations) in a technical committee process to reach consensus on open *interfaces* [1] that would enable their systems to communicate across networks. OGC's highly successful 1999 Web Mapping Testbed prompted the consortium to make testbeds for its main method: the development of specifications. These specifications are software interface specifications that provide a common software syntax and semantics for system-to-system requests in distributed geospatial computing environments. They provide the foundation for making Web mapping as open as the Web itself. (McKee 2001)
Today, luckily, almost all of the geoprocessing software vendors ("old" and new ones) focus on Web mapping where interoperability is of highest priority. Therefore, the chore of downloading, converting and merging spatial data is becoming increasingly unnecessary. (McKee 2001)
This lesson will list a few specifications that OGC developed for Web Mapping issues and go into further details of the specifications "Web Map Service" and "Web Feature Service". Exercises should help to understand how these services work.

[1] An interface, in software terms, is software that enables independent systems to act on each other or communicate with each other.

1.1. Terms

Learning Objectives

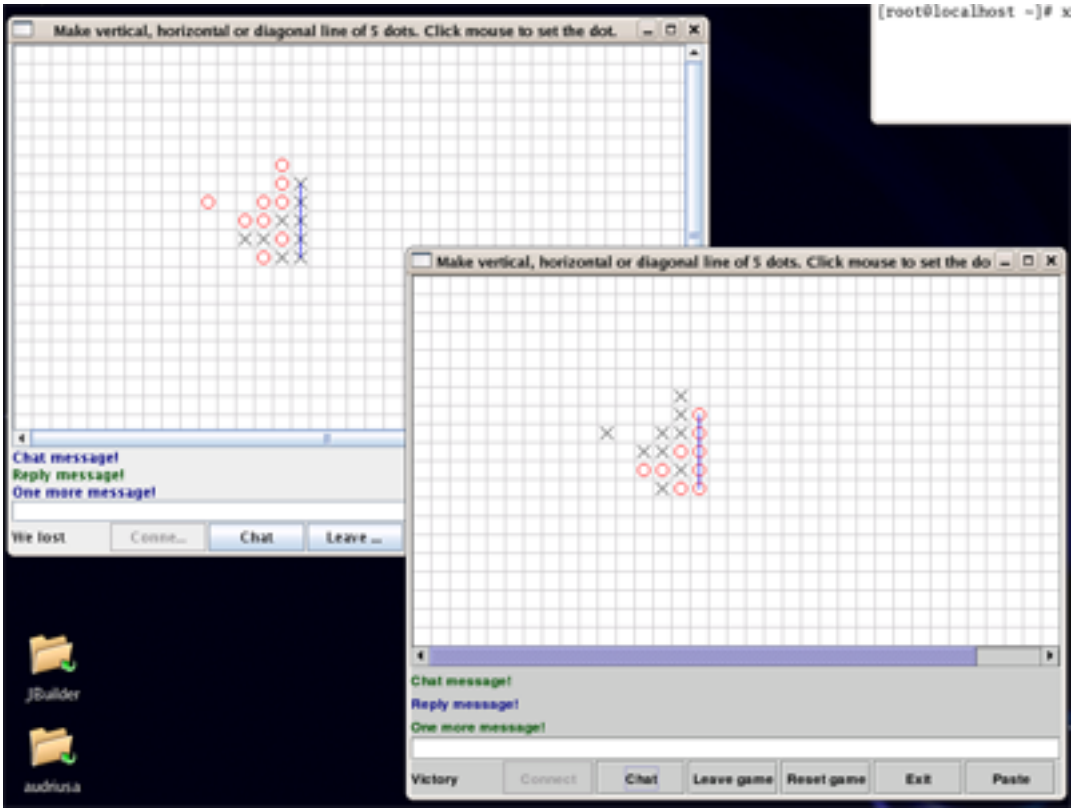
- You will be able to define the terms interoperability, OGC, and Web Mapping.

Introduction

Interoperability, Open Geospatial Consortium (OGC), and Web Mapping are terms that are often used in this lesson. Therefore, we first want to give you definitions of these terms.

1.1.1. Interoperability

Interoperability in general can be defined as follows:
"*Interoperability is the ability of products, systems, or business procedures to work together to accomplish a common task. The term can be defined in a technical way or in a broad way, taking into account social, political and organizational factors.*" (Wikipedia)
In this lesson, we only focus on the technical side of interoperability:
According to (2003) interoperability is defined as follows: "*The capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units.*" (ISO/IEC 2003, p. 9)
Interoperability describes the capability of e.g. different software components to exchange data via a common set of business procedures, and to read and write the same file formats and use the same protocols (Wikipedia). Therefore, interoperability is to overcome import/export obstacles, and distributed resource access barriers imposed by heterogeneous processing environments and heterogeneous data.
The following image demonstrates an example for interoperability: playing the two role network game, when one of the player clients (top left) runs under one system (e.g. Sun Microsystems) and another under an other system (e.g. Macintosh). The applications execute the same bytecode and interoperate using standardised messages for communication.



Interoperability (Wikipedia)

1.1.2. Open Geospatial Consortium (OGC) and OpenGIS

Open Geospatial Consortium (OGC)

The Open Geospatial Consortium (OGC) is an international consortium of more than 330 companies, government agencies and universities participating in a consensus process to develop publicly available specifications that support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT. The specifications empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications. (OGC)

You will learn more about OGC and it's activities in **Unit 1.2. Open Geospatial Consortium (OGC)**

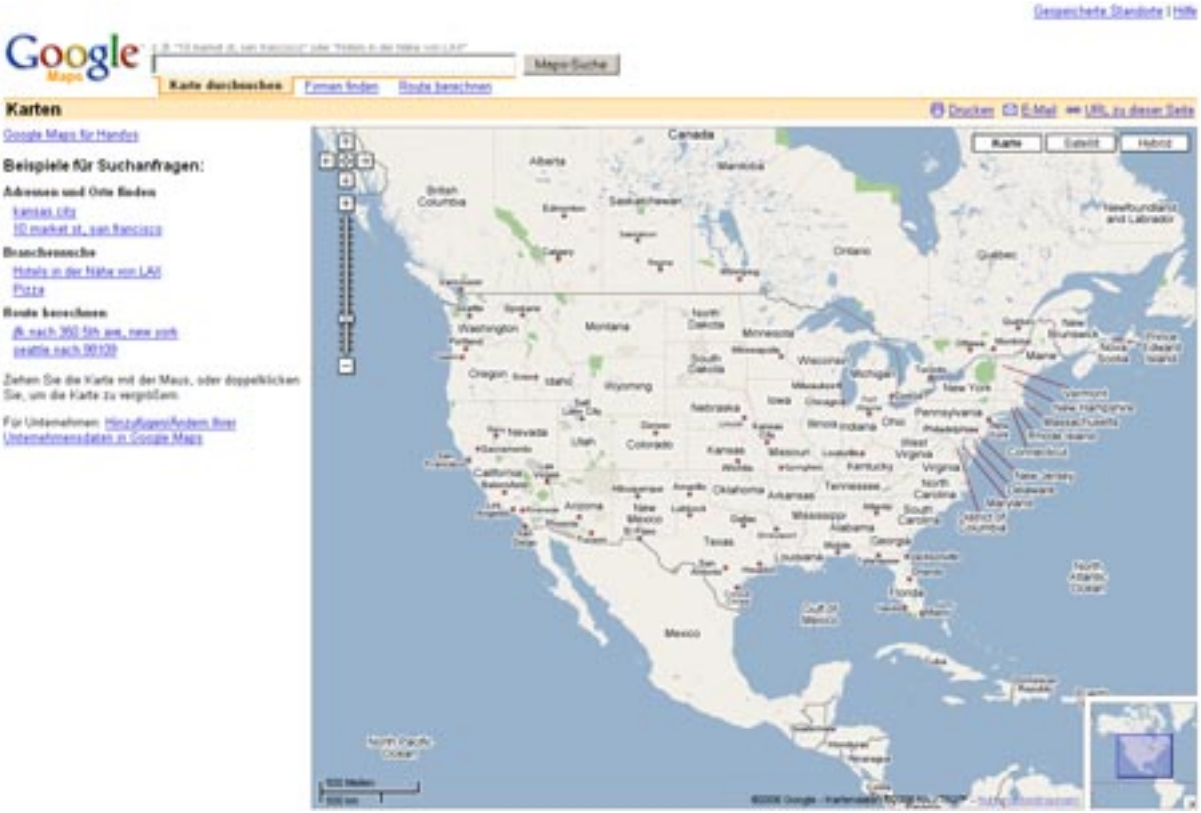
OpenGIS

OpenGIS is a Registered Trademark of the OGC and is the brand name associated with the Specifications and documents produced by the OGC (OGC). OpenGIS specifications are the main "products" of the OGC and have been developed by the membership to address specific interoperability challenges. Ideally, when specifications are implemented by two different software engineers working independently, the resulting components plug and play, that is they work together without further debugging. (OGC)

1.1.3. Web Mapping

Web mapping is viewing geographic information over the World Wide Web, including the presentation of general purpose maps to display locations and geographic backdrops. Web mapping is mainly used for tasks such as planning trips, determining geographical positions, finding landmarks, and obtaining addresses (Whatis.com). But they can be used for much more tasks as well.

Organisations and companies have been providing online mapping services for years. The first well-known web mapping service was **MapQuest** which was launched in 1996. Since then, several other Internet companies, including **Google** and **Yahoo**, have come online with mapping services.



Web Mapping Service Google Maps (GoogleMaps)

Most of these web mapping systems have been implemented as a set of proprietary systems. As a result of this isolated development, online mapping services from different vendors cannot interoperate.
Here comes OGC into play: to address the interoperating problems, OGC developed a non-proprietary Web mapping approach based on open interfaces, encodings, and schemas.
You will learn more about these non-proprietary web mapping approaches within this lesson.

1.2. Open Geospatial Consortium (OGC)

Learning Objectives

You will be able...

- ...to define what OGC stands for and what their main activities are.
- ...to explain the difference between OGC's Abstract Specifications and Implementation Specifications.
- ...to chart the steps of an OGC process.
- ...to explain why you should use GML.

