# Semantic Transformation by Means of Conceptual Modeling Techniques

Andreas Morf[1], Peter Staub[2]
[1] Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology
Zurich, CH
morf@geod.baug.ethz.ch
[2] Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology
Zurich, CH
staub@geod.baug.ethz.ch

## SUMMARY

*Nowadays the application of conceptual modelling techniques which allow a continuous design workflow is almost ever restricted to proprietary systems. Within geo-information communities and spatial data infrastructures the need for communication and definition of guidelines based on conceptual, system independent level is growing. The development and adoption of exact formal schema languages is meeting these requirements but didn't yet reach a level where interoperability is supported by description of mapping rules allowing semantic transformations between different application schemas. This paper will provide an introduction to an ongoing research project with the aim to detect a concise set of metamodel constructs to support the mapping out of transformation rules within a conceptual modelling framework.*

**KEYWORDS:** *Semantic interoperability, Semantic Transformation, Conceptual Data Modelling, INTERLIS, UML, GML*

## INTRODUCTION

As the number of different data formats in the geo-information domain is still rising, the need for standardization has been taken into account and different specifications have recently been developed especially by the OpenGIS consortium to improve interoperability between different (proprietary) geographic information systems. Besides the abstract specifications most of these standards address the logical or physical modelling level. This is of course a tremendous improvement towards interoperable systems but the following aspects are missing:

- Modelling techniques on a conceptual and therefore system independent level. The demand for a human readable visual notation like UML and an exact formalism to provide automated processing of models provided by domain experts is undoubted.
- Means to setup rules on a conceptual level to achieve semantic transformation between different models or different shapes of a modelled domain.

Such a framework leads to the automatic generation of interoperable facilities within a GI service architecture:

- deduction of data formats or schema descriptions (like XML-Schema, GML)
- data checking services
- data transformation
- generation of logical data models (SQL or proprietary DDL implementations)
- generation of graphical representations of geo-data
- automatic deduction of service description and communication schemas (WSDL, SOAP)

## BASIC MODELLING FRAMEWORKS

The following sections provide a short introduction to the relevant foundations which are needed as a basis for the developed concepts on semantic transformation on a conceptual level:

**INTERLIS 2**

During the last two years the second version of INTERLIS, an exact formal conceptual schema language, was developed to integrate concepts of object oriented design. The architecture is now based on following constructs:

- inheritance and polymorphism
- association and aggregation
- metamodels and –objects
- modelling facilities for visualisation of geo-data
- use of UML as graphic representation of data models

As a normative part of the INTERLIS-Standard the transfer service was also reengineered and is now based on the W3C XML recommendation. Encoding rules permit the automatic deduction of an XML-Schema out of a user-designed data model and therefore implements one of the conceivable derivations of service descriptions on the logical or system-dependent level. Other key features of the renewed transfer service are the so-called incremental subsequent delivery and the enabled polymorphic data exchange. The latter allows an import interface to read data belonging to an extended model without knowing anything about the model extensions.

**UML-Profile**

Communication between domain experts, data modellers and implementers often is difficult when using a formal textual description language. The Unified Modeling Language as a graphical tool for modelling purposes is well established within the software industry and is well suited to improve the design process and communication between involved parties when creating GI-service architectures.

The UML specification also provides a framework which allows metaclasses from existing metamodels of UML-core to be extended to tailor the UML-metamodel to different domains. This mechanism was used to create a profile which supports the features of the INTERLIS modelling language. As a first result, a software tool called UML/INTERLIS-Editor was implemented. It provides export of textual model descriptions according to the INTERLIS specification from a graphical, user designed UML model.

## REQUIREMENTS - DRAFT PROPOSALS

One of the main investigation topics presented in this paper is the analysis and creation of patterns which are well suited for the GIS-domain and provide a comprehensive set of transformation rules allowing the user to formulate intelligible data and transformation models:

As the conceptual schema language INTERLIS is based on object oriented design principles the transformation facilities have to be founded on object algebra which has to be aligned and extended with respect to the spatial characteristics of the schema language.

