



Introducing OGC Web Services at FMI

EGOWS 2008 Ljubljana
Ilkka Rinne, Finnish Meteorological Institute



Presentation outline

Overview of the OGC Web Services (WMS, WFS, WCS and CS-W)

Meteorology and OGC: GML in observations and forecasts

Technology: Experiences from OGC Web Service implementations used at FMI (UMN MapServer & Geoserver)

Real world uses for the OGC Web Services: Experiences from the FMI pilot projects

What's in the horizon: Plans and expectations in the near future



Open Geospatial Consortium (OGC) Web Service

Web Map Service (WMS)

Superimposed "flat" map-like images.

Images: png, svg, KML,...

Web Feature Service (WFS)

Geospatial objects:
types or instances

GML documents

Web Coverage Service (WCS)

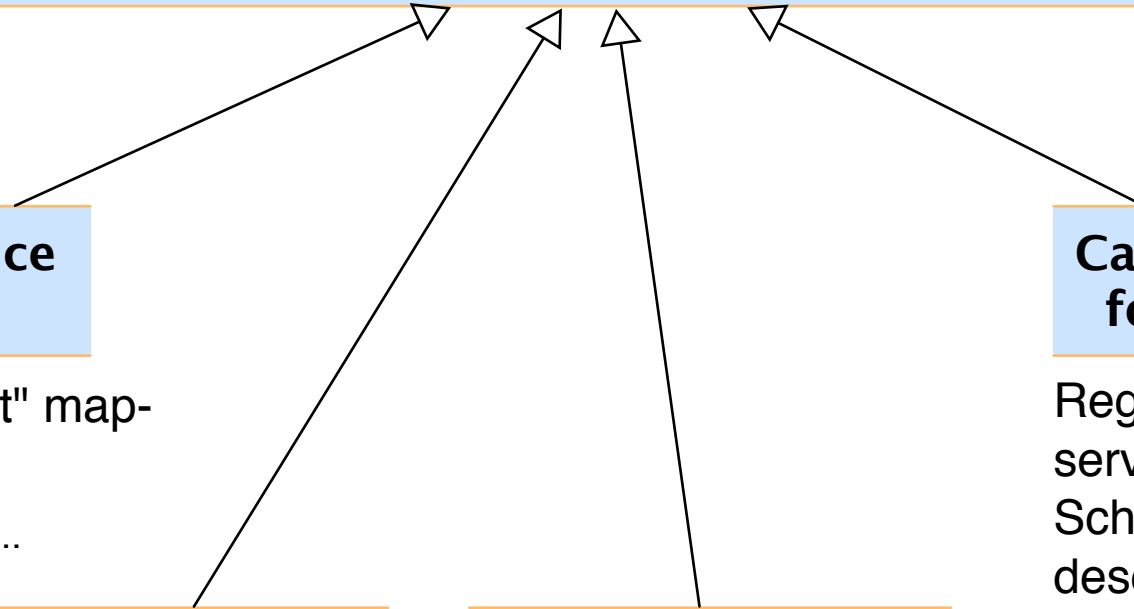
Gridded data with GML
metadata

Binary files: GeoTiff, HDF-EOS,
CF-NetCDF, JPEG2000,...

Catalogue Service for Web (CS-W)

Registries for provided
services, GML Application
Schemas, visualization
descriptions, standardized
terms, ontologies,...

