Semantic Interoperability through the Definition of Conceptual Model Transformations

Hans Rudolf Gnägi, Andreas Morf, Peter Staub

Institute of Geodesy and Photogrammetry, GIS and Theory of Errors Group
Swiss Federal Institute of Technology, Zurich
ETH Hönggerberg, CH-8093 Zurich, Switzerland

{ gnaegi | morf | staub } @geod.baug.ethz.ch

SUMMARY

To solve the currently urgent problem of collecting critical data across system and country borders without modification of any source system, we present a model-based 4step procedure. Its core is a new language for "horizontal" mapping of conceptual models, which is introduced by a small but multi-face example. It is further shown that this PIM-PIM mapping of platform independent models needs to be completed by an appropriate "vertical" PIM-PSM mapping between platform independent and platform specific models from the conceptual level to the (physical) format level. Together with geometric operations this combination of two model mapping types provides an effective semantic interoperability. Our work is based on practical experiences with the application of the MDA in Switzerland and a short review of these is given as introduction. Finally, the outlook sketches the near future with the integration of our 4step procedure into web-services providing e.g. a transformation enabled model driven WFS.

KEYWORDS: *Conceptual Modelling, Model Mapping, Model Transformation, Interoperability, Model Driven Architecture*

INTRODUCTION

The INSPIRE project shows the need for collecting environmental, risk and catastrophe data across different system and country borders to get an overview on the European level and/or alarms on the level of neighboured countries. It is impossible: to modify all the nationally grown different data structures according to a new European one, and is impossible: to solve all the necessary data transformations by programming interfaces at the format level for every pair of interacting systems. This corresponds to the following general situation: A given real-world selection can be modelled in different ways and different data structures can administrate the same information content. How can the corresponding data be brought together?

We propose a four-fold procedure as an answer to this question:

(i) Lift the data structure description to the system independent conceptual level.
(ii) On the conceptual level, add the possibility to define mappings between different data structures of the same content.
(iii) Provide rules for an automatic deduction of exchange formats (and basics for other services) form the conceptual description of data structures.
(iv) Implement generic tools performing the automatic transformation of the data corresponding to the conceptual structure description from (i) given the mapping of (ii).

We have implemented and tested this procedure in a practical bottom up approach – introduced by Morf & Staub, 2004 – and are now integrating it stepwise in an object oriented top down frame. Recent solutions for steps (ii) and (iii) will be the main topic.

We start with a clearly restricted but implemented and working bottom up approach to the four-fold procedure by looking at INTERLIS experiences in Switzerland. Then we show, how step (i) has to fit into the object oriented principles for data structuring and software developing given by UML 2.0 and the different abstraction levels from reality selection to data. With this basis, we look at the general possibilities for data structure transformations and we will show that steps (ii) and (iii) can be seen as two aspects of the same theoretical back-ground. A short section will give an overview of the state of implementations of generic model driven soft-ware to perform data refor-matting according to model mapping of step (iv).

Whereas we first focus on the data level, we show in a following section how steps (i) through (iv) fit into the world of ontologies. Finally, conclusions and future work follow in the last section.

