OSGeo Journal

The Journal of the Open Source Geospatial Foundation

Volume 3 / December 2007

2007 FREE AND OPEN SOURCE SOFTWARE
FOR GEOSPATIAL (FOSS4G) CONFERENCE
VICTORIA CANADA SEPTEMBER 24 TO 27, 2007

Proceedings of FOSS4G 2007

A Model-Driven Web Feature Service for Enhanced Semantic Interoperability

Peter Staub

Introduction

This article addresses current research issues in the field of interoperability of heterogeneous GIS. We focus on heterogeneity at the level of conceptual data models. The presented research project of a model-driven Web Feature Service aims at enhancing semantic interoperability. The approaches of data interoperability such as OGC web services (OWS) are combined with methods of model interoperability. The model-driven approach of conceptual data modelling allows for keeping data models independent from any particular system.

Interoperability is a crucial capability to deal with in the context of geospatial applications and information communities. The use of web services is well-established and useable in a standardised way due to the efforts of the OGC. However, OWS such as the Web Feature Service (WFS) (3) provide data interoperability, but no model interoperability. Conceptual model mappings are a precondition for semantic interoperability but are not supported by OWS.

Among European initiatives for geodata infrastructures – such as INSPIRE[13] – the need for interoperability not only on the data level, but also on the model level, grows. The research project described in this article was initiated in the context of a project in the Lake Constance region[14]. The mentioned project aims at creating a cross-border web-based GIS for applications.

In the presented research project, we introduce a **model-driven WFS** (**mdWFS**) which combines both the advantages of OWS for data interoperability and those of the model-driven approach for conceptual data modelling. Furthermore, formalism for establishing conceptual model mappings is developed and a prototype is implemented. Because of this combination, the mdWFS we introduce is an approach that provides enhanced semantic interoperability.

Fundamentals of Data Modelling and Semantic Interoperability

The Model-Driven Approach

The main idea of the model-driven approach is to describe (geo-)data models using a conceptual schema language (CSL). The use of a CSL for modelling allows for keeping data structures independent from any particular system or transfer format such as XML or GML. Virtually any transfer format can be derived from the conceptual schema (syn. model) automatically – given an adequate model compiler.

If you want to reach semantic interoperability, you will have to create conceptual model mappings. A conceptual model mapping is converted into mapping functions $\mathcal{F}_M$ from a source schema A to any target schema B:

$$A \xrightarrow{\mathcal{F}_M} B$$

The model-driven approach consists of four steps (see figure 1):

1. Specification of an *application domain* (i. e. “what we are talking about”)
2. Specification of a *CSL* with a coherent UML metamodel
3. Description of the application domain with the chosen CSL $\rightarrow$ *conceptual schema*, platform independent model (PIM)
4. Derivation of any format schema (e. g. a GML Application Schema) $\rightarrow$ *logical and physical schema*, platform specific model (PSM)

As mentioned above, we assume that the generation of the logical schema is *automatically* carried out by a compiler and the encoding is done by an adequate encoding program.

In the presented research project, the (textual) CSL *Interlis* is applied for data modelling. Interlis is a Swiss standard (8) and is widely applied in cadastral and planning applications. Interlis is based on a UML 2 profile and a compiler[15] generates XML schemas (Interlis format) or GML Application Schemas from any given Interlis data model.

[13]INSPIRE project website: http://www.ec-gis.org/inspire/index.cfm
[14]Bodensee-Geodatenpool (Lake Constance geodata pool) project website: http://www.bodensee-geodatenpool.net
[15]Interlis compiler: see http://www.interlis.ch. The compiler is free and open source.