XML documents

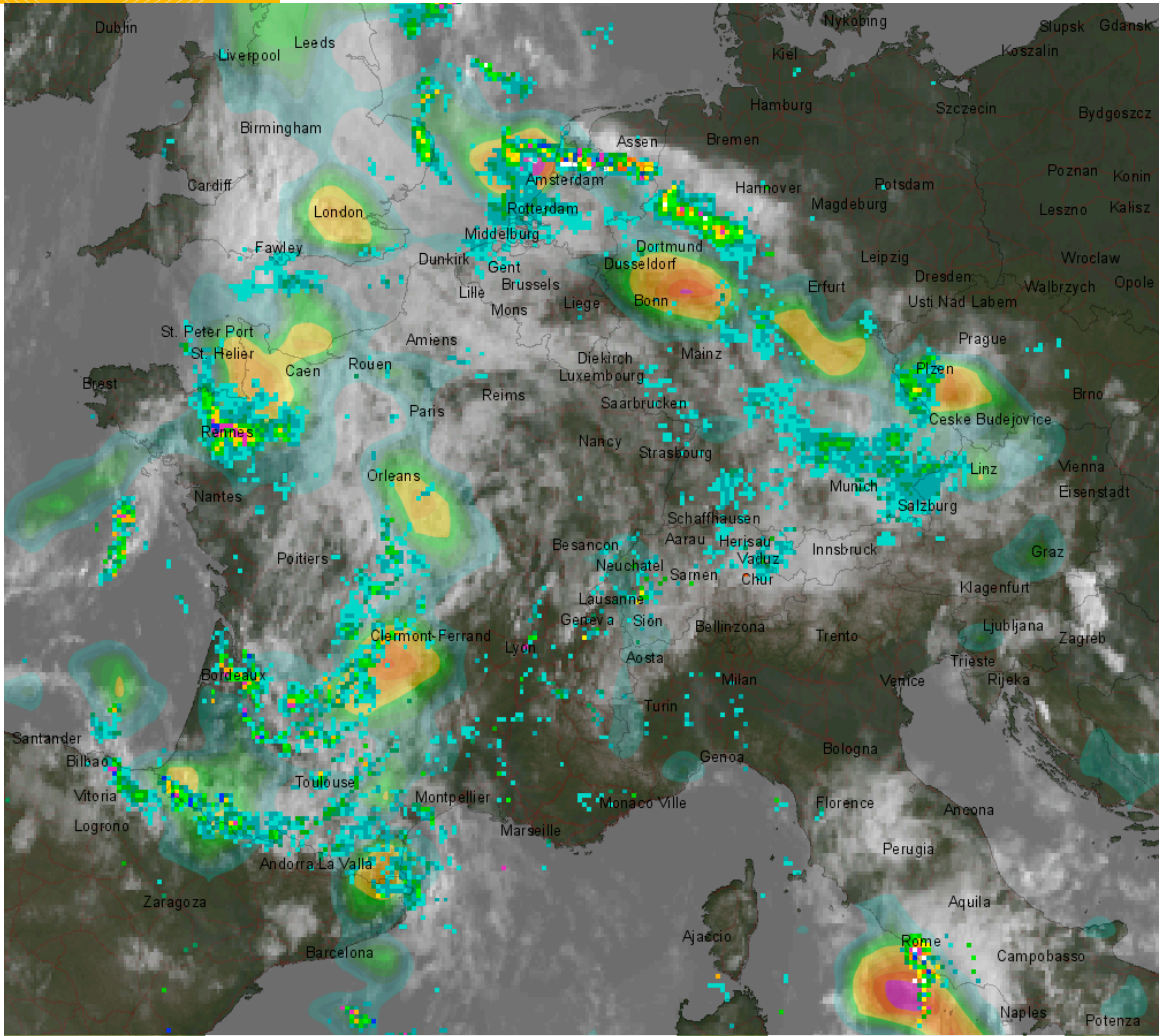




Web Map Service (WMS)

Map-like views of pre-rendered information: HTTP queries to a WMS server return pre-rendered images (layers) from different sources to be layered on top of each other in the client application.

Available layers, projections, and other parameters can be queried with GetCapabilities-request.



- Map
- Roads
- Satellite
- NWM Precipitation
- Radar Precipitation



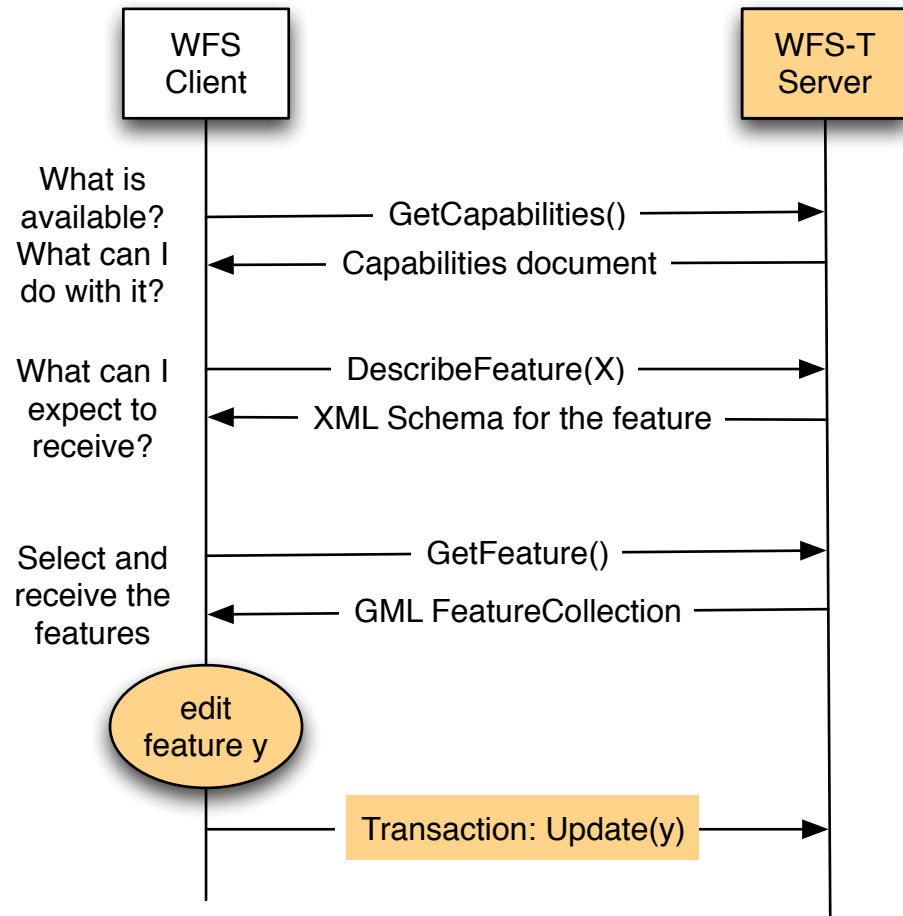
Web Feature Service (WFS)