Besides the constructs descending from relational algebra (like POJECT, JOIN, UNION, INTERSECT etc.) there were introduced some facilities to open up the potential of object orientation.

**Transformation of Geometric Data Types**

The underlying abstract metamodel of geometric data types is constituted by a hierarchical architecture. Therefore access to subcomponents of geometric objects (like edges of a polygon) is achieved by using unnesting operators in a similar way as with other encapsulated types.

The following example shows an unnesting operation and serves the edges of a 'SURFACE'-type attribute named 'Geometry' within the class 'LandCover':

```
INSPECTION OF LandCover -> Geometry;
```

**Integration Alternatives**

Two approaches have to be considered when dealing with mapping rules within a modelling language:

View mechanism: According to current database systems the definition of views based on available modelling constructs like classes or associations leads to the desired results. One of the emerging problems with this proceeding is the absence of an independent conceptual model for transformed data ('executed' views). Normally, a client is not interested in a model including all mapping rules and dependencies to the original models. Therefore, this approach requires the derivation of such self-contained models out of view definitions.

Independent mapping constructs: Another attempt consists of applying mapping instructions which are beyond the scope of the target data structures. Such a transformation relies on one or several input models and results in the generation of the content for an independent output model. As the INTERLIS 2 specification already implements a view mechanism this approach requires the introduction of several constructs in the metamodel of the conceptual schema language.

## IMPLEMENTATION – PRACTICAL EXAMPLES

A formal description of the metamodel for a transformation framework is a main goal of this research. This paper is not the place for publication of such a specification. Instead we will present some constructs by means of a few examples carried out with view-mechanism:

```
TOPIC topic_A =
 CLASS Class_A =
  AttrA: TEXT*20;
  GeometryA: SURFACE
 END Class_A;
END topic_A;
TOPIC topic_B =
 CLASS Class_B =
  AttrB: TEXT*20;
  GeometryB:
 END Class_B;
END topic_B;
VIEW TOPIC topic_View =
 DEPENDS ON topic_A, topic_B;
 VIEW Class_View
  ...
 END Class_View;
END topic_View;
```

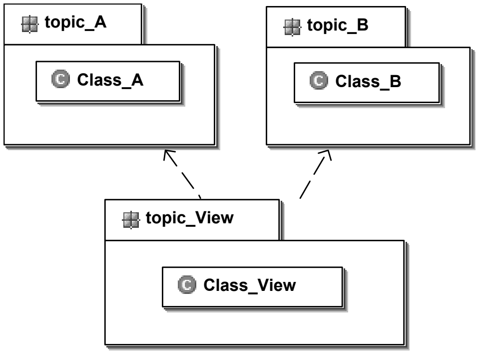


*Figure 1:* Example data model in UML and deducted formal textual notation showing the package/topic dependencies.

Based on the Model introduced before, the following example shows a simple join operation:

```
 VIEW Class_View_2
JOIN OF A ~ topic_A.Class_A, B ~ topic_B.Class_B;
WHERE A->AttrA == B->AttrB;
= ATTRIBUTE
  ALL OF A;
  AttrAB : TEXT*40 := B->AttrB;
 END Class_View_2;
```

## USE-CASES

### Data Transformation

Starting from a dataset available in a proprietary format, the following processing scheme may be applied:

1. Preparation of a conceptual model which represents the logical input format as far as possible
2. Conversion of the input data to the INTERLIS transfer format which corresponds to the prepared model. Existing software tools (1:1 processors) may be applied to handle this step.
3. Setting up a model extension which depends on a certain number of models created just before. Within this extension, the transformation classes reflecting the desired output model have to be built.
4. Transformation of the underlying data by means of software that is able to interpret the formation rules presented within this paper.
5. If needed: 1:1 processing the transformed data into a desired proprietary output format.

Due to the modularity of this application flow the adaptation to a variety of input/output formats is quite simple by providing appropriate 1:1 processors.

