



ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

Implementing Weather Object Editor for SmartMet II

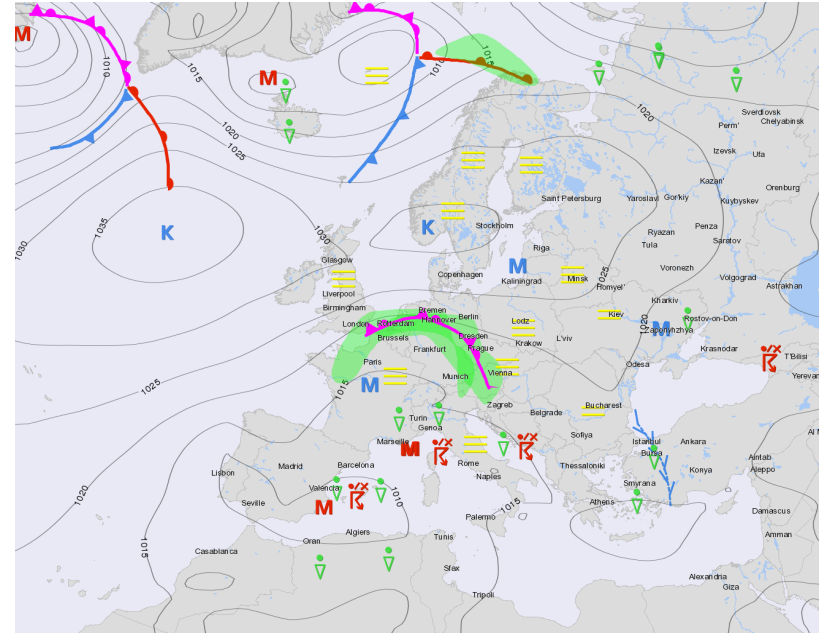
21st EGOWS, ECMWF, Reading, UK

Ilkka Rinne, Finnish Meteorological Institute



Scope: Meteorological Objects

- **(Weather) conceptual models, sensible weather objects, synoptic features, areas for specific type of weather conditions, severe weather warnings, ...**
- **Semantically rich *abstractions* of the observed and predicted weather phenomena and their development in time.**
- **For different audiences: meteorologists, flight personnel, public safety and rescue officials, other citizens,...**



A map (stationary image, animation) is only one type of presentation of the modelled objects.