Web Feature Service defines protocols for requesting as well as (optionally, WFS-T) inserting and modifying GML Features (Geospatial objects) through HTTP requests.

GML Features are geospatial objects semantically describing real-world entities, like Meteorological objects (fronts, precipitation areas), weather observation readings, observation stations or roads.

WFS can be used as a client-server data retrieval, server-to-server data synchronization or as a common access interface on top of heterogeneous geospatial data stores.

Optional Transaction requests (WFS-T) enable inserting, updating, deleting and locking GML Features:





Web Coverage Service (WCS)

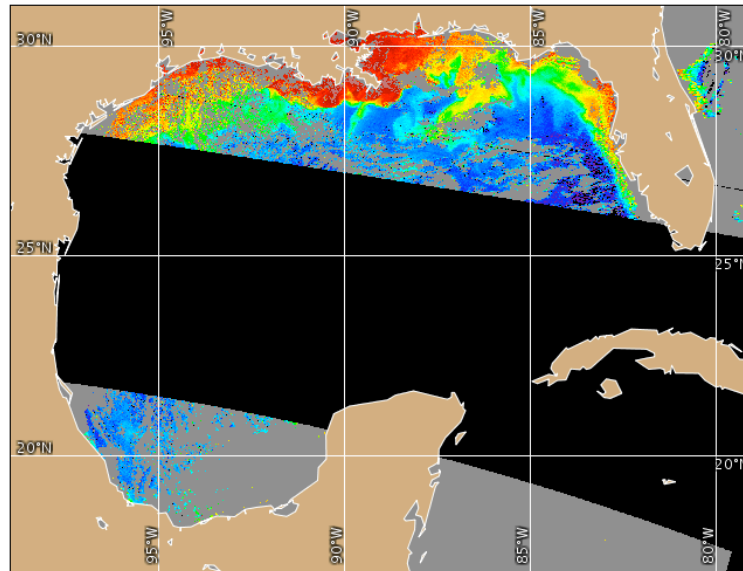
Similar to WMS and WFS, but instead of images or features deals with coverages: collection of mappings from geometry (grid points) into parameter (temperature, pressure,..) values = grid data.

Analogically to WFS, the client requests the WCS server for available data (GetCapabilities), asks it to describe the interesting coverages (DescribeCoverage) and create and return the requested subset of the available coverage data (GetCoverage). Optionally only the HTTP URLs for the response data sets can be returned as an XML document.