Introduction

As you aready learned in the previous unit, OGC developed publicly available specifications (at no cost) that support interoperable solutions that "geo-enable" the Web. These specifications empower developers to make spatial information and services accessible and useful with all kinds of applications. When they are implemented by two different software engineers working independently, the resulting components work together without further debugging. (OGC)

OpenGIS logo (OGC)

This unit introduces OGC in more details. Their vision, mission, goals, and activities are listed and the specifications that are important for Web Mapping issues are presented as well.
Since the Geography Markup Language (GML) is the basis for the standards "Web Map Service (WMS)" and "Web Feature Service (WFS)" that will be introduced in further chapters, we shortly explain you what GML was defined for and why you should use it.

1.2.1. Vision, Mission & Goals

Who is OGC
The Open Geospatial Consortium (OGC) is an international consortium of more than 330 companies, government agencies and universities participating in a consensus process to develop publicly available specifications that support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT. The specifications empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications. (OGC)

Vision
"*Realisation of the full societal, economic and scientific benefits of integrating electronic location resources into commercial and institutional processes worldwide.*" (OGC)

Mission
"*To serve as a global forum for the collaboration of developers and users of spatial data products and services, and to advance the development of international standards for geospatial interoperability.*" (OGC)

Stratetic Goals
The following stratetic goals are defined by OGC (OGC):

1. Provide **free and openly available standards** to the market, tangible value to Members, and measurable benefits to users.
2. Lead worldwide in the **creation and establishment of standards** that allow geospatial content and services to be seamlessly integrated into business and civic processes, the spatial web and enterprise computing.
3. Facilitate the adoption of open, spatially enabled reference architectures in enterprise environments worldwide.
4. Advance standards in support of the **formation of new and innovative markets and applications** for geospatial technologies.
5. **Accelerate market assimilation** of interoperability research through collaborative consortium processes.

1.2.2. Products

Products

As you could read in the chapters above, the most important activity of OGC is the development of specifications. These specifications can be divided into two groups:

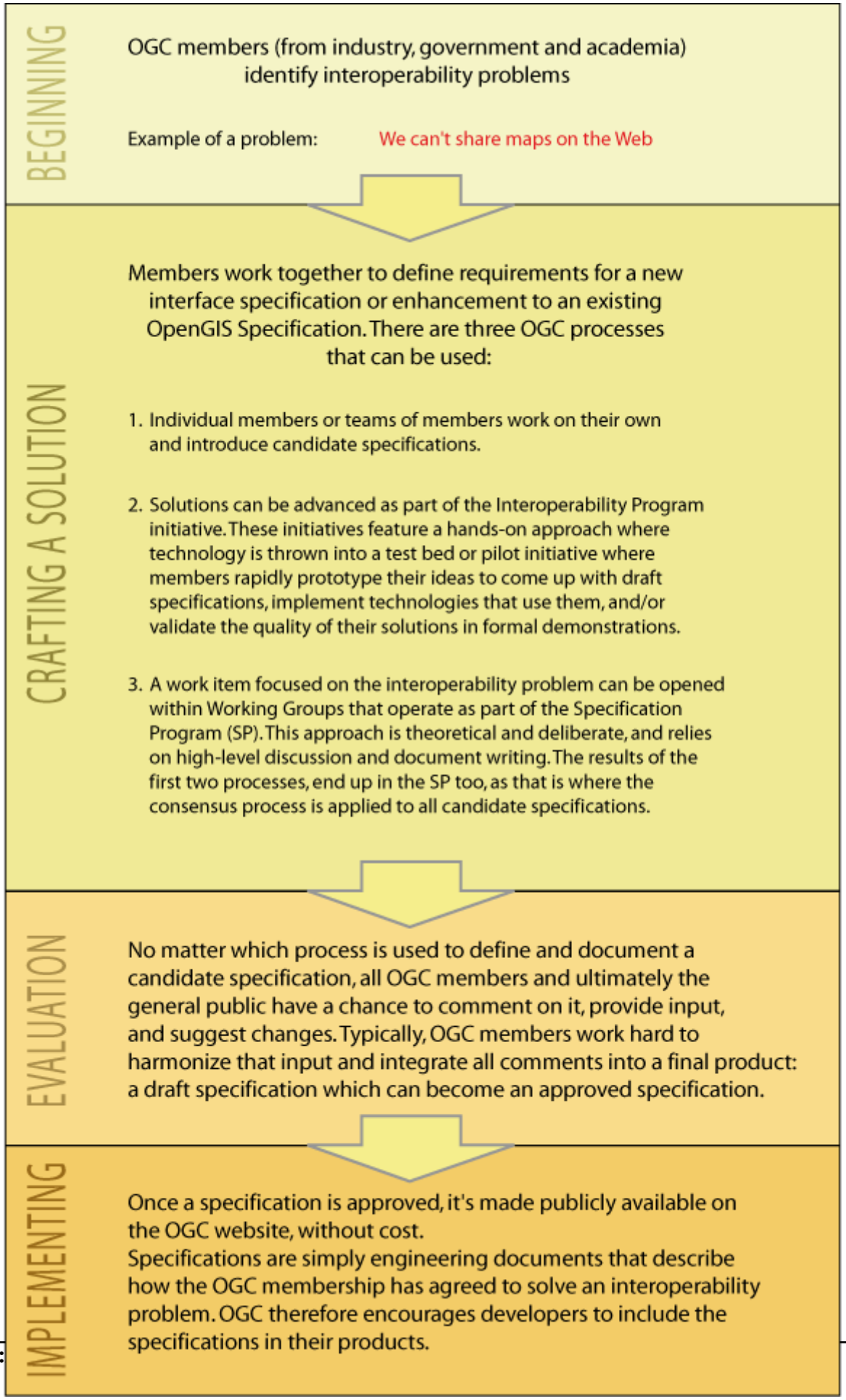
- **Abstract Specifications**: They provide the conceptual foundation for most OGC specification development activities. Open interfaces and protocols are built and referenced against the Abstract Specification, thus enabling interoperability between different brands and different kinds of spatial processing systems. The Abstract Specification provides a reference model for the development of *OpenGIS*[2] Implementation Specifications (OGC). E.g. the Abstract Specification *Features* defines what features are and *Relationship Between Features* defines the relationship between those features. If you are interested in the available Abstract Specifications, have a look at the following link: **OGC Abstract Specifications**.
- **Implementation Specifications**: They are written for a more technical audience and detail the interface structure between software components. An interface specification is considered to be at the implementation level of detail if, when implemented by two different software engineers in ignorance of each other, the resulting components plug and play with each other at that interface (OGC). A few Implementation Specifications will be introduces later in this lesson.

If you are interested in more information about the Open Geospatial Consortium, have a look at their homepage which is rich in information: **OGC Webpage**

1.2.3. OGC Process

OGC has a member approved process that encourages collaboration among and between OGC members to define, document, and implement open specifications that solve geospatial interoperability problems. The OGC exists to enable a fast, effective, inclusive, user-driven process to develop, test, demonstrate, and promote the use of geospatial information and services by using OpenGIS® Specifications (OGC). The following figure shows the different steps of such a process.

[2] OpenGIS is a Registered Trademark of the OGC and is the brand name associated with the specifications and documents produced by the OGC. OpenGIS specifications are the main "products" of the OGC and have been developed by the membership to address specific interoperability challenges. Ideally, when specifications are implemented by two different software engineers working independently, the resulting components plug and play, that is they work together without further debugging.



Process of the development of an OGC specification

For further information about the OGC process and the OGC Programs see

- **OGC Process**
- **OGC Programs**

OGC and ISO

The OGC has a close relationship with *ISO/TC 211* [3]. The OGC abstract specification is being progressively replaced by volumes from the ISO 19100 series under development by this committee. The OGC Standards Web Map Service, Simple Features and GML (are explained later on in this lesson) for example are already ISO standards. (Wikipedia)

1.2.4. Implementation Specifications

We here present you a list of OGC Implementation Specifications that are important for Web Mapping issues (available in the year 2006). You do not have to memorise all the standards and their usage. This list should just serve as a reference book where you have listed a short definition of the standards.

- **Filter Encoding (Filter)**
  Defines an XML encoding for filter expressions. A filter expression constrains property values to create a subset of a group of objects. The goal, typically, is to operate on just those objects by, for example, rendering them in a different colour or saving them to another format. The filter encoding is a common component that can be used by a number of OGC Web Services. Any service that requires the ability to query objects from a Web-accessible repository can make use of the XML filter encoding. (OGC)
- **Geography Markup Language (GML)**
  Is an XML encoding for the transport and storage of geographic information, including both the geometry and properties of geographic features (Reichhardt 2003).
  We go into further details of GML later on in this lesson.
- **Simple Features (SFO, SFS, SFC)**
  The three OpenGIS Simple Features Implementation Specifications define interfaces that enable transparent access to geographic data held in heterogeneous processing systems on distributed computing platforms. The Simple Feature Specification application programming interfaces (APIs) provide for publishing, storage, access, and simple operations on Simple Features (point, line, polygon, multi-point, etc). The purpose of these specifications is to describe interfaces to allow GIS software engineers to develop applications that expose functionality required to access and manipulate geospatial information comprising features with 'simple' geometry using different technologies. (OGC)
- **Styled Layer Descriptor (SLD)**
  Is an encoding that extends the Web Map Service specification to allow user-defined symbolization of feature data. It allows users (or other systems) to determine which features or layers are rendered with which colours or symbols. (OGC)
  We go into further details of SLD later on in this lesson.
- **Web Coverage Service (WCS)**
  Extends the Web Map Server (WMS) interface to allow access to geospatial "coverages" (raster data sets) that represent values or properties of geographic locations, rather than WMS generated maps (pictures). A WCS provides access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering, multi-valued coverages, and input into scientific models and other clients. (OGC)
- **Web Feature Service (WFS)**

[3] ISO/TC 211 is a standard technical committee formed within ths International Organisation for Standardisation (ISO), tasked with covering the areas of digital geographic information and geomantics. It is responsible for preparation of a series of International Standards and Technical Specifications numbered in the range starting at 19101.

Allows a client to retrieve and update geospatial data encoded in GML from multiple Web Feature Services. The specification defines interfaces for data access and manipulation operations on geographic features, using HTTP as the distributed computing platform. Via these interfaces, a Web user or service can combine, use and manage geodata - the feature information behind a map image - from different sources. (OGC)
We go into further details of WFS later on in this lesson.