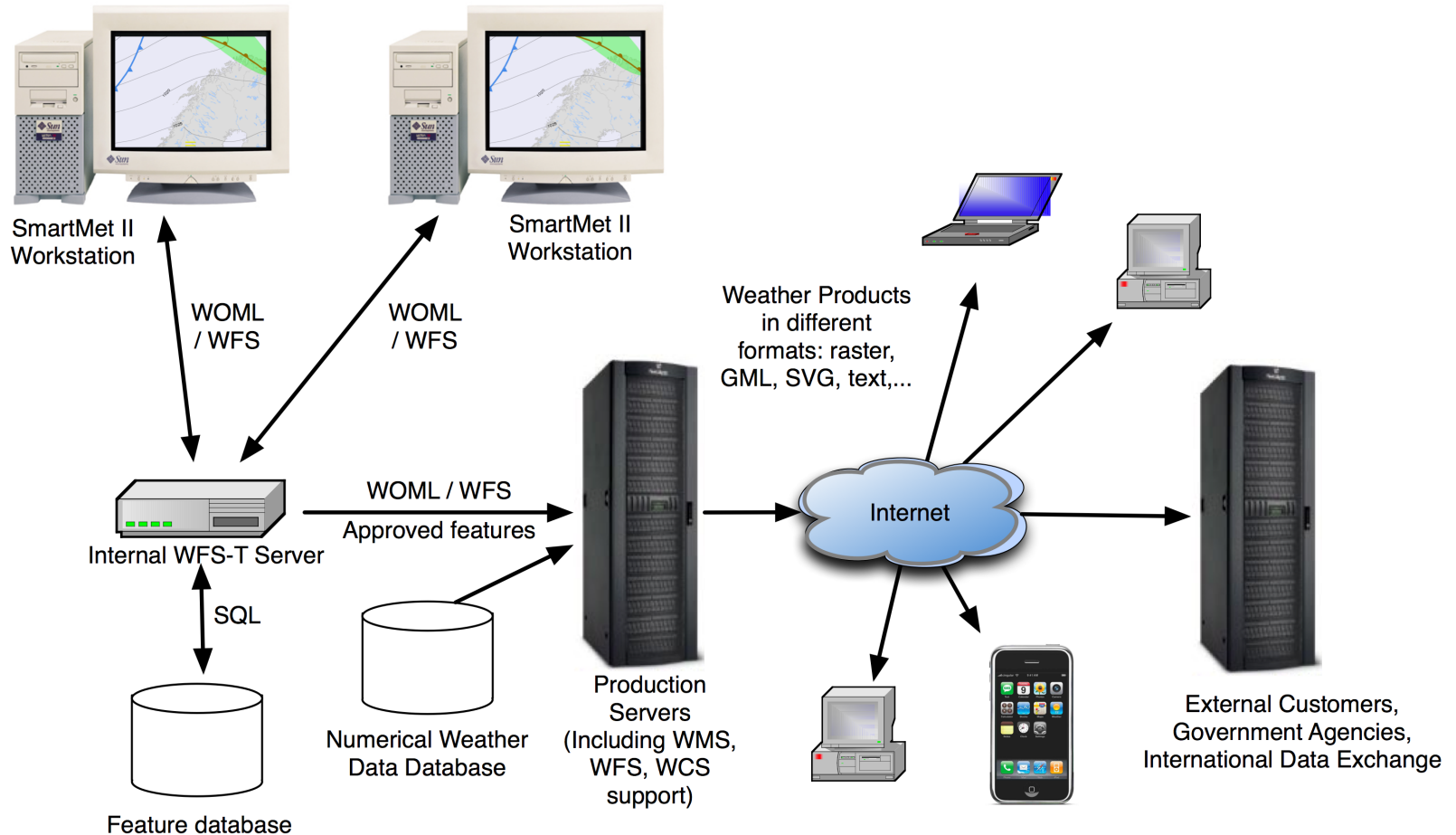


Meteorological Objects as GML Features

- **GML Feature is a good implementation model for identifiable weather abstraction entities:**
 - Geospatial location and shape as functions of time.
 - Standard base language, standard access protocol (OGC Web Feature Service).
 - Integration with other GML based data.
- **An open XML-based format is ideal for post-processing:**
 - Transformations into different text and XML based formats using XSLT.
 - Machine validatable, still human-readable (in theory, at least).
- **Thus: Weather Objects Modelling Language (WOML)**