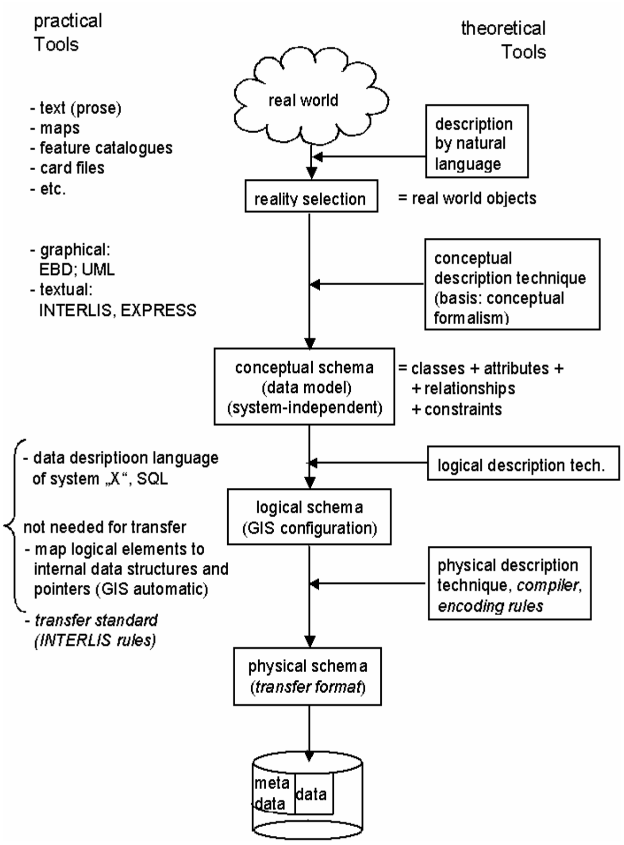

Figure 1: Data modelling, from reality to data using four levels

BOTTOM UP REALISATION: INTERLIS EXPERIENCES

In 1982, Switzerland started the revision of the official surveying with the aim to replace the drawn map as legal reference by digital data. Because of the extreme federalism in Switzerland (consisting of 26 independent states called "Cantons") there existed at that time at least 26 different systems treating official surveying data and it was not possible to select one of these as official system and to force 25 cantons to replace their systems by the official one.

Decision 1 was: Any system is allowed but it has to be able to accept and to produce the official survey data in a special format, the Swiss official surveying interface (*amtliche Vermessungs-schnittstelle [AVS]*)

Decision 2 was: Not to define the format on the format level but to describe the corresponding data structure on the conceptual level as a conceptual model and to define rules for an automatic calculation of the format description out of the conceptual model.

Figure 1 shows the developed model driven approach using four abstraction levels from reality to stored data:

- For the description of the reality selection a natural language was used.
- For the description of the conceptual model (i.e. to realise step (i) of the procedure mentioned in the introduction), the entity-block-diagram was applied and a textual conceptual scheme language (CSL) INTERLIS has been developed and applied.

- The logical level can be skipped if a transfer format is needed (see figure 1). The logical model (or schema) has to be provided if a database or a GIS shall be configured based on the conceptual model.
- At the physical level a proprietary transfer format has been invented (INTERLIS transfer format ITF) and the simple description of it can automatically be derived from the conceptual schema by rules defined in chapter 3 of Swiss standard SN612030 (SNV, 1999) This completed step (iii) of the procedure in this introduction.

Based on the CSL INTERLIS 1 (SNV, 1999), a set of system-independent tools has been developed since 1990. According to the steps mentioned in the introduction these are:

(i) An *INTERLIS compiler* checks the syntax of a conceptual schema (freeware and open source).
(ii) Different existing conversion tools allow creating model-mappings but only on the physical level. The conceptual schema is present but merely to understand what has to be defined on the format level. One advantage of these conversion tools is a big variety of useful transformation functions, especially spatial operations and filters.
(iii) The INTERLIS compiler produces the description of the corresponding transfer format according to an error-free conceptual schema (the physical, format-, or encoding schema, respectively). This encoding schema can as well be produced for XML – transferring all INTERLIS possibilities – and for GML.
(iv) Tools supporting step (ii) include the automatic transformation of the corresponding data in standard format. As an application, the aforementioned tools provide transformations between INTERLIS format such as XML or GML and most of the existing proprietary standard formats.

OBJECT ORIENTED DATA MODELLING (STEP I)

With the version 2.0 of the Unified Modelling Language UML a comprehensive and consistent definition of the object oriented methodology has been provided by the Object Management Group (OMG) (OMG, 2001, 2002, 2003, 2004). From the aim to change the formalism of INTERLIS from (object) relational to object oriented by the new version of INTERLIS 2, follows that the INTERLIS 2 metamodel has to be designed as a profile of the UML metamodel. We give here only a short report of the actual state.