- **Web Map Context Documents (WMC)**
  Is a companion to the OpenGIS Web Map Service. It describes how to save a map view comprised of many different layers from different Web Map Servers. A 'context' can be encoded and saved so that Web maps created by users can be automatically reconstructed and augmented by the authoring user or other users in the future. (OGC)
- **Web Map Service (WMS)**
  Provides three operations (GetCapabilities, GetMap, and GetFeatureInfo) in support of the creation and display of registered and superimposed map-like views of information that come simultaneously from multiple remote and heterogeneous sources. (OGC)
  We go into further details of WMS later on in this lesson.

People who would like to know more about these and other OGC specifications should visit the **OGC website** where you find much more information.

1.2.5. Geography Markup Language

Geography Markup Language (GML) is an XML encoding for the transport and storage of geographic information, including both the geometry and properties of geographic features. It is an open data exchange standard well-suited for transmitting small to medium-sized volumes of information. GML is usable with all standard XML tools. (Reichhardt 2003)
GML was defined to (Reichhardt 2003):

- Be sufficiently extensible to support a wide variety of spatial tasks.
- Allow for the efficient encoding of geospatial geometry (points, lines, polygones).
- Provide encodings of spatial information and spatial relationships that are both machine-readable and human-readable
- Separate content from presentation (different maps can be produced from the same data)
- Permit easy integration of spatial and non-spatial XML data.
- Enable interoperability of independently-developed applications.

GML is great! (Geoconnections.org)

"*A GML document is an XML-encoded text file and by itself, does not "do" anything.*" (Geoconnections.org)

Why should I use GML?
According to (Geoconnections.org) there are five main reasons why to use GML:

1. GML is an OGC and ISO standard for encoding geographic objects. As such, many vendors have already created products to produce or consume GML, and are more likely to do so. Since it has been extensively reviewed, GML is more likely to be complete and error-free than other methods of encoding geographic objects.
2. Since it is XML based, it is Web-compatible, and inherits from XML all the tools and methods that work on XML schemas and instances.

3. It is object-based, so everything you need to know about a geographic highway (e.g. a piece of highway) can be encoded in a single object without having to look up other pieces of information.
4. It is modular so that you can encode a single object, or a large set of features corresponding to an entire map sheet with many layers (with various levels of complexity in between).
5. GML is mandated by and embedded in other OGC specifications, namely web feature services and filter encoding.

The following image shows an extract of a GML file:

```
<gml:Polygon>
     <gml:outerBoundaryIs>
             <gml:LinearRing>
                     <gml:coordinates>0,0 100,0 100,100 0,100 0,0</gml:coordinates>
             </gml:LinearRing>
    </gml:outerBoundaryIs>
 </gml:Polygon>
 <gml:Point>
    <gml:coordinates>100,200</gml:coordinates>
 </gml:Point>
 <gml:LineString>
    <gml:coordinates>100,200 150,300</gml:coordinates>
 </gml:LineString>
```

Extract of a GML file

If you are interested in details about GML have a look at the **GML Specification (PDF)**. Be aware that this specification is neither a quick nor a simple read and you need a basic understanding of XML.

1.3. Web Map Service (WMS)

Learning Objectives

You will be able...

- ...to explain how a WMS request works.
- ...to list the three operations that are defined by WMS
- ...to request a map using a free map server.

Introduction

A Web Map Service (WMS) is a standard to provide visualisation of geospatial data over the Internet. WMS allows you to consume information on a map over the Internet or to publish map layers from your GIS or image processing system onto the Web. In this unit, we only look at the consumption of map information over the Internet.
WMS operations can be invoked using a standard web browser by submitting requests in the form of URLs. The content of such URLs depends on which operation is requested. Therefore, when requesting a map, the URL indicates what information is to be shown on the map, what portion of the earth is to be mapped, the desired coordinate reference system, the output image width and height, the image format, etc. (Wikipedia)

Map from a Webmap server (UMN MapServer)

A Web Map Service is usually not invoked directly. More often, it is invoked by a client application that provides the user with interactive controls (Wikipedia). Nevertheless, in this unit we want you to invoke the Web Map Service directly. Only by typing the URL in the browser's address bar and changing its parameters, you comprehend how a Web Map Service works. Since within applications, these actions normally are invisible for the user because they happen in the background, you never get to know how Web Map Service requests function.

1.3.1. What is WMS and how does it work?

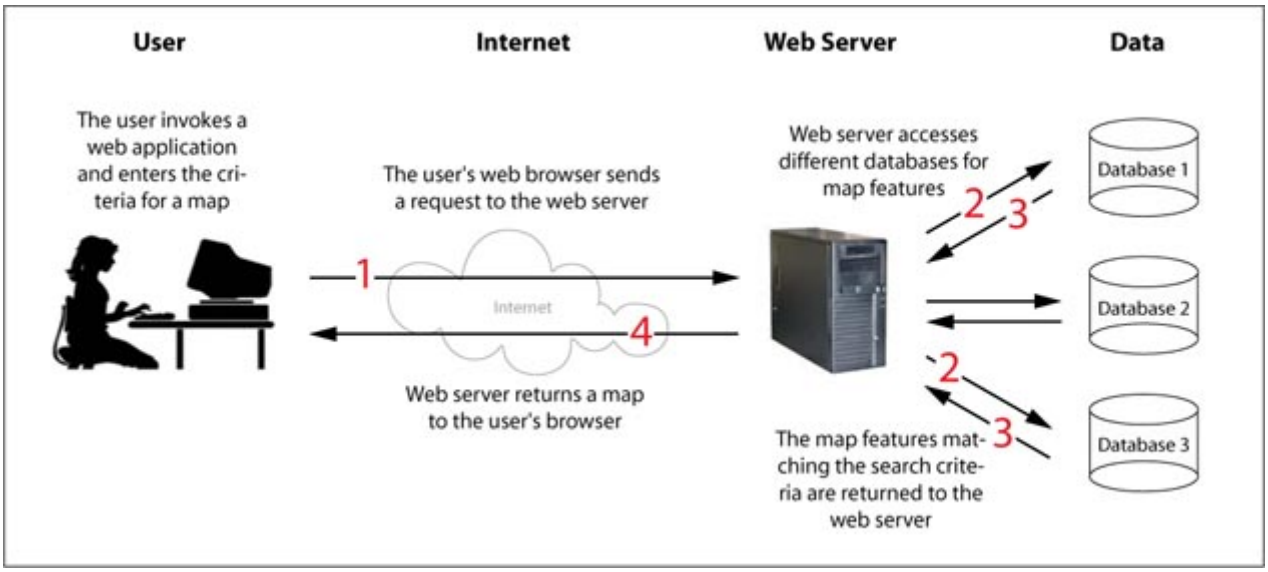
Web Map Service (WMS) provides uniform access by Web clients to maps rendered by map servers on the Internet. Thus, WMS (Reichhardt 2003):

- Enables the dynamic construction of a map as a picture, as a series of graphical elements or as a packaged set of geographic feature data.
- Enables answers to basic queries about the content of the map.
- Can inform other programs about the maps it can produce and which of those can be queries further.

WMS supports the creation and display of registered and superimposed maps that come simultaneousuly from multiple sources both remote and heterogeneous. The WMS returns a raster image ready for portrayal.

How does it work?

When client and server software implements WMS, any client can access maps from any server. Any client can combine maps (overlay them like clear acetate sheets) from one or more servers. Any client can query information from a map provided by any server. While programmers need to write code to implement the specifications, end users can take advantage of products that include them to publish and access geospatial information. Software buyers can choose the best solution for their needs and not worry about if it will work with other solutions; if they all implement the same standard (WMS) they will all work together. (OGC)

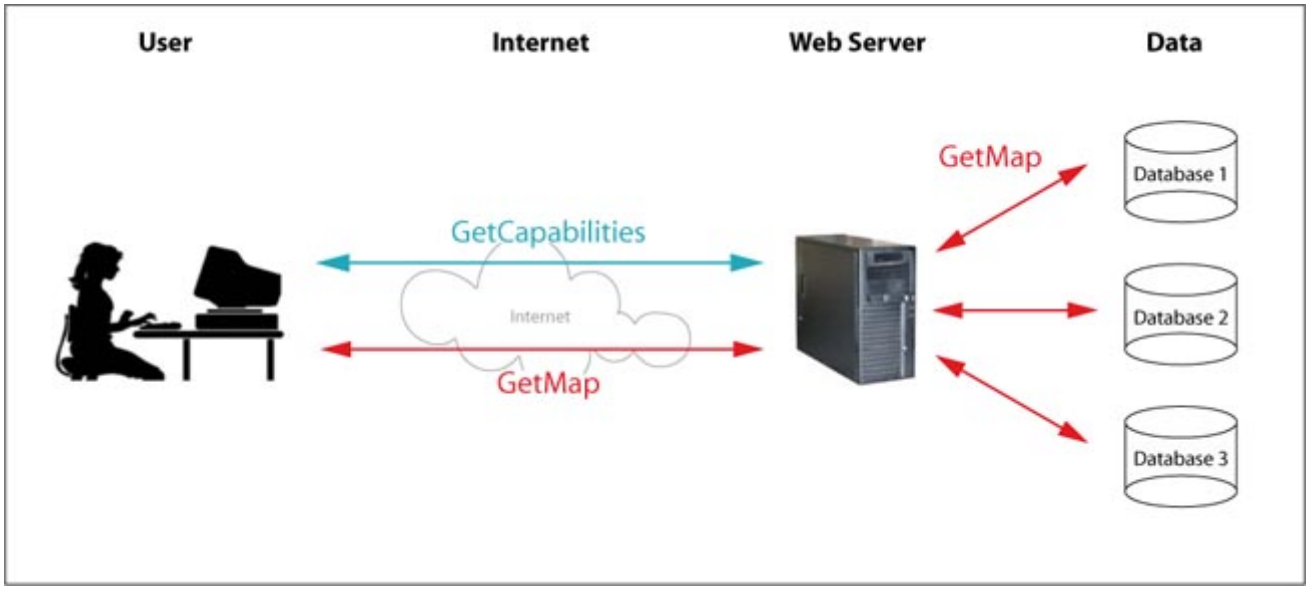


How does a Web Map Service work

In particular WMS defines the following operations:

1. How to get and provide information about what types of maps a server can deliver (**GetCapabilities**)
2. How to request and provide a map as a picture or set of features (**GetMap**)
3. How to get and provide information about the content of a map such as the value of a feature at a location (**GetFeatureInfo**)

The GetCapabilities file lies on the server and therefore a GetCapabilities request ends on the web server (the web server sends the file back to the user). On the other hand, a GetMap request asks for data that is stored in databases and therefore the web server has to contact the databases and extract the requested data. The following figure shows the reach of a GetCapabilities and a GetMap request.