Request: GetCoverage

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGetCoverage.xsd"
service="WCS" version="1.1.2">
  <ows:Identifier>Cov123</ows:Identifier>
  <DomainSubset>
    <ows:BoundingBox crs="urn:ogc:def:crs:OGC:2:84">
      <ows:LowerCorner>-71 47</ows:LowerCorner>
      <ows:UpperCorner>-66 51</ows:UpperCorner>
    </ows:BoundingBox>
  </DomainSubset>
  <Output format="image/netcdf"/>
</GetCoverage>
```

Response: a data grid





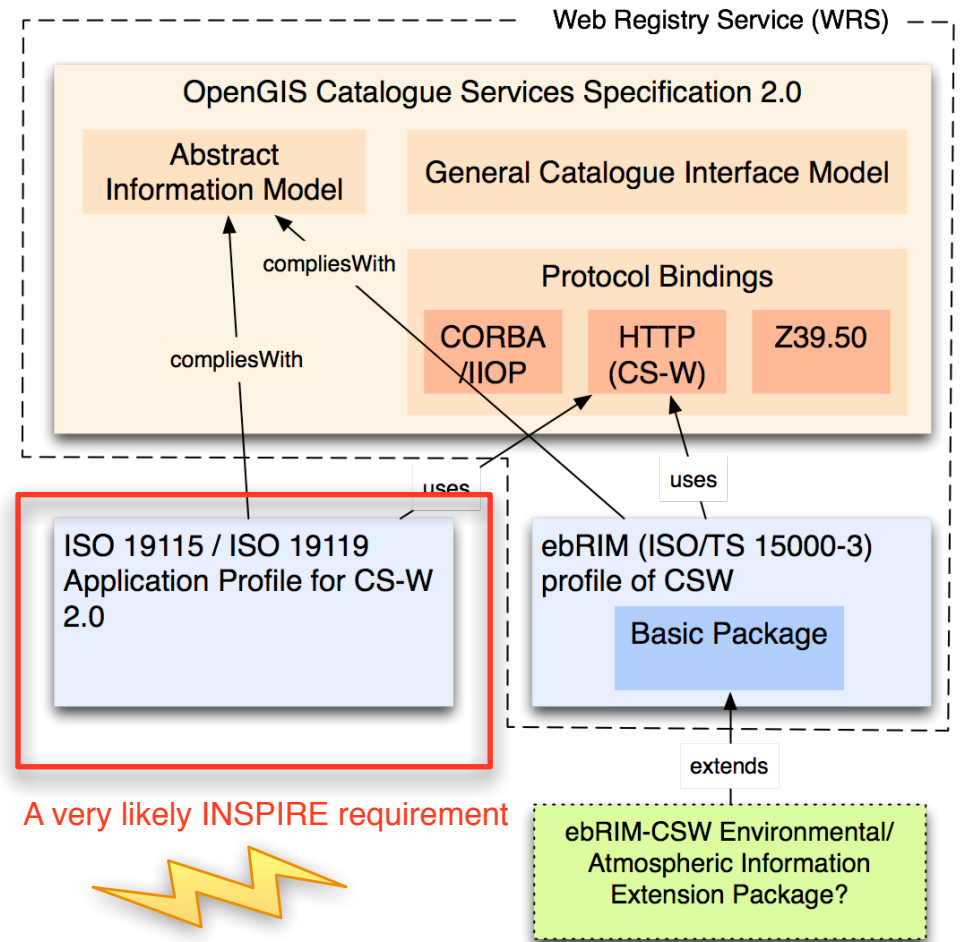
Catalogue Service for Web (CS-W)

"Catalogue services support the ability to publish and search collections of descriptive information (metadata) for data, services, and related information objects."

"Metadata in catalogues represent resource characteristics that can be queried and presented for evaluation and further processing by both humans and software."

"Catalogue services are required to support the discovery and binding to registered information resources within an information community."

(OpenGIS Catalogue Services Specification)



A very likely INSPIRE requirement



OGC: ebRIM is the preferred meta-model for future App. Profiles

CS-W = OpenGIS Catalog Service + HTTP Protocol Binding

Web Registry Service = CS-W with ebRIM AP



Feature Portrayal

WMS can be used as presentation front-end for feature and coverage services: A WMS instance retrieves the data and renders it into an image using predefined rendering rules.

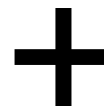
Presentation rules and symbols are described in separate documents having their own language, like Styled Layer Descriptor (SLD) or XSLT (to create SVG or KML).

These presentation rules can be maintained in a CS-W registry and shared with all interested parties.

Dedicated Feature Portrayal Service (FPS) specification is under discussion in OGC

A Cold Front GML Feature

```
</gml:Envelope>
</gml:boundedBy>
<gml:featureMember>
  <metobj:ColdFront gml:id="coldfront_1">
    <gml:name>Unnamed Cold Front</gml:name>
    <gml:boundedBy>
      <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.6:4326">
        <gml:pos srsDimension="2">-20.2293 35.034</gml:pos>
        <gml:pos srsDimension="2">-14.0565 61.447</gml:pos>
      </gml:Envelope>
    </gml:boundedBy>
    <metobj:Path srsName="urn:ogc:def:crs:EPSG:6.6:4326">
      <gml:segments>
        <gml:LineStringSegment>
          <gml:poslist>
            -20.2293 35.034 -20.0914 35.2434 -19.9551 35.4569 -19.8222 35.6788
          </gml:poslist>
        </gml:LineStringSegment>
      </gml:segments>
    </metobj:Path>
  </metobj:ColdFront>
</gml:featureMember>
```