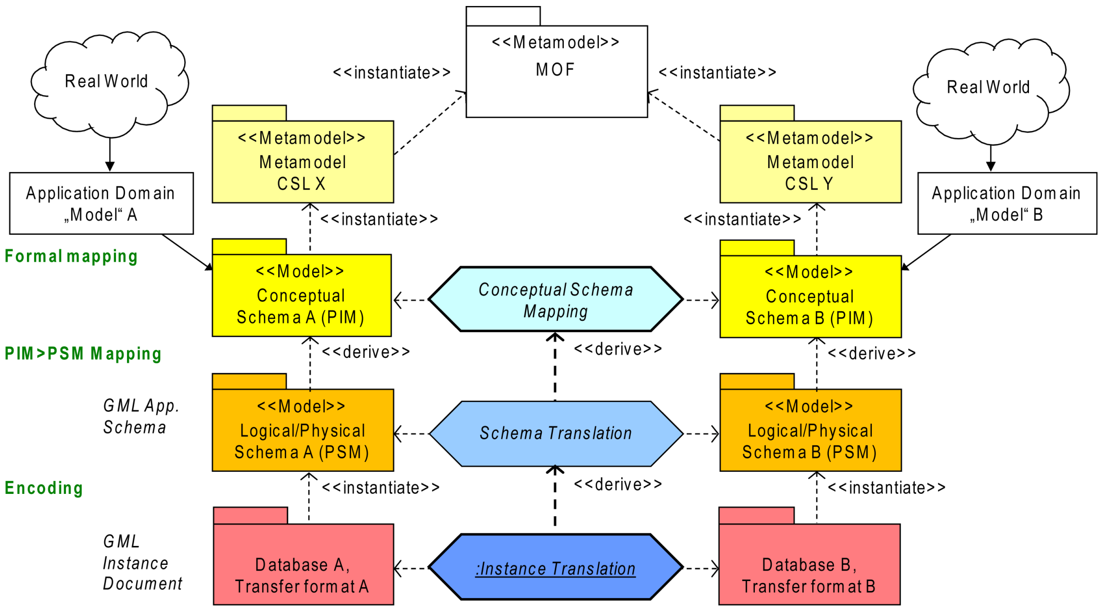
Real World
Application Domain „Model" A
<<instantiate>>
<<Metamodel>> MOF
<<instantiate>>
Real World
Application Domain „Model" B
<<Metamodel>> Metamodel CSL X
<<Metamodel>> Metamodel CSL Y
<<instantiate>>
<<instantiate>>
Formal mapping
<<Model>> Conceptual Schema A (PIM)
Conceptual Schema Mapping
<<Model>> Conceptual Schema B (PIM)
PIM>PSM Mapping
<<derive>>
<<derive>>
<<derive>>
GML App. Schema
<<Model>> Logical/Physical Schema A (PSM)
Schema Translation
<<Model>> Logical/Physical Schema B (PSM)
Encoding
<<instantiate>>
<<derive>>
<<instantiate>>
GML Instance Document
Database A, Transfer format A
:Instance Translation
Database B, Transfer format B

Figure 1: Model-driven approach and semantic interoperability

According to the Model-Driven Architecture (MDA) specified by the Object Management Group (OMG) (4), the generation of a format schema from a conceptual schema is referred to as a PIM–PSM mapping. In terms of mapping directions, the PIM–PSM is also called a "vertical" mapping, whereas model mappings for semantic interoperability are "horizontal" (i. e. PIM–PIM) mappings (see next section).

Semantic Interoperability

Technically, there are two main aspects characterising "interoperability": 1) *Data interoperability* = the ability of a system or process to exchange datasets using certain data transfer formats. 2) *Model interoperability* = the ability to establish conceptual model mappings in order to execute semantic data transformations.

To achieve semantic interoperability, different data models have to be mapped. A translator then interprets the mapping rules from the conceptual model mapping and executes the instance translation automatically.

Semantic transformation approaches can be classified as follows (1):

- *Level of abstraction:* Semantic transformation can be performed on different levels of abstraction (on the conceptual level, on the logical level and on the physical (i. e. transfer format) level). A semantic transformation on the conceptual level is platform independent, whereas approaches on the logical or physical level are platform specific.
- *Orientation:* Horizontal semantic transformation between different schemas on the same level of abstraction (PIM–PIM; PSM–PSM) vs. vertical semantic transformation between different levels of abstraction (PIM–PSM).
- *Level of automation:* Creating mapping rules by hand vs. automated schema matching which is only partially practicable.

Shortcomings of Existing Approaches

One possibility is to integrate all data into one central system. This is very costly and requires expert knowledge. In order to integrate the data into the central system, 1:1-format conversions have to be carried out. This is often lossy because a data format which is different from the original one is in general not able to express the entire semantics of the original data format. Besides that, the inevitable redundant data storage possibly causes outdated data.