There is an obvious equivalence between the UML and INTERLIS 2 language elements for class, attribute and association. The INTERLIS 2 elements MODEL and TOPIC correspond to the UML element package. The possibility to define attribute types by the DOMAIN construct is equivalent to the DATATYPE specification possibility of UML. The possibility to introduce UNITS in INTERLIS 2 does not explicitly exist in UML but units can nevertheless be introduced as attributes of DATATYPEs. Constraints dealing only with “usual” attributes of non-geometric data types can immediately be mapped to OCL-constraints of UML. But constraints with geometry type attributes have to use basic operations on the classes of the corresponding geometric objects, which needs a (small) adaptation of the syntax for INTERLIS 2 functions.

Geometric data types themselves are not included in the UML definition. The international standard ISO19107 “Geographic information – Spatial schema” (ISO, 2001) provides a comprehensive overview on all geometric data types existing in GIS as extensive inheritance hierarchies. INTERLIS 2 on the other side has condensed these inheritance hierarchies into data types, which can be used as attribute types and need not to be included by inheritance. To find out which “equivalence type” exists between these two different concepts, is in working process. The necessity to reduce the huge set of geometric data types in ISO19107 is obvious. Most systems are not able to implement all these types. Therefore interoperability is only guaranteed, is standard profiles of ISO19107 are defined as it is done by the standard ISO19137 “Geographic information – Core profile of spatial schema” (ISO, 2005).

MODEL MAPPING (STEP II)

The experiences with establishing a geo-data model-mapping and the corresponding data transformation using existing tools showed that simple attribute mappings or type conversions are not sufficient. A set of “elementary” class transformations is needed. These class transformations include: copy, split, merge classes and generate associations. To explain some problems with model mapping, which we found, and the corresponding solutions, which we propose, let us have a detailed look at an example of using the functionality “generate association” in a model mapping.

Spatial objects may have an implicit geometric relationship with each other, e.g. a house meets one parcel (or possibly more than one). It doesn’t make sense to model spatial relations conceptually. Nevertheless, it can be transformed into an explicit object oriented association between two classes in a target model as the following example in figures 2 and 3 shows. Given objects of class `Building` can meet one or more objects of class `Parcel`, because both are of geometry type `GM_Surface`. We call this fact a “virtual association” between these two classes. The target data model on the right side of figure 3 describes two (non-spatial) classes `Realty` and `House` which are related by the “real” object oriented association shown in Figure 2.

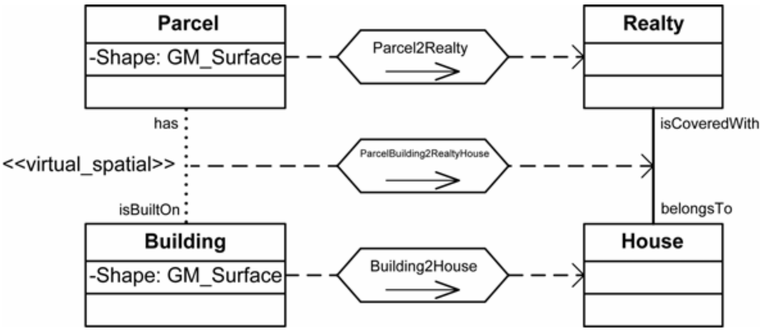

Figure 2: Model mapping including a virtual association

```
MAPPING MODEL RealEstate =
 TRANSFORMATION GROUP Group1 =
  SOURCE { Parcel, Building
   { VIRTUAL ASSOCIATION ParcelBuilding =
     has -- Parcel; // Role-Def 1
     isBuiltOn -- Building; // Role-Def 2
     WHERE intersects(Building.Shape, Parcel.Shape);
    END ParcelBuilding; } }
  TARGET { Realty, House, ASSOCIATION RealtyHouse }
  MAPPING {
   RULE Parcel2Realty { Realty := Parcel; }
   RULE Building2House { House := Building; }
   RULE ParcelBuilding2RealtyHouse {
    RealtyHouse.isCoveredWith := ParcelBuilding.has;
    RealtyHouse.belongsTo := ParcelBuilding.isBuiltOn; } }
END RealEstate;
```

Figure 3: Mapping language applied to the example of Figure 2.