GetMap and GetCapabilities request

1.3.2. GetMap

"*The GetMap operation returns a map. Upon receiving a GeetMap request, a WMS shall either satisfy the request or issue a service exception.*" (OGC 2004)
The following line shows a GetMap request (the URL is all one line, broken up here for readability):

http://www2.dmsolutions.ca/cgi-bin/mswms_gmap?SERVICE=WMS&VERSION=1.1.1
&REQUEST=getmap &layers=prov_bound,popplace&STYLES=&SRS=EPSG:4326
&BBOX=-173.537,35.8775,-11.9603,83.8009 &WIDTH=1024&HEIGHT=768
&FORMAT=image/png
Copy the request parameters into the address bar of your browser and send the request yourself or click the following link to execute the request above **valid request**

The parameters "http://www2.dmsolutions.ca/cgi-bin/mswms_gmap? indicate the location of the map server. The parameters for the GetMap request follow after the server location definition.
A GetMap request consists of the following mandatory request parameters

Request Parameter	Description
VERSION=version	Request version
REQUEST=GetMap	Request name
LAYERS=layer_list	Comma-separated list of one or more map layers.
STYLES=style_list	Comma-separated list of one rendering style per requested layer.
CRS=namespace:identifier	Coordinate reference system.
BBOX=minx,miny,maxx,maxy	Bounding box corners (lower left, upper right) in CRS units.
WIDTH=output_width	Width in pixels of map picture.
HEIGHT=output_height	Height in pixels of map picture.

FORMAT=output_format	Output format of map.

There are six more GetMap parameters that can be declared in the URL, but those parameters are optional. Please have a look at the following pdf file (page 26) if you are interested in those parameters: **OGC Web Map Service Interface**
The result of a GetMap request is a raster image.

Result of a GetMap request (UMN MapServer)

GetMap Exercise
Check out the GetMap request by typing the request string in the address bar of your browser. Watch out that there are no spaces within your URL string. You will get a raster image that you can save on your computer.
To get a new image, you have to change the parameter values of the GetMap request. The parameters *version, request, and coordinate system* are fix for each request. All other parameters are changeable. But since at the moment, you do not know neither the minimum and maximum coordinates nor the names of the available layers or the available image formats, you can only change the width and height of the requested map (you will be able to change all values after the GetCapabilities chapter). Try it out!

You have no idea how to do it?

You have to change the parameters "width=1024&height=768" of the GetMap request URL. Exchange the values 1024 and 671 with other values such as 350 and 100 and have a look at the result.

Take into account that the changing of the width and height parameters only change the width and height of the output image. If their proportion (width / height of output image) is different from the proportion of the original bounding box (width / height of bounding box measured in real coordinates) the content of the output image is distorted. Try it out by taking arbitrary numbers for width and height!

1.3.3. GetCapabilities

"*The purpose of the mandatory GetCapabilities operation is to obtain service metadata, which is a machine-readable (and human-readable) description of the server's information content and acceptable request parameter values.*" (OGC 2004)
The following line shows a GetCapabilities request (use a text editor if you want to view the file "mswms_gmap" (best with Oxygen or UltraEdit).

http://www2.dmsolutions.ca/cgi-bin/mswms_gmap?
SERVICE=WMS&VERSION=1.1.1&REQUEST=getcapabilities

The parameters "http://www2.dmsolutions.ca/cgi-bin/mswms_gmap?" indicate again the location of the map server. The parameters for the GetCapabilities request follow after the server location definition.
A GetCapabilities request consists of the following two mandatory request parameters

Request Parameter	Description
SERVICE=wms	Service type
REQUEST=GetCapabilities	Request name

There are three more GetCapabilities parameters that can be declared in the URL, but those parameters are optional. Please have a look at the following pdf file (page 15) if you are interested in those parameters: **OGC Web Map Service Interface**
The result of a GetCapabilities is an XML file, that contains the mentioned service metadata.

```
<?xml version='1.0' encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://schemas.opengeospatial.net/wms/1.1.1/capabilities_1_1_1.dtd"
 [
 <!ELEMENT VendorSpecificCapabilities EMPTY>
 ]>  <!-- end of DOCTYPE declaration -->

<WMT_MS_Capabilities version="1.1.1">

<!-- MapServer version 4.6.0 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBMP OUTPUT=PDF OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=FREETY

<Service>
  <Name>OGC:WMS</Name>
  <Title>DM Solutions&#39; WMS Demo Server</Title>
  <Abstract>This demonstration server was setup by DM Solutions Group (http://www.dmsolutions.ca/) and is powered by MapServer
  <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://dev1.dmsolutions.ca/cgi-bin/mswms_gmap?"/>
</Service>

<Capability>
  <Request>
    <GetCapabilities>
      <Format>application/vnd.ogc.wms_xml</Format>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://dev1.dmsolutions.ca/cgi-bin/mswms
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://dev1.dmsolutions.ca/cgi-bin/mswm
        </HTTP>
      </DCPType>
    </GetCapabilities>
    <GetMap>
      <Format>image/gif</Format>
      <Format>image/png</Format>
      <Format>image/png; mode=24bit</Format>
      <Format>image/jpeg</Format>
      <Format>image/wbmp</Format>
      <Format>image/tiff</Format>
```

Result of a GetCapabilities request (UMN MapServer)

GetCapabilities Exercise

Check out the GetCapabilities request by typing the request string in the address bar of your browser (or click on the link above). Watch out that there are no spaces within your URL string. Save the resulting file on your computer and open it with an editor. Have a look at the file. You there find information about the GetMap parameters *version, layers, styles, crs (or srs), bbox, and format*. Look for them within the xml-file. Tip: Look for the content of the tags <Layer> and <GetMap>.

You did not find the sections in the GetCapabilities-file?

Information according to the layers:

```
<Layer queryable="1" opaque="0" cascaded="0">
    <Name>drainage</Name>
    <Title>Lakes</Title>
    <SRS>EPSG:42304</SRS>
    <LatLonBoundingBox minx="-169.629" miny="39.2232"
                       maxx="-15.1085" maxy="83.0129" />
    <BoundingBox SRS="EPSG:42304"
                 minx="-2.1694e+06" miny="-386968"
                 maxx="2.79747e+06" maxy="3.74336e+06" />
    <Style>
      <Name>default</Name>
      <Title>default</Title>
      <LegendURL width="18" height="12">
         <Format>image/png</Format>
         <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                         xlink:type="simple"
                         xlink:href="http://dev1.dmsolutions.ca/cgi-bin/
                         mswms_gmap?version=1.1.1&amp;service=WMS&amp;
                         request=GetLegendGraphic&amp;
                         layer=drainage&amp;format=image/png"/>
      </LegendURL>
    </Style>
</Layer>
```

Layer

Information according to the format:

```
<GetMap>
  <Format>image/gif</Format>
  <Format>image/png</Format>
  <Format>image/png; mode=24bit</Format>
  <Format>image/jpeg</Format>
  <Format>image/wbmp</Format>
  <Format>image/tiff</Format>
```

Format

With the information you find in the GetCapabilities-file you are able to change the remaining parameters of which you were not able to change in the GetMap Exercise (bbox, format, layers). Now, create a new request by changing the parameters according to the information of the GetCapabilities-file.

Possible solutions are:

Result 1 with layers cities and boundaries (UMN MapServer)

Result 1 with layers roads, cities, boundaries, and parks (UMN MapServer)

Result 1 with layers boundaries and waterbodies (UMN MapServer)

You have no idea what to do?

The following two blocks show possible GetMap requests with different parameter values.
http://www2.dmsolutions.ca/cgi-bin/mswms_gmap?SERVICE=WMS&VERSION=1.1.1
&REQUEST=getmap&layers=prov_bound,popplace&STYLES=&SRS=EPSG:4326
&BBOX=-173.537,35.8775,-11.9603,83.8009&WIDTH=1024&HEIGHT=768&FORMAT=image/png
http://www2.dmsolutions.ca/cgi-bin/mswms_gmap?SERVICE=WMS&VERSION=1.1.1
&REQUEST=getmap&layers=prov_bound,popplace,drainage,drain_fn&STYLES=&SRS=EPSG:4326
&BBOX=-173.537,35.8775,-11.9603,83.8009&WIDTH=1024&HEIGHT=768&FORMAT=image/jpeg

1.3.4. GetFeatureInfo (optional Attribute)

"*GetFeatureInfo is an optional operation. It is only supported for those layers for which the attribute queryable="1" (true) has been defined or inherited. A client shall not issue a GetFeatureInfo request for other layers. A WMS shall respond with a properly formatted service exception (XML) response (code = OperationNotSupported) if it receives a GetFeatureInfo request but does not support it.*
The GetFeatureInfo operation is designed to provide clients of a WMS with more information about features in the pictures of maps that were returned by previous Map requests. The canonical use case for GetFeatureInfo is that a user sees the response of a Map request and chooses a point (I,J) on that map for which to obtain more information. The basic operation provides the ability for a client to specify which pixel is being asked about, which layer(s) should be investigated, and what format the information should be returned in." (OGC 2004)
If you want to learn more about the GetFeatureInfo operation, have a look at the following pdf file (pages 31-33):**OGC Web Map Service Interface**

1.3.5. Free Mapservers

There are several mapservers that are free to use. We here list you two of them:

UMN Mapserver
"*MapServer is an Open Source development environment for building spatially-enabled internet applications. [...] MapServer excels at rendering spatial data (maps, images, and vector data) for the web.*" (UMN MapServer)
MapServer was originally developed at the University of Minnesota (UMN) in cooperation with NASA and the Minnesota Depatement of Natural Resources.
The UMN WebServer features:

- A multitude of raster and vector data formats

  - JPEG, PNG, GIF, TIFF / GeoTIFF, etc. (**full list**)
  - ESRI shapefile, GML, PostGIS, Interlis, etc. (**full list**)
- OGC Web Specifications
  - WMS, WFS, SLD, GML, WCS, Filter Encoding, WMC, SOS
- On the fly projection

The following GetCapabilities request returns the XML-file where you find all information about the available layers (You already used this URL in the GetCapabilities chapter of this unit). Use a text editor if you want to view the file "mswms_gmap" (best with Oxygen or UltraEdit).
http://www2.dmsolutions.ca/cgi-bin/mswms_gmap?SERVICE=WMS&VERSION=1.1.1&REQUEST=getcapabilities

QGIS mapserver
"*QGIS mapserver is an open source WMS (1.3.0 and 1.1.1) implementation with advanced cartographic features.*" (QGIS mapserver)
It was originally developed at the Instiute of Cartography at ETH Zurich.
The QGIS mapserver features:

- Enhanced WMS (Web Map Service) via HTTP GET. Supports GetCapabilities, GetMap, GetStyle, GetFeatureInfo and custom styling with Styled Layer Descriptor (Supported standards: WMS 1.3.0, WMS 1.1.1, and SLD 1.0.0).
- SOAP via HTTP POST. Compatible with the ORCHESTRA and SANY Service Oriented Architecture.
- Native configuration with SLD. User friendly map symbolisation with QGIS Desktop and PublishtoWeb plugin.
- Cartographic extensions to SLD (diagrams, patterns and custom symbols with Scalable Vector Graphics). Exchange of cartographic rules with the GetStyle operation.