Existing OWS such as the WFS have some shortcomings with regard to semantic interoperability: OWS allow for syntactic interoperability (i. e. data interoperability) but not for semantic interoperability (i. e. model interoperability). Conceptual models of source systems are hidden from target systems and semantic transformations are not supported. So, the

WFS lacks in the ability to handle conceptual model information aside from data information.

Concept of the Model-Driven WFS

Preconditions for a Web-Based Semantic Transformation

If we want to have a web service that allows for data interoperability *and* that is able to store and deliver conceptual schemas, a number of preconditions must be fulfilled. It must be assured that conceptual schemas are described (i. e. modelled) using a textual CSL with its graphic representation in UML 2 (and the respective exchange format XMI). Furthermore, a formal language is needed for expressing schema mapping rules on the conceptual level of abstraction. Finally, we use a standard WFS interface to provide satisfying data interoperability.

Web Service Requirements

Web-enabling semantic transformations means in our case actually designing a web service. This service has to comply with two main requirements:

1. Provide access to geospatial data based on the data's original conceptual schema (source model) and on any user-defined conceptual schema acting as the target model.
2. Interoperability with existing OWS.

The mdWFS Interface

We designed a service called "model-driven Web Feature Service" (mdWFS) taking these requirements into account. The mdWFS has the task to store and deliver conceptual schemas and to carry out semantic transformations (PIM–PIM mappings) by means of interpreting transformation models. After a semantic transformation, the mdWFS configures a standard WFS to provide a service for data interoperability. The standard WFS is configured according the target model but delivers transformed feature data from the source model.

WFS Protocol Extensions

In order to create a WFS that is able to store and deliver conceptual schemas, the OGC WFS specification needs to be extended. In the mdWFS specification, the extensions described below are applied (1):

- To provide a service protocol for the mdWFS, a new request parameter `SERVICE=mdWFS` is implemented.
- The `GetCapabilities` request is extended to provide a `SchemaList`. This list includes each conceptual schema that is available in the service.
- The `DescribeFeatureType` request is extended to provide the `XMI` format for transferring model information.
- Finally, a whole new request `DoTransform` is introduced. This request transfers the conceptual mapping schema to the mdWFS and invokes the semantic transformation.

UMLT, a Conceptual Schema Mapping Language

Concept of UMLT

We introduce a conceptual mapping language that can be used to create conceptual mapping schemas (syn. transformation schemas) for semantic transformations. This formal language must comply with several requirements in order to be useable. Transformation schemas must be comprehensible also for non-computer scientists. Therefore, a UML 2 metamodel as well as syntax for a human useable textual notation (HUTN) is developed. Transformation schemas are represented in visual form (UML activity diagrams), in textual form (derived from Interlis CSL) and XML (i. e. XMI), respectively. Common standards in the field of data modelling are taken into account[16]. We also apply an abstraction layer for (geo-)data types.

Two existing approaches from the OMG were examined. First, the Meta Object Facility Query/Views/Transformations formalism (MOF-QVT) (5): this formalism is designed for the transformation of metamodels, e. g. UML→Java. MOF-QVT models are hard to understand and their visual representation helps little. The standard is complex since it actually consists of three languages: Relations, Core and Operational. Furthermore, the MOF-QVT standard is predominantly applied for PIM–PSM implementation mappings.

Another approach that was examined is UML 2 Activities. UML 2 activity diagrams can be used to describe transformations in terms of activity sequences. A clear description of the semantics and of the transfer format (XMI 2.1) is provided in the Su-

[16]Such as standards from OMG, OGC and ISO

perstructure for UML models. UML 2 models are comprehensible and a number of implementations and open source APIs are available.

Because of the above considerations, our conceptual mapping language is based on an independent extension of the UML 2 metamodel. To specify the language elements, a UML 2 model is created and the textual notation of the language is defined by a set of EBNF grammar rules. At project stage, we call our conceptual mapping language "**UMLT**".

