Integrated Software Framework for OGC Web Services

Arne Bröring[1], *Theodor Förster*[2], *Ingo Simonis*[1]

[1] IfGI, Institute for Geoinformatics
Robert-Koch-Str. 26-28, Muenster 48149, Germany
{arneb, simonis}@uni-muenster.de

[2] ITC, International Institute for geo-information and earth observation
P. O. Box 6, 7500 AA Enschede, the Netherlands
foerster@itc.nl

As nowadays different types of geospatial data are available on different sources through OGC Web Services the interest to integrate them is demanding. This is due to the fact that to carry out reliable decisions different types of data from various sources are required. Thus a generic framework is needed, which enables integration of these Web Services. Though a number of approaches are available to support users on a rather abstract level (most of the GI-software providers offer so called suites or portal solutions), support for software developers is hard to find.

Tillman & Garnett (2006) support this integrative notion of OGC Web Services, but only by focussing on *client* applications. As the integration of Web Services on *service* side is also demanding especially in the context of sophisticated web processing and service chaining (Kiehle et al. 2006), we propose an integrative approach for *both* environments. Such an extended approach towards service applications would support the development of sophisticated service chains and decrease the complexity of OGC-based software development.

However until now only specific frameworks are available, either as pure client solutions, such as uDig (http://udig.refractions.net), or as pure service solutions, e.g. deegree (http://deegree.sourceforge.net/). A *generic* solution is still missing. The proposed framework described below offers developers a customizable and extendable system of cooperating classes supplying a reusable design which is applicable for client and server applications.

Looking from the perspective of the Sensor Web Enablement (SWE) initiative (Sliwinski 2005), different sensors and other support data are required to extract reliable information. This case was the driving force for the SWE Working Group within the 52°North Open Source initiative (http://www.52n.org; Kraak et al. 2005) to come up with an integrated framework named the *OGC Web Service Access Framework* (OX-Framework).

It is the aim of the OX-Framework to provide an integrative view to access *all* kinds of OGC Web Services and thereafter to visualize and process the queried data. The variety of different services and data encodings makes it necessary to build up a *flexible* architecture.
The OX-Framework supports flexibility by applying three concepts:

- Layer-Architecture
- Plugin-Concept
- Listener-Concept

The **Layer-Architecture** reduces the complexity of the OX-Framework by structuring it into three layers (Fig 1): Service-Adapters, Core and Utilities.