The following GetCapabilities request returns the XML-file where you find all information about the available layers:
http://karlinapp.ethz.ch/fcgi-bin/qgis_map_server/europa_diagram/qgis_map_serv.fcgi?SERVICE=WMS&Version=1.3.0&REQUEST=GetCapabilities

Demis Mapserver
Demis is an OpenGIS Web Map Server that contains world map data. The following GetCapabilities request returns as well the GML-file that contains all information about the available layers.
http://www2.demis.nl/mapserver/request.asp?request=GetCapabilities&version=1.0.0

Other Mapservers
The following link lists servers which can be accessed publicly.
List of OGC WMS Services

Exercise
There are a lot of applications on the Internet that implemented Web Map Services. We here list you two of them:

- **Australian Natural Resources Atlas**: WMS for basic map and land use data
- **Atlas of Canada**: WMS for basic map and thematic data

Search the Internet for three more applications that implemented Web Map Services. Put the links to these applications on the discussion board "Web Map Service". Answer the following questions for each application and put the answers on the discussion board as well:

- Why do you know that the application has WMS implemented?
- Which data is loaded with WMS?

Have a look at at least three applications that your colleagues found. Do you agree with the answer your colleage gave or not? If not, comment on the discussion board.

1.3.6. Styled Layer Descriptor

"*The Styled Layer Descriptor (SLD) is a companion specification to the web map server (WMS) interface specification.*" (Geoconnections.org)

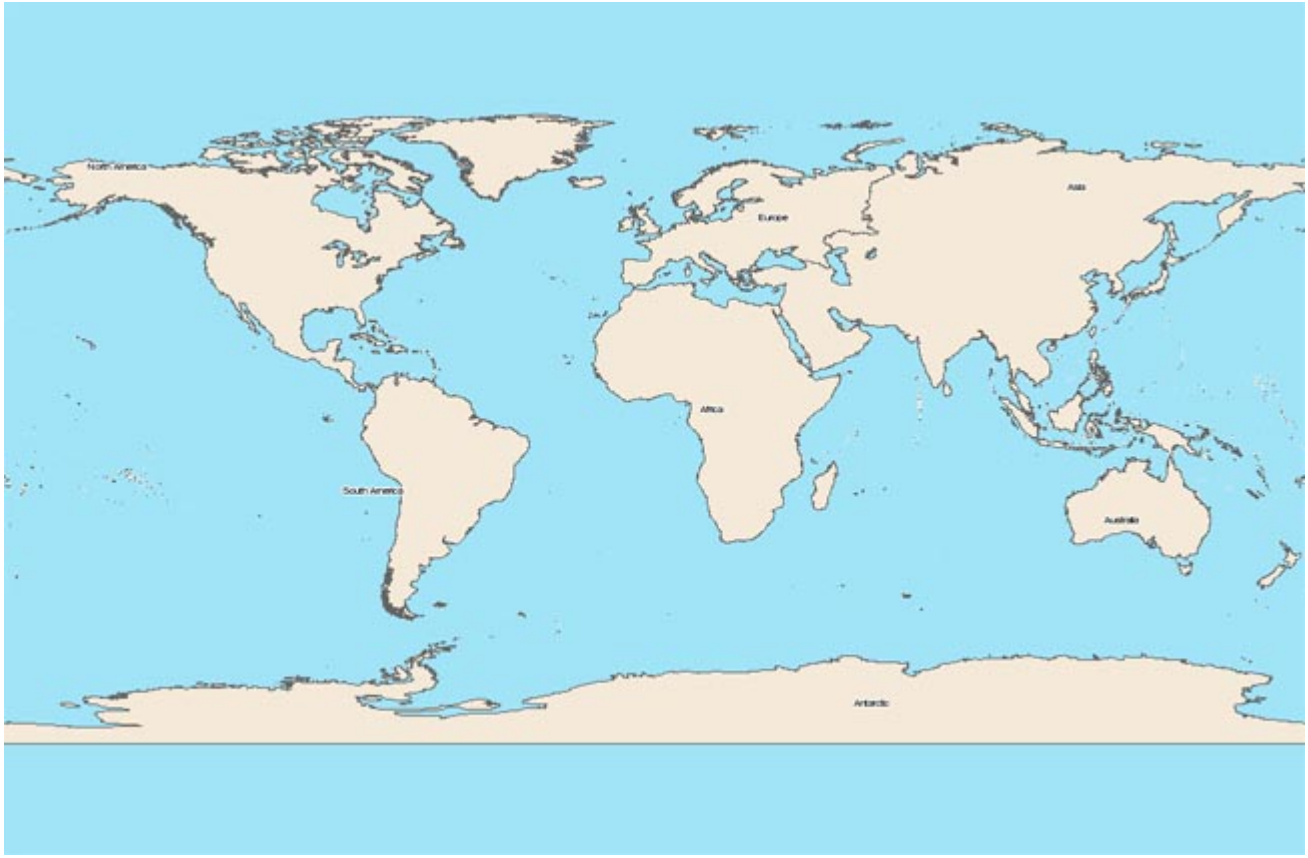
The OpenGIS Styled Layer Descriptor (SLD) Implementation Specification is an encoding that extends the Web Map Service specification to allow user-defined symbolization of feature data. It allows users (or other systems) to determine which features or layers are rendered with which colours or symbols. (OGC)

An example: if a water-bodies layer on a WMS server portrays the water features with the default colour of black, the adding SLD support on the server allows the WMS client to specify that it wants the water features returned as blue. (Geoconnections.org)

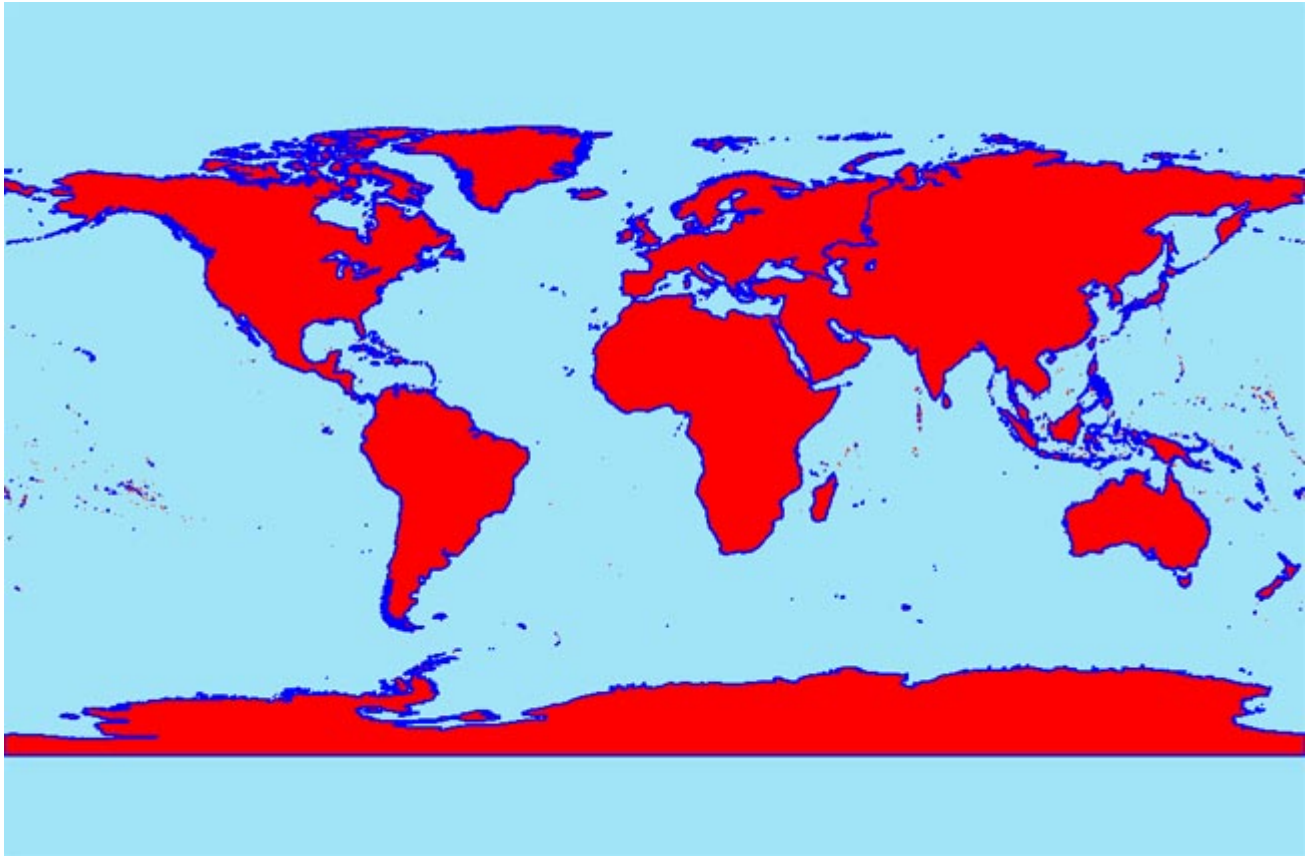
The importance of visual portrayal of geographic data cannot be overemphasised. It is needed for geospatial consumers (either humans or machines) to control the visual portrayal of the data with which they work. The current OpenGIS Web Map Service (WMS) specification supports the ability for an information provider to specify very basic styling options by advertising a preset collection of visual portrayals for each available data set. However, while a WMS currently (in 2006) can provide the client with a choice of style options, the WMS can only tell the client the name of each style. It cannot tell the client what portrayal will look like on the map. More importantly, the client has no way of defining its own styling rules. The ability for a human or machine client to define these rules requires a styling language that the client and server can both understand. This language is called the Styled Layer Descriptor (SLD). (OGC)

The SLD gives the WMS client some control over the visual appearance of the map, as well as the existing control over the combination of layers and viewing geometry. (Geoconnections.org)

Following you see a request result of a web map service without SLD (above) and with SLD (below).