UMLT Language Elements

The language elements of UMLT are an inheritance of UML 2 Activities (7). We introduce the following language elements (see figure 2):

- `StructuredTransformation`
- `SelectionCriteria`: selection of input data through a logical expression.
- `VirtualAssociation`: manage input objects that are not actually associated with an association object. These input objects may have link attributes or foreign key attributes that are evaluated at runtime in order to get calculated relations[17]. During a semantic transformation, such objects can be associated *in a virtual way* if needed. The `VirtualAssociation` is introduced (in contrary to a common "derived association") to provide a means to explicitly specify the join property of the association with the `joinCriteria` expression.
- `TransformationAction`: inheritance from a UML `OpaqueAction` providing an activity element which cannot be structured any further. This is a transformation's elementary action.
- `AssignmentDefinitions`: address primitive types or expressions as value specification.
- `MappingRule`: the actual object mapping. Built as a composition of assignment definitions.
- `AssociationBinding`: selecting associated input objects, one may define how these associations are evaluated during input.
- `JoinType`: an enumeration type to specify the join type of the association binding.

Prototype Implementation

In the context of the presented research project, a proof-of-concept prototype is implemented. Besides the WFS protocol extension and the UMLT language specification, this prototype consists of a model parser, a mapping model editor and a prototype test bed. The model parser and the editor are developed in the Eclipse environment[18]. The model parser creates an XMI file from a UML/Interlis data model and also from a UMLT mapping model.

In the prototype test bed, we use an ORACLE Spatial database and a deegree WFS implementation[19] on which the mdWFS is built upon. Figure 3 shows the steps of a semantic transformation using mdWFS.

We primarily focused on the WFS extension and on the conceptual mapping language UMLT. We consider ORACLE Spatial as a very suitable RDBMS for our needs, providing powerful spatial features. Therefore, we use the RDBMS we already had at hand although it is not a FOSS solution. Principally, an mdWFS can be applied on any RDBMS with a spatial extension.

Before you can start working with mdWFS, you need to configure the database according to the source data model A. This can be done using an existing FOSS tool called "ili2ora"[20]. This tool allows to configure an ORACLE Spatial database according an UML/Interlis-data model and to import feature data into this database.

1. Client B sends a model-catalogue request to the mdWFS
2. The mdWFS provides a catalogue of available data models
3. Client B chooses a source data model (i. e. model A) and orders the model information
4. The mdWFS fetches model information A and sends the model (XMI) or a model reference to the client B
5. Client B creates the model mapping $M : A \xrightarrow{\mathcal{F}_M} B$ by specifying adequate UMLT mapping rules
6. The transformation model *and* the target model B are parsed and translated into XMI and sent to the mdWFS in a `DoTransform`-request
7. According to the target model B, the mdWFS configures an ORACLE Spatial database using ili2ora again

[17] A different example is a geometry/topology relation: a building *on* a parcel
[18] Eclipse: `http://www.eclipse.org`
[19] deegree project page: `http://www.deegree.org`
[20] Source: `http://www.eisenhutinformatik.ch/interlis/ili2ora/`