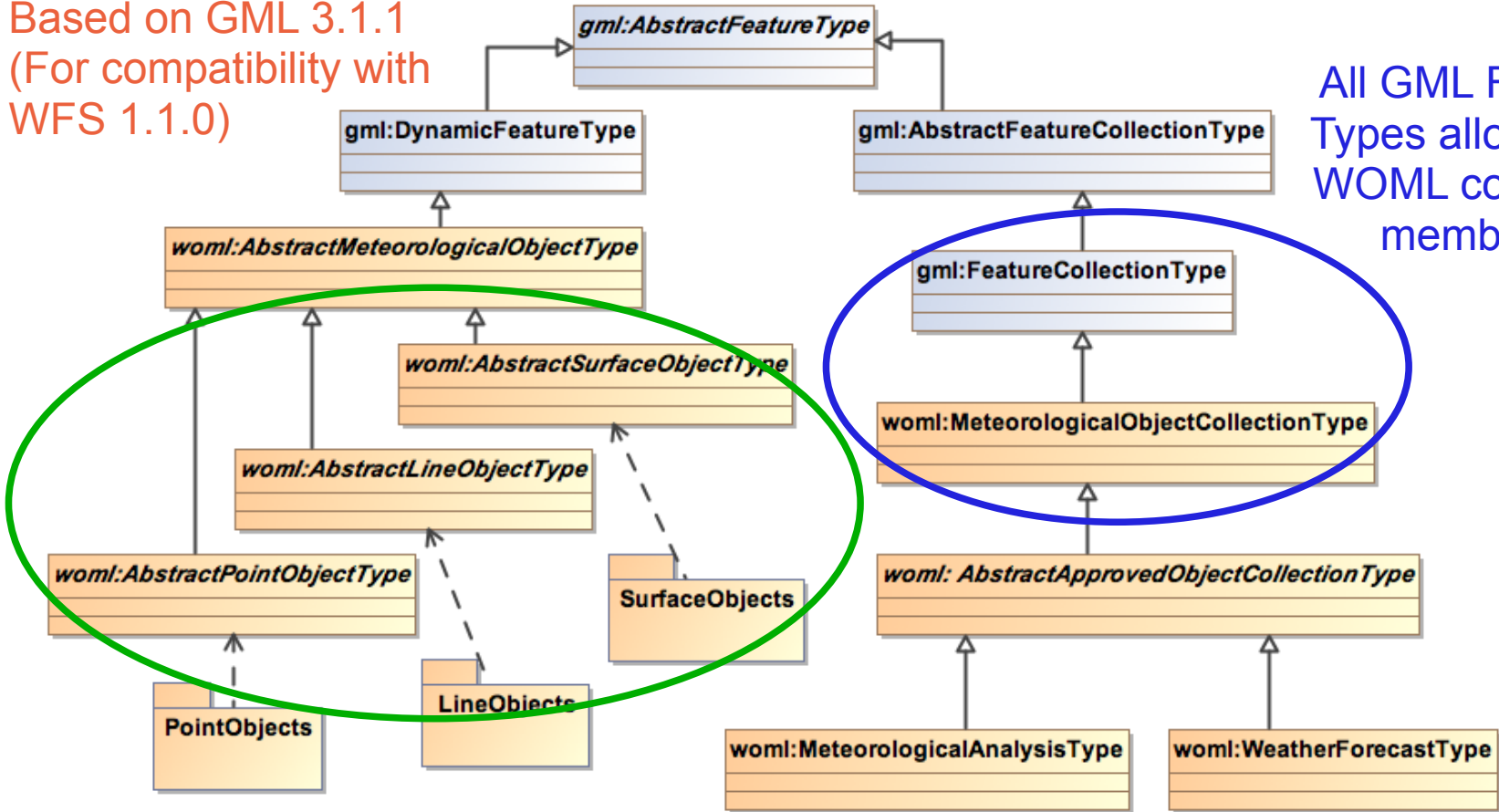
WOML as a common conceptual model for weather analyses or forecasts





WOML Core Types

Based on GML 3.1.1
(For compatibility with
WFS 1.1.0)



All GML Feature
Types allowed as
WOML collection
members

Geometry-based typing:
Extended types can at least be properly geo-located



WOML Editor Components, Case: Fronts

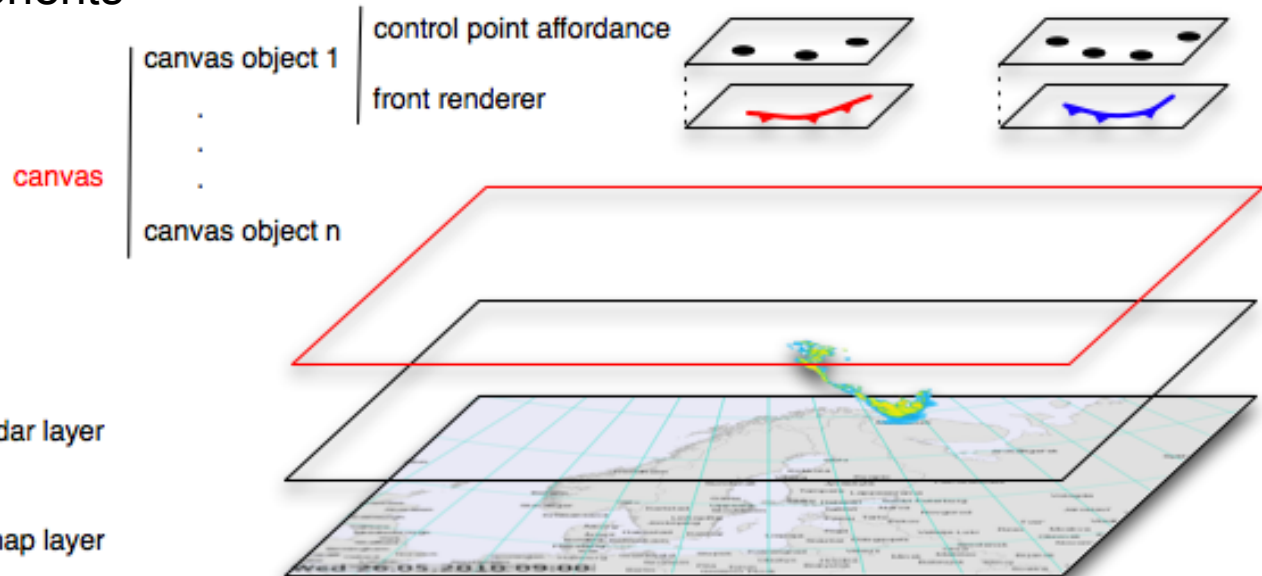
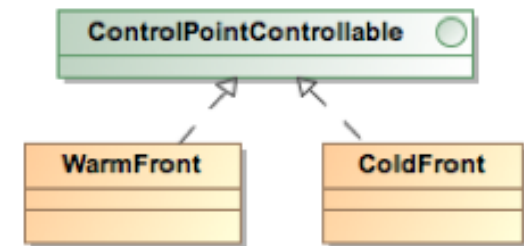
Geometry and other front properties in **model** classes, control point editing operations exposed through a special modification interface.