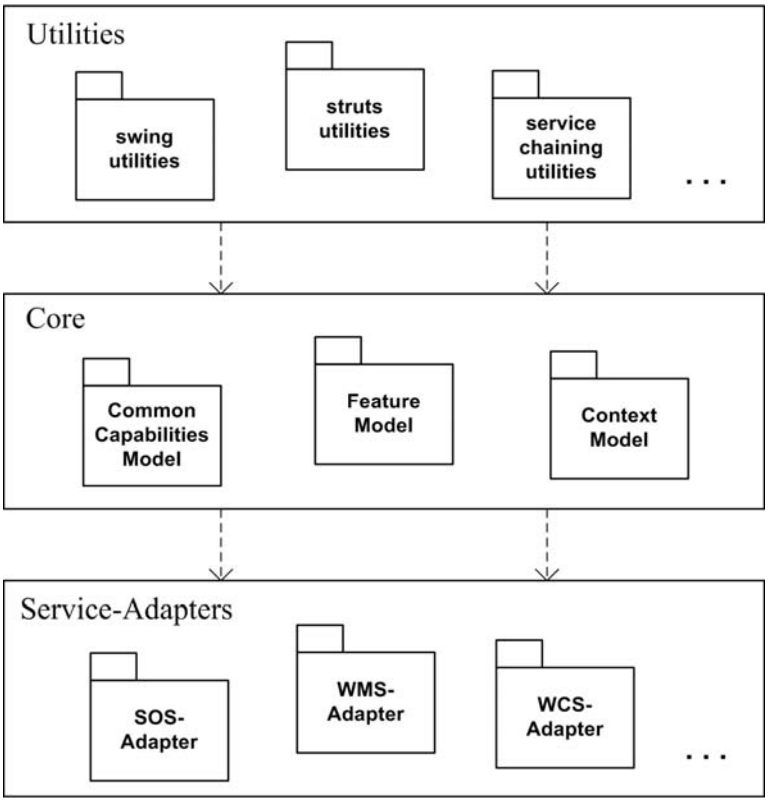

Fig. 1: Layer-Architecture

The *Service-Adapters* layer contains realizations of adapters (see also Fig 2 for an in-detail view) for specific OGC Web Services (e.g. SOS, WMS or WCS). These adapters provide common facilities to the Core for service access in the form of Service-Connectors, data visualization engines (Renderers) and feature marshalling (Feature-Stores).
The Service-Connector initializes the *Common Capabilities Model* of the particular service type and is able to trigger its operations. The Renderer converts the received data to a graphical representation. The Feature-Store provides marshalling facilities for received feature data to the Cores

Feature Model. All the Service-Adapters communicate with each other through the *Core*. This communication is enabled by common data models, which reflect the integrative approach of the OX-Framework.
In detail the Core incorporates a three-folded data model: The *Common Capabilities Model* implements the OWS Common Specification (Whiteside 2005) and introduces thereby the integrative view on service access to the architecture. The *Feature Model* provides a basis for accessing, visualizing and processing of vector data based on (Reynolds 2005; Kottman 1999). The *Context Model* enables persistence and exchange functionality for client projects. It maps a user session – so called "Context" - to an XML-encoding compliant to the Web Map Context Documents specification (Humblet 2003).
The *Utilities* provide functionality for specific UI-frameworks (e.g. Struts or Swing). Those components help the user of the OX-framework to build up a client application or a new service using the framework for service-chaining.

The **Plugin-Concept** enables the developer to customize and extend the framework with the required Service-Adapters in a dynamic way. Hence it is possible to build up client- and service-oriented applications with the OX-Framework.

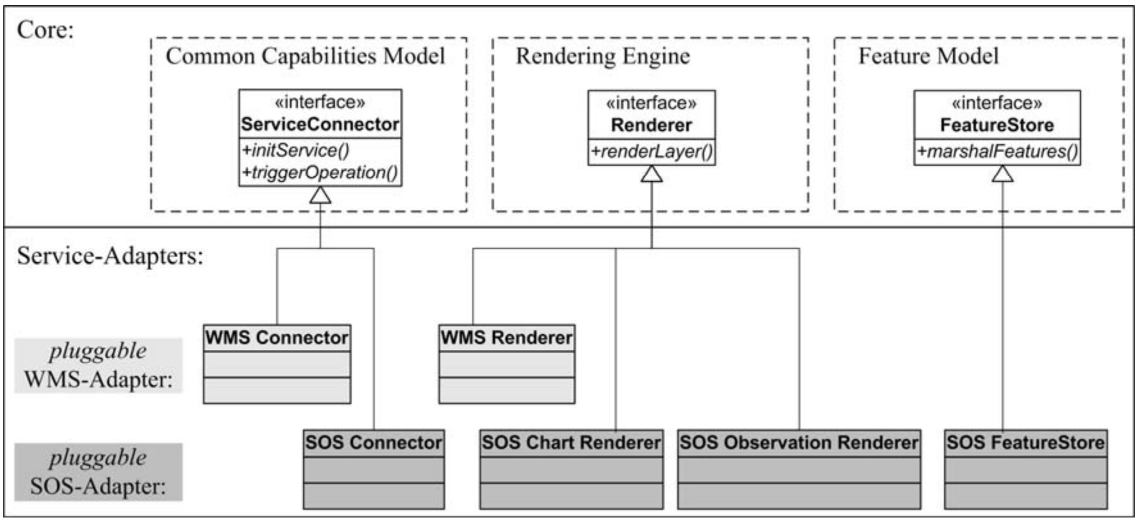

Fig. 2: Plugin-Concept

Additionally the **Listener-Concept** is an important feature of the OX-framework because it affords a high degree of extensibility and transparency which endows the developer with the absolute control over the framework. Figure 3 shows exemplary the functionality of the Listener-Concept.