Web Map Service without SLD (UMN MapServer)



Web Map Service with SLD (UMN MapServer)

If you are interested in how to use the Styled Layer Descriptor consult the following literature (please read first the chapters Web Map Service and Web Feature Service):

- **Styled Layer Descriptor (SLD) Implementation Specification**

1.3.7. Unit-Summary

Web Map Service is a standard that provides a standard interface for querying and accessing map layers from a mapping server.
WMS defines the following operations:

1. How to get and provide information about what types of maps a server can deliver (**GetCapabilities**)
2. How to request and provide a map as a picture or set of features (**GetMap**)
3. How to get and provide information about the content of a map such as the value of a feature at a location (**GetFeatureInfo**)

The result of a GetMap request is a raster image as it is shown in the image below:

Result of a GetMap request (UMN MapServer)

The result of a GetCapabilities request is a GML file that contains all map service metadata. The following image shows an extract of a GetCapabilities GML file:

```
<?xml version='1.0' encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM "http://schemas.opengeospatial.net/wms/1.1.1/capabilities_1_1_1.dtd"
 [
 <!ELEMENT VendorSpecificCapabilities EMPTY>
 ]>  <!-- end of DOCTYPE declaration -->

<WMT_MS_Capabilities version="1.1.1">

<!-- MapServer version 4.6.0 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBMP OUTPUT=PDF OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=FREETY

<Service>
  <Name>OGC:WMS</Name>
  <Title>DM Solutions&#39; WMS Demo Server</Title>
  <Abstract>This demonstration server was setup by DM Solutions Group (http://www.dmsolutions.ca/) and is powered by MapServer
  <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://dev1.dmsolutions.ca/cgi-bin/mswms_gmap?"/>
</Service>

<Capability>
  <Request>
    <GetCapabilities>
      <Format>application/vnd.ogc.wms_xml</Format>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://dev1.dmsolutions.ca/cgi-bin/mswms
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:href="http://dev1.dmsolutions.ca/cgi-bin/mswm
        </HTTP>
      </DCPType>
    </GetCapabilities>
    <GetMap>
      <Format>image/gif</Format>
      <Format>image/png</Format>
      <Format>image/png; mode=24bit</Format>
      <Format>image/jpeg</Format>
      <Format>image/wbmp</Format>
      <Format>image/tiff</Format>
```


Result of a GetCapabilities request (UMN MapServer)

The result of a GetFeatureInfo request is an XML file as well.

> Keep in mind that a Web Map Service is usually not invoked directly. More often, it is invoked by a client application that provides the user with interactive controls. Nevertheless, you are able to request a map with a browser as we did in this unit.

1.4. Web Feature Service (WFS)

Learning Objectives

You will be able...

- ...to explain how a WFS request works.
- ...to list the three mandatory operations that are defined by WFS
- ...to request features from a free map server.

Introduction

A Web Feature Service (WFS) is an interface allowing requests for geographical features across the Internet using platform-independent calls in the form of URLs. Geographical features can be thought of as the "source code" behind a map, where as the WMS interface return only an image, which can not be edited or spatially analyzed. (Wikipedia)
The WFS specification defines interfaces for describing data manipulation operations of geographic features. Data manipulation operations include the ability to (Wikipedia):

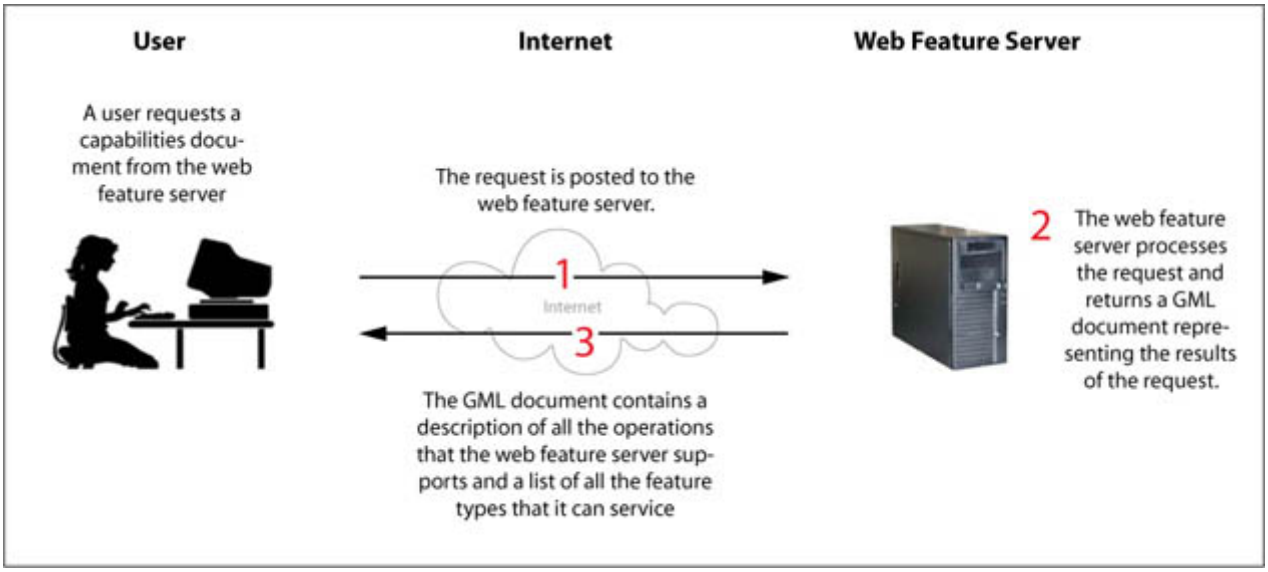
- Get or query features based on spatial and non-spatial constraints
- Create a new feature instance
- Delete a feature instance
- Update a feature instance

Data is passed back and forth between a Web Feature Server and a client with the Geography Markup Language (GML). Since a basic Web Feature Service allows only querying and retrieval of features, we only look at that topic in this unit. We provide additional literature in case you are interested in the other operations.

1.4.1. What is WFS and how does it work?

Web Feature Service (WFS) defines operations to manipulate information about geographic features (points, lines, and polygons). These operations allow to execute transactions (query, create, update or delete) on spatial data through the Web. The geometric descriptions of features in the web feature service specification are encoded in GML. (Geoconnections.org)
An example:
"*Let's say you are building a map with a web map service, and would like to highlight a certain river. Since the web map server delivers only an image of the data, you cannot change the data in any way. You need a web feature service to give you control over the features included in your map.*" (Geoconnections.org)

How does it work?
A WFS request consists of a description of the query and data transformation operations that are to be applied. The request is generated on the client and is posted to a WFS server. The WFS Server reads and executes the request, returning the result in a feature set encoded in GML. A GML-enabled client can then use the feature set. (Reichhardt 2003)
Have a look at the following graphic which illustrates how a web feature request functions.



How does a WFS work

The web feature service specification defines the following operations:

Operation	Description
GetCapabilities (mandatory)	A WFS must be able to describe its capabilities. Specifically, it must indicate which feature types it can service and what operations are supported on each feature type.
DescribeFeatureType (mandatory)	A WFS must be able, upon request, to describe the structure of any feature type it can service.
GetGmlObject (optional)	A WFS may be able to service a request to retrieve element instances by traversing XLinks that refer to their XML IDs. In addition, the client should be able to specify whether nested XLinks embedded in returned element data should also be retrieved.
GetFeature (mandatory)	A WFS must be able to service a request to retrieve feature instances. In addition, the client should be able to specify which feature properties to fetch and should be able to constrain the query spatially and non-spatially.
Transaction (optional)	A WFS may be able to service transaction requests. A transaction request is composed of operations that modify features; that is create, update, and delete operations on geographic features.
LockFeature (optional)	A WFS may be able to process a lock request on one or more instances of a feature type for the duration of a transaction. This ensures that serializable transactions are supported.

1.4.2. GetCapabilities

The following line shows an example of a WFS GetCapabilities request

http://publicatlas.microimages.com/sample_cgi/sample.cgi?service=WFS&request=GetCapabilities&version=1.1.0

The following graphic shows an extract of a possible GML result:

```xml
...
<FeatureTypeList>
 <Operations>
     <Query/>
 </Operations>
 <FeatureType>
     <Name>park</Name>
     <Title>Parks</Title>
     <SRS>EPSG:42304</SRS>
     <LatLongBoundingBox minx="-173.433" miny="41.4271" maxx="-13.3643" maxy="83.7466"/>
 </FeatureType>
 <FeatureType>
     <Name>rail</Name>
     <Title>Railroads</Title>
     <SRS>EPSG:42304</SRS>
     <LatLongBoundingBox minx="-137.447" miny="37.7146" maxx="-46.6233" maxy="66.7201"/>
 </FeatureType>
 <FeatureType>
     <Name>road</Name>
     <Title>Roads</Title>
     <SRS>EPSG:42304</SRS>
     <LatLongBoundingBox minx="-148.059" miny="35.882" maxx="-33.7745" maxy="72.5503"/>
 </FeatureType>
</FeatureTypeList>
...
```


Result of the GetCapabilities request

Check it out! Type the GetCapabilities command in the address bar of your browser and have a look at the resulting file (you can save the file on your computer with right click on the browser window and choose "save page as...").
Count the layers that are listed in the GetCapabilities result. How many are there and what are they called like?

There is only 1 layer provided:

- TR_EarthQuakes (Earth Quakes)

Now that we know the layers that lie on the web feature server, we can request the single features of each layer. But first, you need some basic skills of how to build such a feature request.
There are several request attributes (query elements) - optional and required ones - for each WFS operation (see list in chapter **1.4.1. What WFS is**). The indication of the operation together with these query elements are used to build a WFS request.
In this lesson, we only discuss two query elements of the GetFeature operation. If you are interested in all query elements of each WFS operation, have a look at the following pdf file: **Web Feature Service Implementation Specification**.