Renderer components determine feature visualization (**view**).

Affordance components edit the model data through the modification interface (**controller**).

Canvas sits on top of other workstation data layers

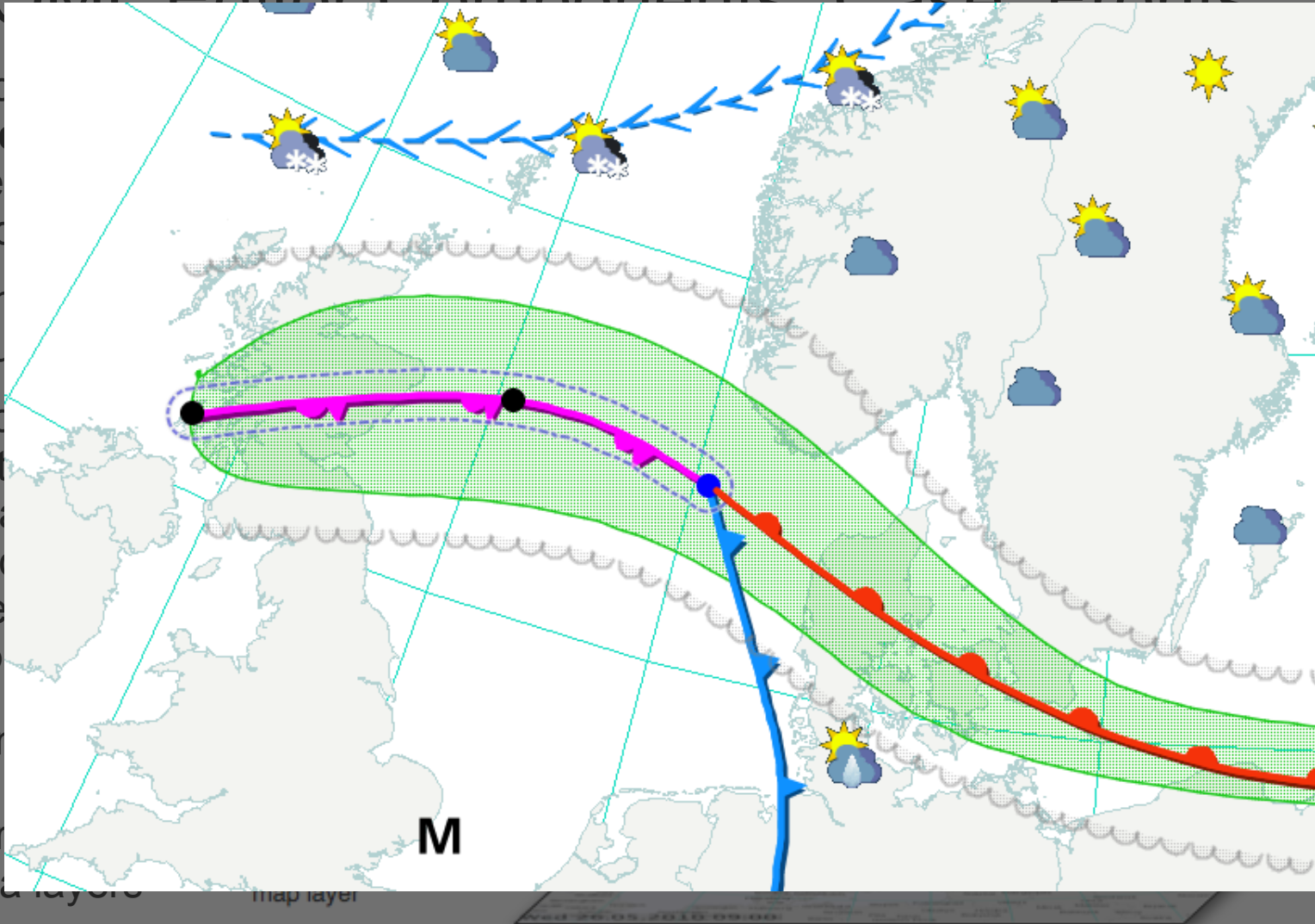
Model-View-Controller (MVC)