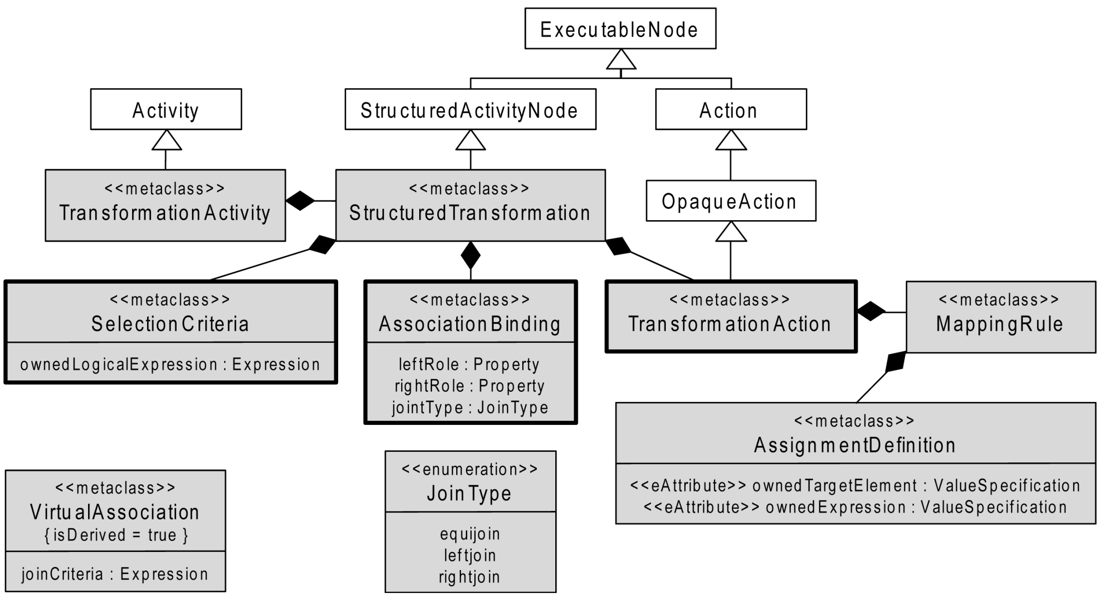
ExecutableNode
Activity
StructuredActivityNode
Action
<<metaclass>> TransformationActivity
<<metaclass>> StructuredTransformation
OpaqueAction
<<metaclass>> SelectionCriteria
ownedLogicalExpression : Expression
<<metaclass>> AssociationBinding
leftRole : Property
rightRole : Property
jointType : JoinType
<<metaclass>> TransformationAction
<<metaclass>> MappingRule
<<metaclass>> AssignmentDefinition
<<eAttribute>> ownedTargetElement : ValueSpecification
<<eAttribute>> ownedExpression : ValueSpecification
<<metaclass>> VirtualAssociation
{ isDerived = true }
joinCriteria : Expression
<<enumeration>> JoinType
equijoin
leftjoin
rightjoin

Figure 2: UMLT language elements

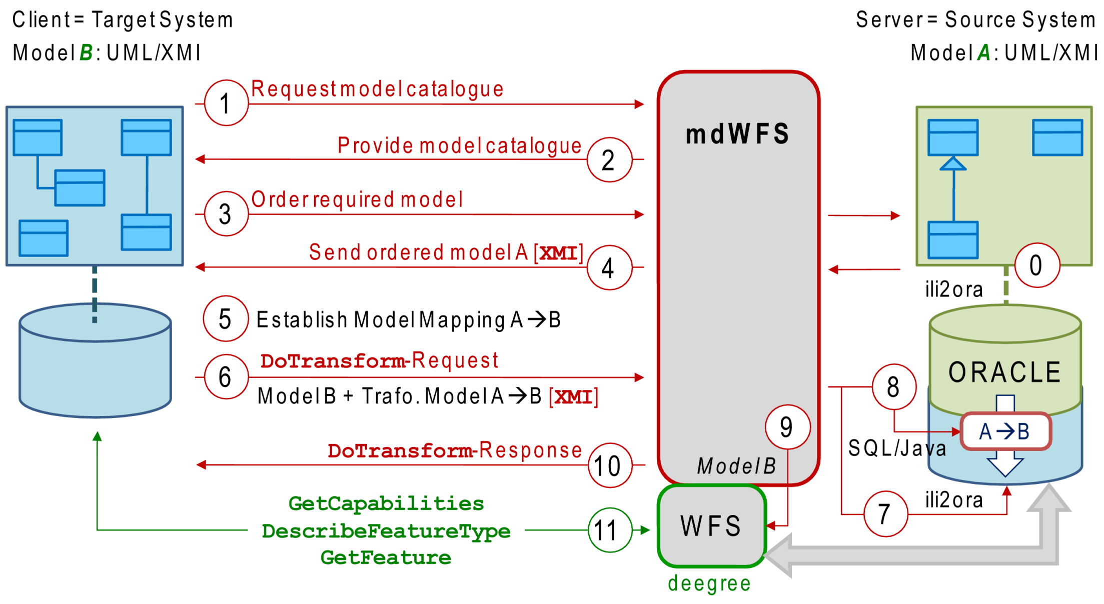

Figure 3: Prototype implementation test bed

8. The mapping rules from the transformation model are translated into SQL statements and Java instructions in order to actually transform feature data from source/server model *A* into target/client model *B*
9. The mdWFS configures a standard WFS (deegree) according to the target model *B*. This WFS is connected to the database containing the transformed feature data
10. After finishing the transformation, the mdWFS sends a `DoTransform`-response to the client *B*
11. Client *B* accesses the transformed feature datasets from model/database *A*, transformed into model structure *B*, via standard WFS requests.