```
<-<LineSymbolizer>
  - <Stroke>
    - <GraphicStroke>
      - <Graphic>
        - <Mark>
          <WellKnownName>Cold</WellKnownName>
        - <Fill>
          <CssParameter name="fill">#0000ff</CssParameter>
          <CssParameter name="fill-opacity">1.0</CssParameter>
        </Fill>
        - <Stroke>
          <CssParameter name="stroke-width">1.0</CssParameter>
          <CssParameter name="stroke-opacity">1.0</CssParameter>
          <CssParameter name="stroke">#0000ff</CssParameter>
          <CssParameter name="stroke-linejoin">round</CssParameter>
          <CssParameter name="stroke-linecap">square</CssParameter>
        </Stroke>
      </Mark>
    <Opacity>30.0</Opacity>
    <Size>28.0</Size>
    <Rotation>0.0</Rotation>
```

A Styled Layer Descriptor (SLD) for Cold Fronts



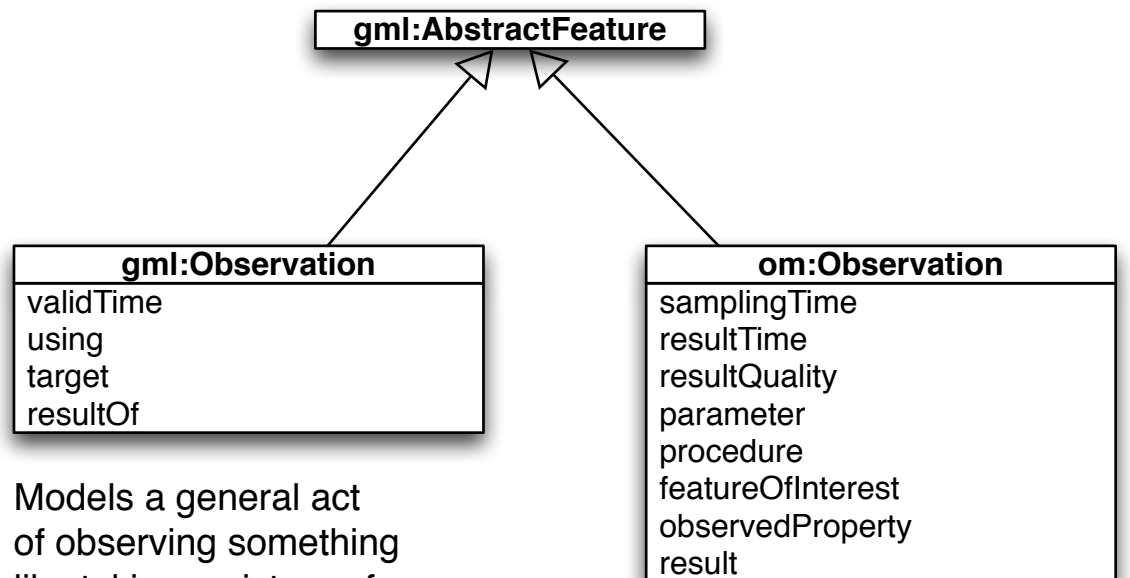


Weather Observations & OGC

Two models for **observation events**:
GML Observation in the GML Core schema and the *O&M Observation* in the OGC Observations & Measurements (O&M) schema.

Meteorological observations may be measurements with numeric values, category observations (generic type of weather, type of cloud), truth-valued (is it raining?).

Results for observations over spatial areas and/or non-singular time instance are coverages.



Models a general act of observing something like taking a picture of a flower

Simple

"Scientific" observation from measurement collection point of view

More complex, more flexible

O&M model suits meteorological observations well, but its properties need to be more specific to enable interoperability in climate-environmental community



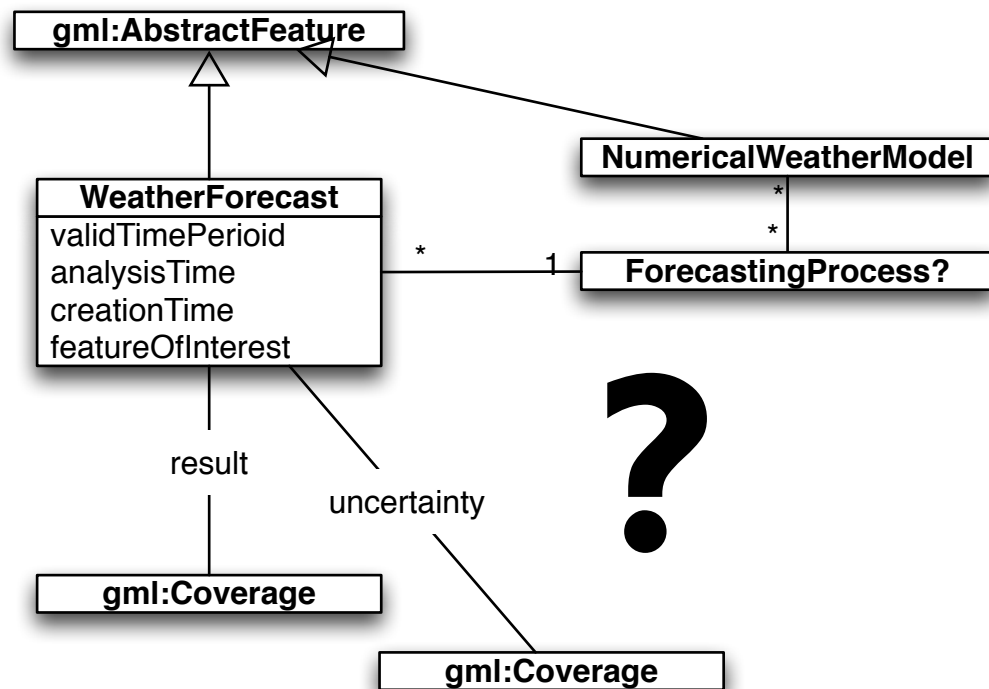
Weather Forecasts & OGC

No obvious GML Application schema for modeling forecasting events exists (as far as I know).

Results are coverages in most cases (grids or collections of point-value pairs); the same data structures as with observations are useful, but the overlapping forecast time periods are problematic to express in queries.

Completely open: Expressing forecast uncertainty? Spatial and temporal interpolation of values?

We, as part of the climate-environmental community, should be more active in the standardization process to enable global interoperability.