WQMI Editor Components Case: Fronts

Geo
mo
ope
mo
Ren
visu
Aff
edit
dat
mo
inte
(co
Car
top
wor
data
layers





Feature Portrayal – GML into images

- **OGC Standards: Styled Layer Descriptor (SLD) and Symbology Encoding (SE)**
 - SLD: Enable style binding for WMS layers
 - Typically WFS or WCS services as back-ends
 - SE: define styling rules for features and (grid) coverages (compare: CSS & HTML)
- **Unfortunately SE is currently not rich enough for defining front decorations and some other more complex visualizations**
 - Rendering rules for WOML objects in SmartMet II WOML Editor are hard-coded for now.
 - Hopefully this will change in the future.



Accessing WOML Data

- **Weather object forecasts and analyses stored on the server in WOML format.**
- **Service plugin in SmartMet II for reading/writing the WOML data using HTTP.**
 - WFS(-T) server side solution under construction
- **Apache XMLBeans used for XML – Java class mapping: automatically generated access library based on given XML Schema files.**
- **Weather object model classes wrap the generated classes and add**
 - convenience access methods,
 - undo/redo support (via stored feature states), and
 - sending data change events to view components,



Editor undo/redo using Java StateEditable

- **The controller components decide which user actions should be undoable (feature state has changed significantly).**
- **Model classes provide feature data state snapshots using XML serialization.**
- **Serialized pre and post states of the modified features are stored in the Java Swing undo stack for future undo/redo actions.**
 - Only for the modified features, not the whole forecast.
 - On add or remove feature the whole forecast state is stored.
- **Upon undo action, the current contents of the changed feature are replaced with the stored pre state.**



Editor: Still Under Construction

- **Proper handling of dynamic Features**
 - Objects moving and changing in time (“dynamic”)
 - Can be expressed in WOML features using gml:history property, support missing for now in the editor.
- **Forecast / analysis metadata editor**
 - Forecast begin & end times, used reference data (numerical models, satellite images etc..).
 - Stored in WOML forecast & analysis objects.
 - Displaying the forecasts together with the data fields actually used for creating them.
 - Available in WOML, very basic implementation in the editor.



Editor: Still Under Construction (2)

- **Uncertainty areas**
 - Degree of forecaster's confidence in the objects existing and being located within a geospatial area at certain point of time.
 - Available in WOML, not supported in the editor yet.
- **Forecast / analysis browser**
 - Find created forecasts and analyses, use as basis for new ones.
 - Currently only very basic indication of previously created data available in workstation timeline control.
 - Requires server interface with good search and preview capabilities (WFS + WMS would be optimal).



Next: Aviation Weather Forecasts

- **The Smartmet II WOML Editor is extended for creating graphical weather forecasts for aviation**
 - SigWx (SWC) charts
 - Graphical cross sections between two locations
 - Graphical vertical profile forecast for a single location
- **Model based on**
 - Weather Information Exchange Model (WXXM) defined by Eurocontrol (SESAR) and FAA (NNEW) version 1.1.1
 - Current WOML features
- **Work currently in early design stage.**



Resources

- **A brand new WOML Home:**
 - <http://agora.fmi.fi/display/WOML/>
 - Documentation, schemas, WOML-Java, presentations
 - Including this presentation
- **Feature Portrayal in Met: OGC MetOcean DWG TWiki**
 - http://external.opengeospatial.org/twiki_public/bin/view/MetOceanDWG/FeatureAndCoveragePortrayal
 - Met community requirements for SLD/SE
 - Still missing a lot of examples, please contribute!
- **Apache XMLBeans: <http://xmlbeans.apache.org/>**