Conclusions

The current evolution of GI systems shows that a conceptual schema language is usually applied for geodata modelling. This is a necessary precondition for semantic transformations on the conceptual level. Any given application domain can be characterised by different data structures. This leads to different data models. Therefore, conceptual model mappings must be established in order to achieve semantic interoperability.

The new *mdWFS* presented in this article implements the methodology of the semantic transformation at the conceptual level of abstraction what allows for a much enhanced semantic interoperability.

Potentially, the mdWFS can be integrated in other (OWS based) infrastructures due to the sound basis of GI standards that are applied.

Acknowledgements

I would like to thank my colleagues at the ETH Zurich and at the TU Munich for their company and good collaboration in this research project: Dr.-Ing. A. Donaubauer, H. R. Gnägi, A. Morf, F. Straub. I also appreciate the excellent project lead of Prof. Dr. A. Carosio, ETH Zurich and Prof. Dr.-Ing. M. Schilcher, TU Munich.

This project is co-funded by the German Federal Office for Cartography and Geodesy BKG and the Swiss Federal Office for Topography swisstopo.

Bibliography

[1] A. Donaubauer, F. Straub, M. Schilcher (2007) mdWFS: A Concept of Web-enabling Semantic Transformation. Proceedings of the 10th AGILE Conference on Geographic Information Science, 2007, Aalborg.

[2] H. R. Gnägi, A. Morf, P. Staub (2006) Semantic Interoperability through the Definition of Conceptual Model Transformations. Proceedings of the 9th AGILE Conference on Geographic Information Science, 2006, Visegrád.

[3] OGC Open Geospatial Consortium (2005) Web Feature Service Implementation Specification: 1.1.0. OpenGIS implementation specification OGC 04-094.

[4] OMG Object Management Group (2003) MDA Guide Version 1.0.1. OMG specification omg/2003-06-01.

[5] OMG Object Management Group (2005) MOF 2.0 Query/Views/Transformations Specification. OMG specification ptc/05-11-01.

[6] OMG Object Management Group (2005) MOF 2.0/XMI Mapping Specification, v2.1. OMG specification formal/05-09-01.

[7] OMG Object Management Group (2007) UML Unified Modeling Language: Superstructure, version 2.1.1. OMG specification formal/2007-02-05.

[8] SNV Swiss Association for Standardization (2006) INTERLIS Reference Manual, version 2.3. Swiss standard SN 612031.

Peter Staub
ETH Zurich, Institute of Geodesy and Photogrammetry, GIS Group
http://www.gis.ethz.ch
peterstaub AT ethz.ch

The Open Source Geospatial Foundation, or OSGeo, is a not-for-profit organization whose mission is to support and promote the collaborative development of open geospatial technologies and data. The foundation provides financial, organizational and legal support to the broader open source geospatial community. It also serves as an independent legal entity to which community members can contribute code, funding and other resources, secure in the knowledge that their contributions will be maintained for public benefit. OSGeo also serves as an outreach and advocacy organization for the open source geospatial community, and provides a common forum and shared infrastructure for improving cross-project collaboration.

Published by OSGeo, the OSGeo Journal is focused on presenting discussion papers, case studies and introductions and concepts relating to open source and geospatial software topics.

Proceedings Editorial Team:

- Angus Carr
- Mark Leslie
- Scott Mitchell
- Venkatesh Raghavan
- Micha Silver
- Martin Wegmann

Editor in Chief:
Tyler Mitchell - `tmitchell AT osgeo.org`

Acknowledgements
Various reviewers & the GRASS News Project

The *OSGeo Journal* is a publication of the *OSGeo Foundation*. The base of this journal, the LaTeX 2ε style source has been kindly provided by the GRASS and R News editorial board.

Journal online: `http://www.osgeo.org/journal`
OSGeo Homepage: `http://www.osgeo.org`
Postal mail: OSGeo
PO Box 4844, Williams Lake,
British Columbia, Canada, V2G 2V8
Telephone: +1-250-277-1621

9 771994 189002 00

ISSN 1994-1897