1.4.3. GetFeature

GetFeature Request
A GetFeature request is defined by several query elements. The main element is **typeName** (required attribute) which indicates the name of the feature. The names of the features (or layers) are found in the GetCapabilities result file (see content of the tag <Name> (child element of FeatureType) in the image *"Result of the GetCapabilities request"* above or in the GetCapabilities-XML file you saved on your computer).

You are now ready to create a GetFeature request on your own. Tip: Take the GetCapabilities request we already used and change the request type into GetFeature. Add the typeName attribute and choose one layer name you found in the GetCapabilities response document so that it looks like typeName=layername. Don't forget to set '&' between the different query parameters. Type this URL in the address bar of your browser and have a look at the resulting response.

Solution for GetFeature URL

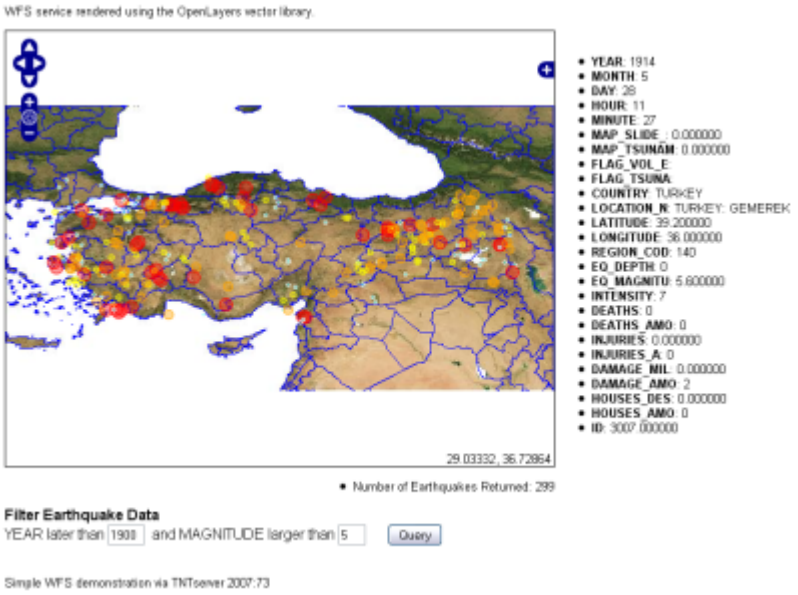
GetFeature URL for the layer "road" (the URL is all one line, broken up here for readability):
http://publicatlas.microimages.com/sample_cgi/sample.cgi?service=WFS&version=1.1.0
&request=GetCapabilities&typename=MII_TREQ%3ATR_EarthQuakes

You certainly noticed that downloading the GML file is quite time-consuming (besides for the layers park and fedlimit). This is because there are so many features within the layers. If you do not need all features, you can limit the number of features returned in the GML by using the MaxFeatures option:
The optional **maxFeatures** (e.g. &MAXFEATURES=100) attribute can be used to limit the number of explicitly requested features that a GetFeature request presents in the response document (OGC 2005).
Try it out by appending the MaxFeatures attribute at the end of your GetFeature URL. Type the new URL in the address bar of your browser and have a look at the resulting response. Compare it to the first result you got by the GetFeature request (especially size and download time).

Solution for GetFeature URL with maxFeature attribute

GetFeature URL for the layer "road" limited on 100 features (the URL is all one line, broken up here for readability):
http://publicatlas.microimages.com/sample_cgi/sample.cgi?service=WFS&version=1.1.0
&request=GetCapabilities&typename=MII_TREQ%3ATR_EarthQuakes&maxfeatures=100

Play with the following application by clicking on the image. With a simple web interface and few buttons it is possible to manipulate the above mentioned parameters of the WFS. In contrary to a WMS (static image), the earthquake attributes



are now accessible via mouseover.

Web Feature Service (Microimages Inc.)

1.4.4. Unit-Summary

Web Feature Service is an interface allowing requests for geographical features across the web using platform-independent calls (the result is a GML-file). Geographical features can be thought of as the "source code" behind the map and can be edited or spatially analyzed. (Wikipedia)
WFS defines the following operations:

- GetCapabilities (*mandatory*)
- DescribeFeatureType (*mandatory*)
- GetGmlObject (*optional*)
- GetFeature (*mandatory*)
- Transaction (*optional*)
- LockFeature (*optional*)

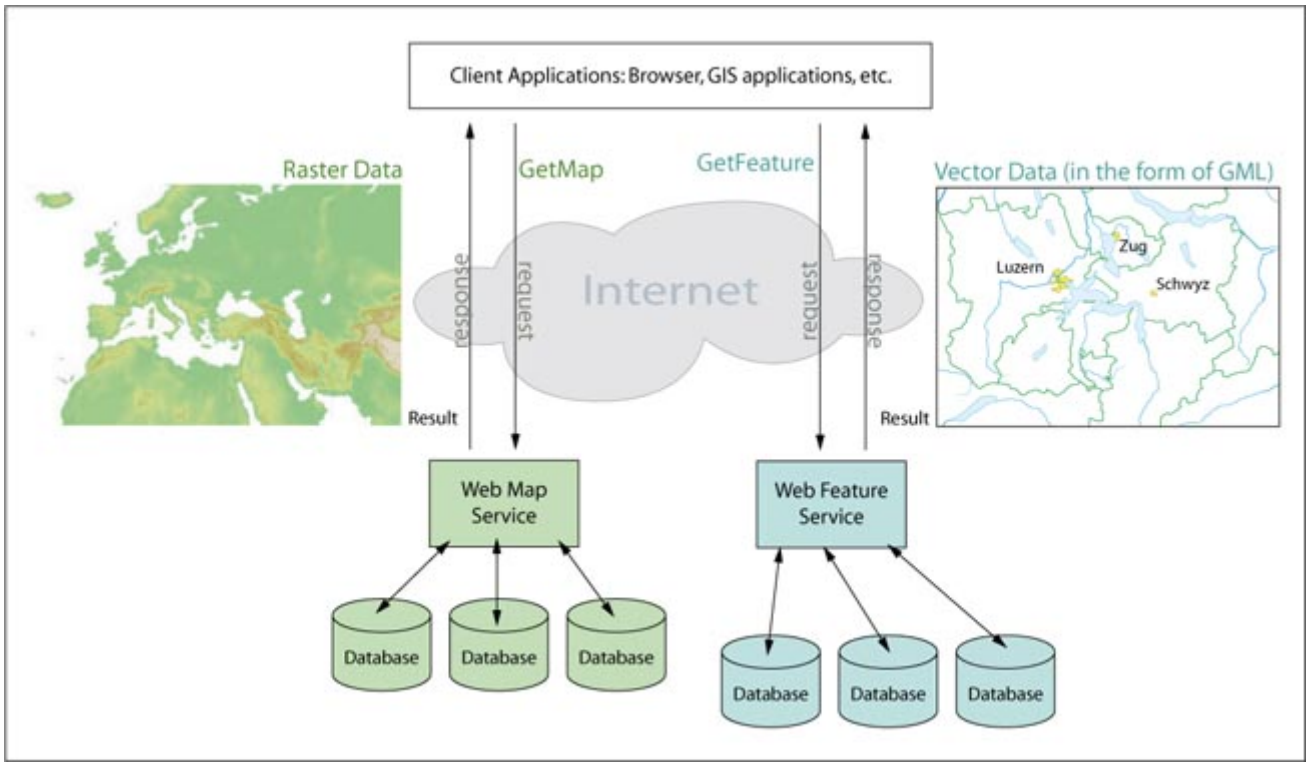
Each operation returns a GML file as result. The GetFeature request is the operation that is mainly used. You can implement the features that you get from this request in an arbitrary application.

1.5. Summary

The Open Geospatial Consortium (OGC) is an international consortium that develops publicly available specifications that support interoperable solutions that "geo-enable" the Web. That means when these specifications are implemented by two different software engineers working independently, the resulting components work together without further debugging. There are a lof of specifications but the most important ones are the Geography Markup Language (GML), Web Map Service (WMS) and Web Feature Service (WFS):