Severe weather warnings: Meteoalarm could be a very good case for defining a GML Application Schema for weather warnings and distributing them through WFS.



Experiences & Technologies

We currently have experience of two server products supporting OGC Web Services at FMI: MapServer and Geoserver

UMN MapServer

Implements WMS, WFS 1.0, WCS 1.0, GML 2.1.2 & 3.1.0 Level 0, SLD and Filter Encoding. Supports many raster formats and vector formats through GDAL and OGC libraries: ArcInfo, DODS/OPeNDAP, GRIB1&2, NetCDF, TIFF/ GeoTIFF, JPEG2000, Netpbm, ESRI ArcSDE & Shapefile,...

Well suited for WMS server: easy to configure, stable, direct Oracle SQL queries possible (even without Spatial), PostgreSQL, SQLite. On-the-fly (re)projections of the data using Proj4. Configuration is file-based (like it or not).

Biggest problems with meteorological grid data formats: reading does not always succeed even if it should be supported, probably caused by slight incompatibilities in data formats.

Recommendation: for easy-to-setup WMS



Experiences & Technologies

Geoserver

Implements WMS, WFS 1.0. & 1.1, WCS, SLD, Google Earth KML as WMS output format.
WFS-T

Supports PostGIS, DB2, Oracle, MySQL and ESRI Shapefiles as input data sources.
Possibility to build more with GeoTools DataStore API. Streaming data: no memory limits for serving large datasets.

Deployed as a standard WAR package inside Java Servlet containers. Integrated Web GUI for configuration, good or bad depending of if you like a click-through or write-through setup.

Currently outperforms MapServer in WFS support.

Directly supports only simple features: feature properties must match columns in the DB tables. Support for complex features is on the roadmap.

Recommendation: For building a Java EE based, extensible geospatial data server, especially for WFS use



OGC Web Service –Related Projects at FMI

Ubcasting

WMS: radar data,
WFS: point observations and forecasts,
WFS-T: chemical release features, triggering the dispersion model calculations

Cooperating EO Sensors (COPS)

WMS: Weather information and air quality maps (temperature, wind,..)
WFS: Observation data from Helsinki Testbed area
WCS: MODIS satellite data received in Sodankylä station

Smartmet II

