GML Application Schema for Meteorological Objects

Ilkka Rinne
Finnish Meteorological Institute

9th June 2009, EGOWS meeting, De Bilt, Netherlands
Prepare for INSPIRE, Learn OGC Web Services

• OGC Web Service interfaces for public access weather data will be mandated by the INSPIRE directive in EU member states in the next 5-10 years.

• Common transfer and archiving formats and protocols will be increasingly important in the Met domain community in the near future:
  • Climate change, rough and rapidly developing weather conditions, international military and rescue operations, flight weather needs,...

• Acting early is seen clearly beneficial at FMI:
  • Possibility to influence the developing standards.
  • Time to learn the new technologies and possibilities.

• FMI became OGC associate member in early 2009
  • Learn by start using the OGC Web Services also in internal development projects.
Meteological Objects at FMI

- Created by meteorologists on duty using SmartMet II workstation.
- Objects with semantics, not static map images.
- Corrected as new information emerges.
- Used for generating different client-specific products.
Goal: MetObjects storage and retrieval using WFS-T

- MetObjects are stored in a (relational / object) database.
- The middle-tier “application server” needs to talk WFS-T and handle the XML ↔ DB queries and responses.
  - The implementation technology doesn't really matter. All access through the WFS protocol.
- Any WFS capable clients can access the stored MetObjects.
  - Production systems, view-only clients,...
FMI MetObjects GML Application Schema

• An XML language for describing Meteorological Objects
  • Domain-specific schema based on Geography Markup Language (GML) = GML Application Schema.

• Defining MetObjects as GML Application Schema was started in autumn 2008 at FMI.

• Primary goal was to model the meteorological objects hand-drawn by the meteorologists as GML Features.
  • GML has well-defined, ready-to-use geometry properties.
  • Cross-platform and programming language solution, easy to post-process and transform if necessary.
  • Directly usable as data payload between Smartmet II editor and the permanent storage server using WFS-T.

• We also wanted to keep the inter-office data exchange in mind.
FMI MetObjects GML Application Schema

• Designed to the easily extendable
  • a couple of abstract Feature types based on the primary geometry (point, line, surface area, volume) and MetObject collections, concrete MetObject types and elements are derived from these.
  • Modeling work still in the beginning: Some synoptic features (fronts, jets, troughs), generic symbols, cloud and rain areas defined at the moment (version 2009/04/28).

• Hand-written XML Schema due to lack of GML methodology knowledge
  • An automated approach for UML-to-XML-Schema generation is usually preferable, using ShapeChange or FullMoon tools.
  • Writing a valid GML Application Schema directly as XML Schema is tedious and hard work even with good XML tools: Learning GML the hard (but oh so educating) way.
FMI MetObjects GML Application Schema

- Eight versions have been published this far, the latest version is “2009/04/28”
- Not very stable yet, change requests keep on emerging from the real-life forecasting work.
- Urge to do international co-operation on this subject.
  - Goal: a common, OGC (and INSPIRE) compliant MetObjects transfer language for the Met domain.
  - Cross-office weather analysis and forecast information exchange using web services (WFS).
  - Off-the-shelf software tools for handling the MetObjects GML.
It's amazing what you can find in the EGOWS archives while preparing for your presentation abstract...

### EGOWS 1990-2009

From this website the electronic versions of almost all presentations of the EGOWS meetings from 1990 up to now are available.

<table>
<thead>
<tr>
<th>Year</th>
<th>Location</th>
<th>Date</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>De Bilt, Netherlands</td>
<td></td>
<td>link to EGOWS2009-site</td>
</tr>
<tr>
<td>2008</td>
<td>Ljubljana, Slovenia</td>
<td></td>
<td>link to EGOWS2008-site</td>
</tr>
<tr>
<td>2007</td>
<td>Dublin, Ireland</td>
<td></td>
<td>link to EGOWS2007-site</td>
</tr>
<tr>
<td>2006</td>
<td>Budapest, Hungary</td>
<td></td>
<td>link to HMS-site</td>
</tr>
<tr>
<td>2005</td>
<td>Exeter, United Kingdom</td>
<td></td>
<td>link to presentations</td>
</tr>
<tr>
<td>2004</td>
<td>Potsdam, Germany</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>Tromsø, Norway</td>
<td></td>
<td>pdf, 59.4 mb (if too large, download the smaller parts)</td>
</tr>
<tr>
<td></td>
<td>or part 1</td>
<td>pdf, 17.9 mb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp; part 2</td>
<td>pdf, 21.3 mb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp; part 3</td>
<td>pdf, 19.9 mb</td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>Rome, Italy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>Zürich, Switzerland</td>
<td></td>
<td>pdf, 17.9 mb</td>
</tr>
<tr>
<td>2000</td>
<td>Helsinki, Finland</td>
<td></td>
<td>pdf, 10.9 mb</td>
</tr>
<tr>
<td>1999</td>
<td>De Bilt, Netherlands</td>
<td></td>
<td>pdf, 14.3 mb</td>
</tr>
<tr>
<td>1998</td>
<td>Norköping, Sweden</td>
<td></td>
<td>pdf, 13.5 mb</td>
</tr>
<tr>
<td>1997</td>
<td>Toulouse, France</td>
<td></td>
<td>pdf, 12.0 mb</td>
</tr>
</tbody>
</table>
1. Terms of Reference