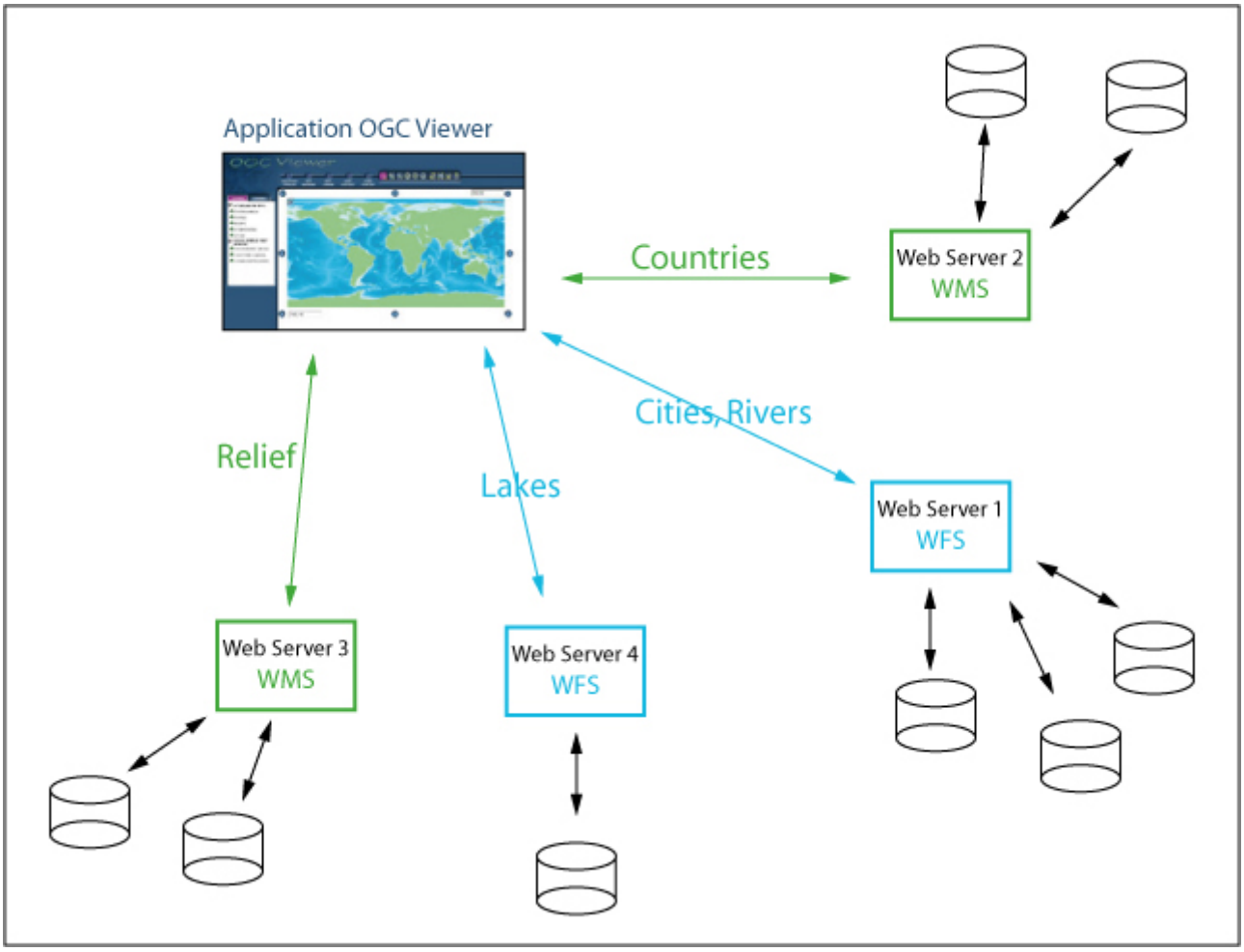
- **GML** is an XML encoding for the transport and storage of geographic information, including both the geometry and properties of geographic features (Reichhardt 2003).
- **WMS** is a standard that provides a standard interface for querying and accessing map layers from a mapping server. GetMap and GetCapabilities are the mandatory operations that are defined by WMS.
- **WFS** is an interface allowing requests for geographical features across the web using platform-independent calls (the result is a GML-file) (Wikipedia). GetCapabilities, DescribeFeatureType, and GetFeature are the mandatory operations that are defined by WFS.

The following graphic shows the difference of the results of a GetMap (WMS) and a GetFeature (WFS) request.



Web Map Service versus Web Feature Service

In this lesson we presented you the services separately. We either talked about WMS or WFS. But normally, these and other services are combined and the data is loaded from an arbitrary number of different web servers. For example: many applications load the raster data from one web server that supports WMS and the features are loaded from another web server that supports WFS.
The following picture demonstrates the combination of different services:



Combination of WMS and WFS

1.6. Self Assessment

The software company Intergraph created an interactive OGC Viewer that is available on the Internet. Intergraph describes it's viewer as follows:

"*Intergraph's OGC (Open GIS Consortium) Viewer is a thin client that enables the user to view one or more OGC WMS (Web Map Service) sites as well as feature data portrayed from one or more OGC WFS (Web Feature Service) sites. The OGC Viewer sends requests, formatted according to OGC specifications, to selected WMS/WFS sites. Data from various sources and formats can be combined and overlaid in one map view. Data from a WMS can be saved as a context document and used again by its creator or another user.*" (Intergraph)

As the description says, you are able to add a new OGC WMS or WFS server to the viewer. The button "Edit Servers" allows you to append new web servers of your choice.

We now want you to add the UMN Mapserver that we introduced you in this lesson. Try it out and explore the application. You may first read the help instructions of the OGC Viewer (to find behind the question mark icon) and read the section "Example Workflow".

You find the OGC Viewer under **www.wmsviewer.com**

Intergraph's OGC Viewer (Intergraph)

1.7. Glossary

EPSG:

"*The "EPSG" namespace prefix refers to the European Petroleum Survey Group geodetic dataset (EPSG), which defines numeric identifiers for many common coordinate reference systems.*

EXAMPLE EPSG:4326 refers to WGS 84 geographic latitude, then longitude. That is, in this CRS the X axis corresponds to latitude, and the Y axis to longitude. " (OGC 2004, p. 10)

Geography Markup Language (GML):

Geography Markup Language (GML) is an XML encoding for the transport and storage of geographic information, including both the geometry and properties of geographic features. It is an open data exchange standard well-suited for transmitting small to medium-sized volumes of information. (Reichhardt 2003)

Interface:

"*An interface, in software terms, is software that enables independent systems to act on each other or communicate with each other.*" (McKee 2001)

Interoperability:

"*Interoperability is the ability of products, systems, or business processes to work together to accomplish a common task. The term can be defined in a technical way or in a broad way, taking into account social, political and organizational factors.*" (Wikipedia)

ISO/TC 211:

ISO/TC 211 is a standard technical committee formed within ths International Organisation for Standardisation (ISO), tasked with covering the areas of digital geographic information and geomantics. It is responsible for preparation of a series of International Standards and Technical Specifications numbered in the range starting at 19101. (Wikipedia)

Metadata:

Metadata is "data about data". "*Metadata is structured, encoded data that describe characteristics of information-bearing entities to aid in the identification, discovery, assessment, and management of the described entities.*" (Durrell 1985)

Open Geospatial Consortium (OGC):

The Open Geospatial Consortium (OGC) is an international consortium of 333 companies, government agencies and universities participating in a consensus process to develop publicly available specifications that support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT. The specifications empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications. (OGC)

OpenGIS:

OpenGIS is a Registered Trademark of the OGC and is the brand name associated with the specifications and documents produced by the OGC. OpenGIS specifications are the main "products" of the OGC and have been developed by the membership to address specific interoperability challenges. Ideally, when specifications are implemented by two different software engineers working independently, the resulting components plug and play, that is they work together without further debugging. (OGC)

Request for Comments:

In internetworking and computer network engineering, Request for Comments (RFC) documents are a series of memoranda encompassing new research, innovations, and methodologies applicable to Internet technologies.

Styled Layer Descriptor:

The OpenGIS Styled Layer Descriptor (SLD) Implementation Specification is an encoding that extends the Web Map Service specification to allow user-defined symbolization of feature data. It allows users (or other systems) to determine which features or layers are rendered with which colours or symbols. (OGC)

Uniform Resource Locator (URL):

A URL is compact string of characters used to identify a resource. It enables interaction with representation of the resource over a network using specific protocols (Wikipedia).

Web Feature Service:

Web Feature Service is an interface allowing requests for geographical features across the Internet using platform-independent calls (the result is a GML-file). Geographical features can be thought of as the "source code" behind the map, whereas the WMS interface returns only an image, which can not be edited or spatially analyzed. (Wikipedia)

Web Mapping:

Web mapping is viewing geographic information over the World Wide Web, including the presentation of general purpose maps to display locations and geographic backdrops. (Whatis.com)

Web Map Service:

Web Mapping Service is a standard that provides a standard interface for querying and accessing map layers from a mapping server. The result of this query is a raster image.

XML:

XML stands for e**X**tensible **M**arkup **L**anguage and is a simple, very flexible text format derived from Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. (WWW-Consortium)

1.8. Bibliography

- **Durrell, W.R.**, 1985. *A Practical Guide to Data Administration*. McGraw-Hill.
- **Geoconnections.org**. *Welcome to the CGDI online training* [online]. Available from: http://www.geoconnections.org/publications/training_manual/e/ [Accessed 24.11.2006].
- **GoogleMaps**. *GoogleMaps* [online]. Available from: http://maps.google.com/ [Accessed 15.11.2006].
- **Intergraph**. *OGC Viewer* [online]. Available from: http://www.wmsviewer.com/main.asp [Accessed 23.02.2007].
- **ISO/IEC**, 2003. *SC36 N0646*. norm.
  Download: http://jtc1sc36.org/doc/36N0646.pdf
- **McKee, L.**, 2001. Geography Connects Cyberspace with the Real World. *GeoWorld*, 2, .
- **Microimages Inc.**. *Sample WFS Application - Earth Quakes* [online]. Available from: http://www.microimages.com/ogc/wfs/earthquakes.htm [Accessed 29.01.2009].
- **OGC**, 2004. *OGC Web Map Service Interface*. OGC 03-109r1.
  Download: http://www.opengeospatial.org/standards/wms
- **OGC**, 2005. *Web Feature Service Implementation Specification*. OGC 04-094.
  Download: http://www.opengeospatial.org/standards/wfs
- **OGC**. Available from: http://www.opengeospatial.org [Accessed 14.11.2006].
- **QGIS mapserver**. *QGIS mapserver* [online]. Available from: http://karlinapp.ethz.ch/qgis_wms/about/index.html [Accessed 29.03.2010].
- **Reichhardt, M.**, 2003. *MapInfo Magazine*, 7, 14.
  Download: http://extranet.mapinfo.com/common/library/mapinfomag_summer2003.pdf
- **Riemer, R., Schneider, T.**. *Untersuchungen zur Interoperabilität von Web Map Services* [online]. Available from: http://www.htw-dresden.de/~fegis/DA/DA_RIEMER_SCHNEIDER_2005/DiplomarbeitRiemerSchneider.htm [Accessed 15.11.2006].
- **UMN MapServer**. *UMN Map Server* [online]. Available from: http://mapserver.gis.umn.edu/ [Accessed 08.12.2006].
- **Whatis.com**. *Online Mapping* [online]. Available from: http://whatis.techtarget.com/definition/0,289893,sid9_gci1179388,00.html [Accessed 15.11.2006].
- **Wikipedia, the free encyclopedia**. *Open Geospatial Consortium* [online]. Available from: http://en.wikipedia.org/wiki/Open_Geospatial_Consortium [Accessed 23.02.2007].
- **Wikipedia, the free encyclopedia**. *ISO/TC 211* [online]. Available from: http://en.wikipedia.org/wiki/ISO/TC_211 [Accessed 23.02.2007].
- **Wikipedia, the free encyclopedia**. Available from: http://en.wikipedia.org [Accessed 15.11.2006].
- **Wikipedia, the free encyclopedia**. *Interoperability* [online]. Available from: http://en.wikipedia.org/wiki/Interoperability [Accessed 14.02.2007].
- **WWW-Consortium**. *Extensible Markup Language (XML)* [online]. Available from: http://www.w3.org/XML/ [Accessed 14.01.2006].