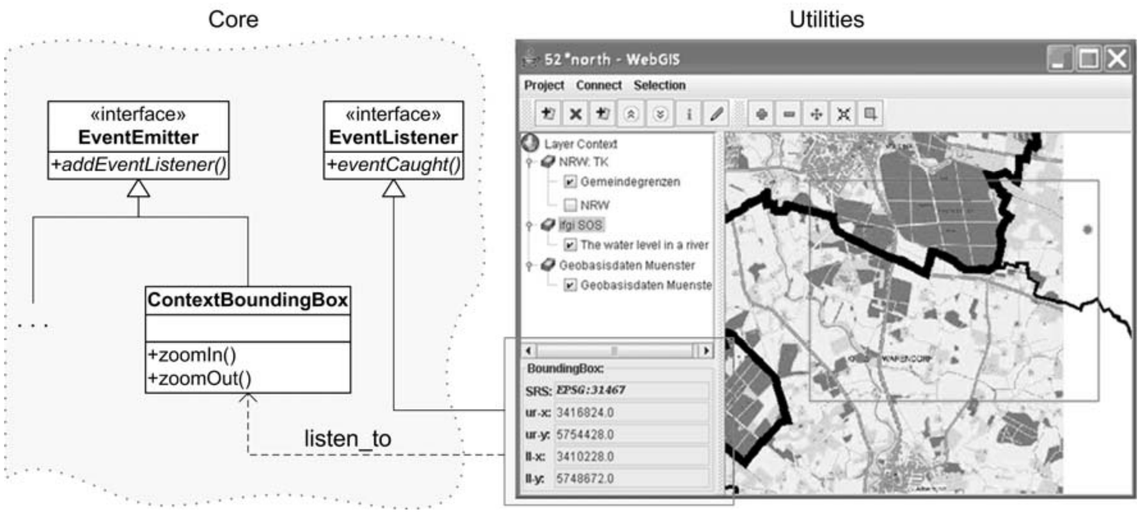

Fig. 3: Example use of the Listener-Concept

The OX-Framework is just evolving within 52°North's SWE Working Group and provides a valid basis for OGC-related software development as demonstrated by the concepts and the implementation at 52°North. Now the framework has to show that it can stand the test in coming sensor web projects. The further development will extend its functionality in sense of web processing, coverage handling and additional Service-Adapter realizations.

References

Humblet, J.-P. (2003). *Web Map Context Documents*. OGC Implementation Specification. OGC Document Number: 03-036r2.

Kiehle, C., K. Greve & C. Heier (2006). *Standardized Geoprocessing - Taking Spatial Data Infrastructures one Step Further*. Proceedings AGILE 2006. 9th AGILE International Conference on Geographic Information Science: 273–282. Visegrad, Hungary.

Kottman, C. (1999). *The OpenGIS™ Abstract Specification Topic 5: Features (Version 4)*. OGC Abstract Specification. OGC Document Number: 99-105r2

Kraak, M.-J., A. Sliwinski & A. Wytzisk (2005). *What happens at 52N? An Open source approach to education and research*. Joint ICA commission seminar 6.-8. July 2005, part of the 22nd ICA conference, ICC 2005: Mapping approaches into a changing world, 2005, 16-20.

Reynolds, G. (2005). *GO-1 Application Objects*. OGC Implementation Specification. OGC Document Number: 03-064r10.

Sliwinski, A., I. Simonis et al. (2005). *Boosting the OGC Sensor Web Enablement Initiative by Open Source Web Services – The Case of 52°North*. AGIT 2005, July 6 - 8, 2005.

Tillman, S. & J. Garnett. (2006). *OWS Integrated Client Architecture, Design, and Experience*. OGC Discussion Paper. OGC Document Number: 05-116.

Whiteside, A. (2005). *OGC Web Services Common Specification*. OGC Implementation Specification. OGC Document Number: 05-008.