### Metadata Extraction

The use of views is quite useful to provide a set of metadata out of an existing geodata model. This method allows creating a distinct data extract using the “Projection” view. Result of every view is a set of view objects that can be integrated into a given metadata model. The parameters of the view need to be set according the metadata model structure to guarantee a proper data integration:

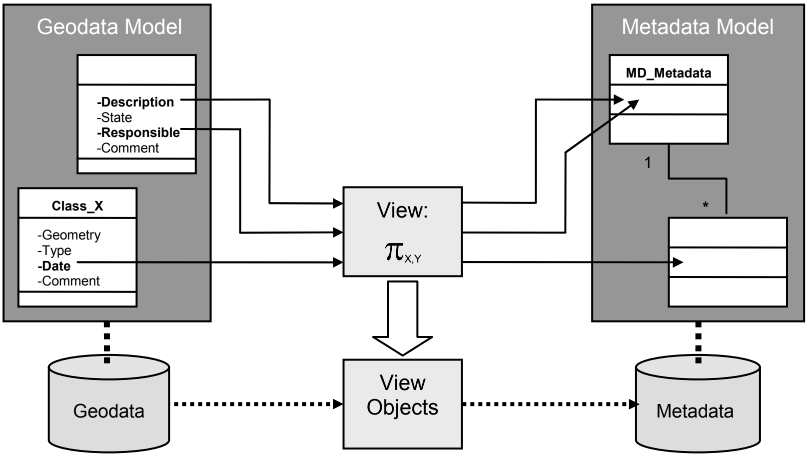


***Figure 2:*** Mapping process using views: Metadata extraction

**Data Visualisation**

To visualize objects out of a dataset it is often more comfortable to rearrange and consolidate information from different associations especially if the data model is very fine grained. The generated graphics may be based on transformed model elements.

Another aspect is the need for transformed geometric properties before the visualizing process. For example imagine a class with an attribute of type 'AREA' (area-wide surface topology). If we want to draw the borders of these objects we have to provide a set of boundary lines to the symbolizer. This can be achieved by applying a un-nest-operation to the geometric attribute which serves a list of the underlying constructs of the data type which are the edges of the polygons.

## OUTLOOK

A variety of applications based on a framework that enforces the definition of transformation rules on a conceptual level to achieve a mapping between different semantics of domain models are imaginable. In the following subsection we will give an overview on different projects carried out at our institute and of which we think they are of interest for the geo-information community:

**Schema Language Extension: 3D-Datatypes**

Subject of a PhD. thesis being carried out by one of the authors is the integration of data types for three dimensional objects. The diversity of currently used logical and physical models for 3D data makes high demands on the finally established modelling elements.

Main focus will be put on data structures that build up 3D objects following the principles of an enhancement (Pfund M., 2001) of the Formal Data Structure (FDS, Molenaar 1990). With the possibility to get access to all substructure of such a data type the introduction of mapping rules to extract necessary information to build up objects of less complexity like surface models used within the CAD domain will be feasible.

**Ontology-based Transformation Approach**

The coupling of detailed domain ontologies with a repository of corresponding application schemas provides an information base to be consumed and evaluated by an inference machine. The mapping of generated conclusions to possible transformation rules is subject to a future research topic dealing with automated model conversion.

## BIBLIOGRAPHY


COGIS - Swiss GIS Coordination Group: INTERLIS 2 – reference manual (German version). www.interlis.ch, 2003.

Molenaar M., 1990, A formal data structure for 3D vector maps, in: Proceedings of EGIS'90, Vol.2, Amsterdam, The Netherlands, pp. 770-781

Object Management Group (OMG): Formal Unified Modeling Language (UML) Specification, Version 2.0, final adopted specification, www.omg.org/uml, 2003

OpenGIS Consortium: Geography Markup Language (GML), Implementation Specification, www.opengis.org, January 2003.

Pfund M., 2001, Topologic data structure for a 3D GIS, Proceedings of ISPRS, Vol. 34, Part 2W2, 23-25 May, Bangkok, Thailand, pp. 233-237

World Wide Web Consortium (W3C): Extensible Markup Language (XML) 1.0 Second Edition.W3C Recommendation, October 2000.