WMS: Client for background maps, or any other geospatial layers for the workstation
WFS-T: Meteorological objects created by the meteorologists (fronts, symbols,...)



Ubcasting

Goal: Develop technologies for enabling small-scale highly targeted, near-realtime combined weather & air quality information for anybody, anywhere (ubiquitous computing).

The use cases of the pilot customers: Provide quick response weather and dispersion information for the rescue activities in the case of chemical or radioactive substance releases.

Continuum of the Helsinki Testbed project: The focus at FMI side is to try to build OGC compatible interfaces for providing point observation, point forecast, lightning and radar data for the external partners to use in their applications.

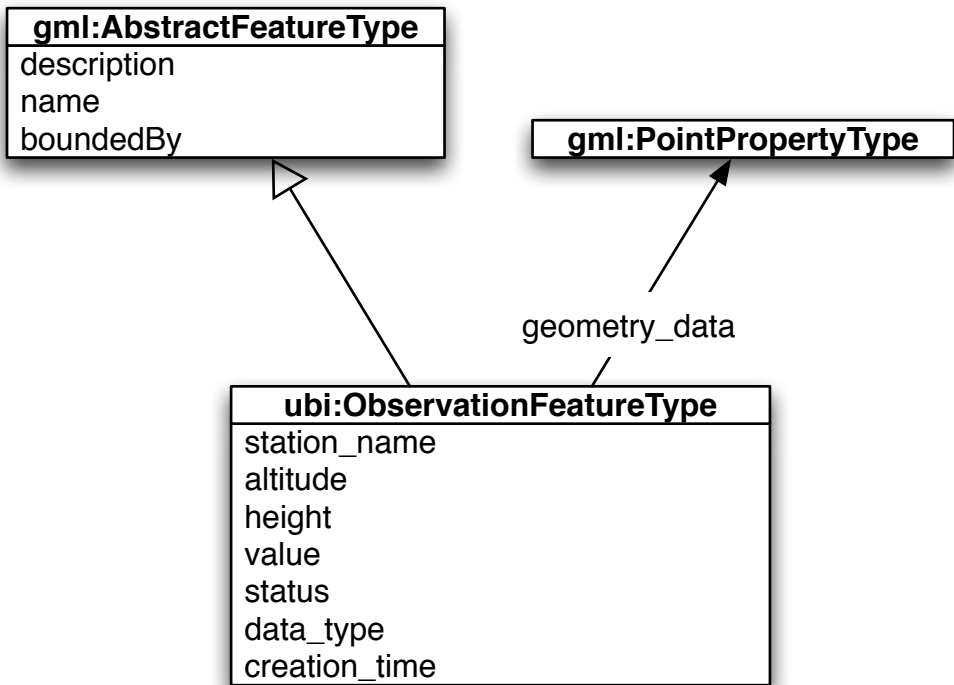
A lot of OGC related discussion and on-going work with Vaisala (a recent OGC member)

Status (June 2008): radar images and simple point forecast features available through Mapserver

During summer-autumn 2008: More elaborate forecast and observation features



Ubcasting Simple GML Features



Good: Can be served by Geoserver with standard setup and easily mapped with PostgreSQL database (one table/view per feature, one column per property)

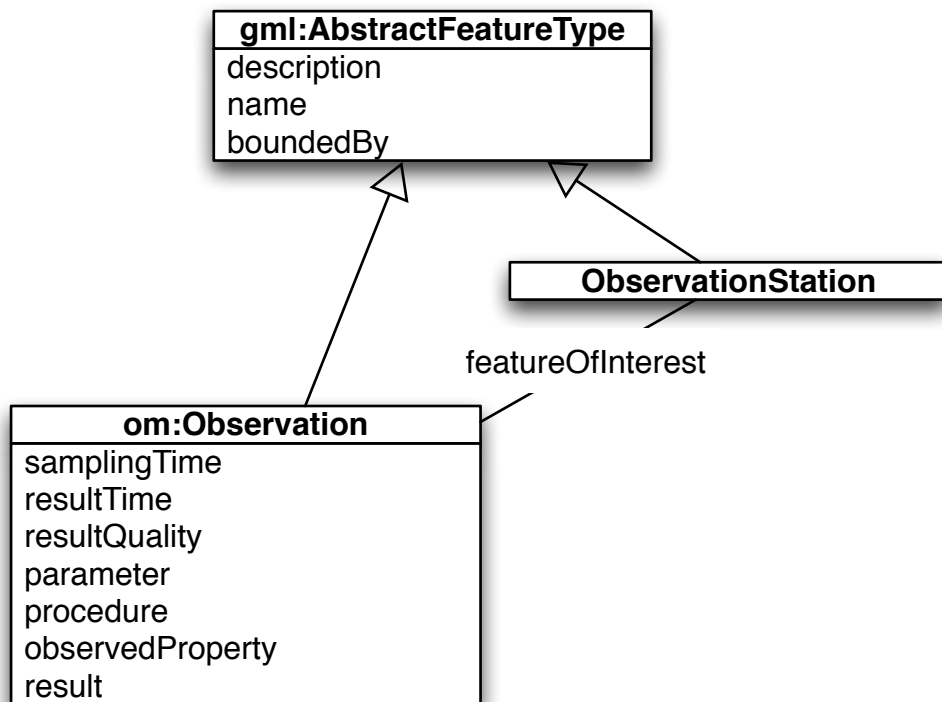
Bad: Verbose and not flexible

WFS GetFeature reply:

gml:FeatureCollection	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
ubi:ObservationFeature	
.	
.	
ubi:ObservationFeature	



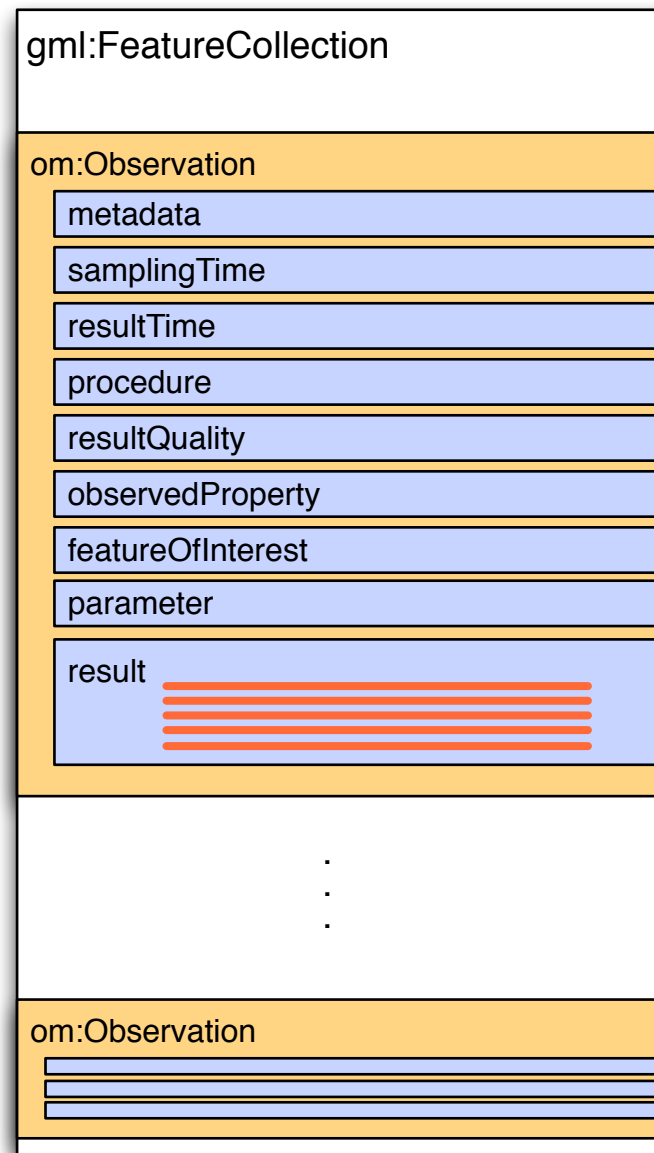
Ubcasting GML Features: Observations & Measurements schema



Very flexible (even too flexible?) structure of the observation events. References to the other GML Features, possibility for expressing many kinds of observations with the same schema (good & bad)

Needs complex DB-object mapping

WFS GetFeature reply:





Cooperating EO Sensors (COPS)

An ESA technology research project for building applications which combine and share in-situ sensor observation data and remote EO (satellite) data using OGC Web Service interfaces.

In the Finnish part, the sensor data from the Helsinki Testbed area is provided by using WFS, and MODIS satellite data coverages by using WCS.

Mixing height and Surface layer model results are provided both as data (WCS) and pre-visualized images (WMS).