The goal of the Working Group is to define a common list of Met Objects with their attributes and to propose it as a standard to WMO. To that purpose the WG will have links with National Hydro-Meteorological Services outside Europe.

This goal should be reached within a relatively short time period because some user applications are already using Met Objects with their own definitions.

The Working Group on Met Objects will be a permanent subgroup of EGOWS and have annual meetings in parallel with EGOWS.
Working Group on Meteorological Objects (wgMO)

- Working Group on Meteorological Objects in Interaction with Gridded Fields (wgMO) was founded after the Workshop on Graphical Interaction (within COST-78) in Helsinki in December 1998.
  - Paper describes the wgMO background and goals.
  - BUFR and XML mentioned as possible transfer formats.
What happened to the MetObjects work?

• Was WG meeting held at EGOWS 2002 in Rome or 2004 in Potsdam (no proceedings available in the archive)?

• wgMO is not mentioned in proceedings of the later EGOWS meetings.
  • Was the issue forgotten, put to hold to wait for the WMO ET actions, discontinued for some reason?
  • I'm too fresh to remember (my first EGOWS was in Budapest in 2006). Maybe someone here knows more?

• The wgMO work on meteorological objects should be continued and improved based on the FMI MetObjects schema:
  • Re-model the current features in UML, auto-generate the XML Schema.
  • Review the 2003 MetObject list and continue adding them to the Application Schema.
So, we have the MetObjects GML under control, how about the technology stack? We need:

- a WFS-T server with Complex Feature and GIS (PostGIS, Oracle Spatial, ...) support, and
- a java library for accessing and modifying the MetObjects data in the Workstation as well as in the end-user product visualization.
WFS-T for Complex Features: Deegree-WFS

- **Client:** the library code and architecture seemed promising for handling the MetObjects GML with Java.
  - Only a limited amount of extension work to be done for the needs of the SmartMet II MetObject editor.
- **Server:** the WFS server implementation supported transactions of Complex GML Features, something no other open source WFS could do at the time (as far as I know).
- **BUT** (as we learned during a couple of man months):
  - The GML handling and visualizing library proved to be difficult or impossible to extend due very limited class visibility and non-replaceable Factory implementations.
  - We were unable to configure the Deegree WFS server to map our GML Application Schema to PostGIS.
  - Configuration error handling and documentation is poor.
WFS-T for Complex Features: GeoServer 2.0?

- GeoServer “Community Schema” AKA support for full ISO Feature model (including Complex Features) has been under development CSIRO in Australia since late 2007.

- It seems that the Complex Feature support will be included in the GeoServer 2.0 release (beta2 should be out in mid June 2009)
  - Not officially announced yet.
  - Documentation and tutorials are still incomplete?
  - No possibility to do the Complex Feature mapping using the GeoServer web GUI?
  - We are really looking forward for the 2.0 release at FMI.

- Maybe we should broaden the GeoServer/GeoTools developer community, participation from other met/hydro offices and/or companies?
XML ↔ Java Object Binding: Apache XMLBeans

• So, the server side implementation looks rather good, how about the client (workstation)?

• After the Deegree failure we decided to generate the Java classes for the MetObjects Application Schema using the Apache XMLBeans framework.
  • Works nice, but classes for all the XML Schema elements must be generated in the same build: In this case major part of the whole GML schema in addition to the MetObjects schema, 2600+ classes in a 3.9 MB jar file.

• Wrappers for the generated XMLBean classes had to be implemented in order to build a convenient API for MetObjects Domain Model objects including Swing undo/redo support and modification tracking needed by the SmartMet II MetObjects editor.
  • Candidate for co-operative open source project? Must be hand-in-hand with the App Schema development process.
Conclusions

• Technology and data modeling activities for making meteorological data web-accessible are continuing at FMI
  • Pressure from INSPIRE directive, will to be at the technology front line in adopting the OGC Web Service standards.

• An extendable, met community agreed GML Application Schema for the Meteorological Objects is needed.
  • Revival of the EGOWS wgMO, activities in the brand new OGC Met DWG?
  • Bottom-up testbed activity gradually leading into standardization.

• Open source technology stack for accessing complex MetObject GML using WFS-T standard is coming together.
  • Interesting new possibilities in code/library level developer co-operation being good candidates for open sourcing.