OGC Sensor Observation Service (SOS) is used in the Spanish subsystem, but from the FMI side we want to use the same WFS planned in the Ubcasting project to distribute weather observations => WFS-SOS adapter. SOS observations are expressed in O&M, so this should not be a big problem.

WCS implementation to use is still a big open question.

Status (June 2008): Detailed technical design phase starting



Smartmet II

The new Java-based Meteorological Workstation in development at FMI

Completely modular (component-based internal architecture), the selected plugin set defines both the GUI features and the operation implementations) => both stand-alone and lightweight, heavily server-dependent configurations possible.

Possibility to include geospatial WMS layers in the display stack (WMS client plugin)

Meteorological Objects editor plugin: visualization and editing of fronts, symbols,.. Serialized as GML Features and stored at the server (could be implemented using WFS-T, but current early version of the server is very simple servlet storing any XML).

Satellite and radar data currently fetched as pre-rendered images through existing HTTP servers. Will move into using WMS and possibly WCS when they become available.

Status (June 2008): early version with MetObject editor plugin released for testing by the meteorologists



OGC Web Services Outlook at FMI for 2008–09

Long-term goal: Simplify the maintenance of the data transfer both in-house and with external partners by using open standard interfaces in the information exchange whenever possible. At the moment OGC Web Services are considered to be a very important part of implementing this scenario.

Gather more information and experience both about the standards and the available implementations through the on-going pilot and research projects.

Take part in community efforts for finding the best practices and improving the standards to better suit the meteorological needs.

Setup a WFS for distributing point observations and forecasts using Geoserver

Find a suitable implementation for delivering satellite data using WCS (Geoserver, THREDDS?)

Continue the planning of an internal Catalog Service, possibly using CS-W