The mapping language which we propose provides the text of Figure 3 for the model mapping presented graphically in Figure 2. A mapping model is subdivided in transformation groups. In each of these groups first the source model is described. In `{ }` the classes and associations which need some transformation before they can be included in the target model have to be indicated. In the same way, the model elements resulting from the mapping have to be expressed in `{ }` for the target model after the key word `TARGET`. After the `RULE` name only a necessary minimum of transformation statements is given in `{ }`. In the example depicted in Figure 3, the `RULE Parcel2Realty` supposes that with the expression `Realty := Parcel`, the mapping system recognises by automatically comparing source and target data model that it has not to transfer the geometry from the source class `Parcel` to the target class `Realty`. In order to find the limits of automatic attribute (or general model element) mapping between conceptual schemas the results of Brodeur, 2003, have to be taken into account.

The importance of effective and efficient geometric functions is indicated by the definition of the virtual association in Figure 3. It has to be calculated, which surface object of class `Parcel` has common inner points with which surface object of class `Building`. If we consider complex computing tasks of this type, the main advantage of the declarative concept we chose for the mapping language instead of a procedural one becomes obvious. Optimized procedures can be implemented using e.g. in our intersection calculation algorithms of the plane sweep type.

The `RULE ParcelBuilding2RealtyHouse` shows how elegant a geometrically evaluated virtual association can be used to generate a normal association.

The example of Figures 2 and 3 finally shows that model mapping is not always reversible. From the geometry of `Parcel` and `Building` the association between `Realty` and `House` can be calculated. But the no longer existing geometry of `Realty` and `House` cannot be reconstructed from the association between the two classes.

So, the mapping of figures 2 and 3 is a mapping between two models on the conceptual level. In IT terminology this is called *PIM-PIM* mapping between platform independent models.

AUTOMATIC GENERATION OF FORMAT DESCRIPTION (STEP III)

If we look at the transition from the conceptual level to the logical and/or physical one (figure 1), we have again a model mapping, but now of another type. These are *PIM-PSM* mappings between a platform independent and a platform specific model. Whereas in the case of PIM-PIM (or “horizontal”) mapping the model changes, but the actual CSL of the models is the same, i.e. both models use the same language elements or, in a more general way, the metamodel of the CSLs of both models is the same. In the case of PIM-PSM (or “vertical”) mapping we look at the same reality selection from different abstraction levels, the model remains in principle the same. The description language changes, namely from a CSL to a *LSL* (logical schema language) or to a *PSL* or *FSL* (physical or format schema language). That means that the metamodel of the description language changes. A vertical mapping can take place as well between two different PSMs, so can be a PSM_1-PSM_2 mapping. In fact, vertical mappings between two levels are done for all reality selections and their conceptual models by one mapping of the metamodels of the description languages of the two levels.

Full semantic interoperability is not achieved without combining these two concepts of model mapping. The model mapping method explained in the last paragraph for phase (ii) has been applied until now only for the “horizontal” mapping of conceptual models but not yet for the “vertical” mapping of metamodels. A practical diagram that combines both horizontal conceptual structure mapping and vertical implementation mapping is given in Figure 4. There ① denotes the vertical (i.e. PIM-PSM) model mapping and ② denotes the encoding of the data objects according to the format description automatically provided by ①. PIM-PSM mapping is not done at the moment by our mapping language of step (ii) between the corresponding metamodels, but can be achieved simply by providing encoding rules for the given CSL (INTERLIS 2).

It is important to notice that a system implementing a PSM (resulting by vertical mapping ① from a PIM) needs encoders and decoders to build up the data objects according to the data format derived by ①.

In addition, Figure 4 shows that – at the moment – we do not intend to get the mapping rules between the PSMs on the logical and/or physical level by applying the vertical mapping ① to the model mapping defined on the conceptual level. The transformation of data objects as instances of the PSM structures is directly derived from the model mapping on the conceptual level

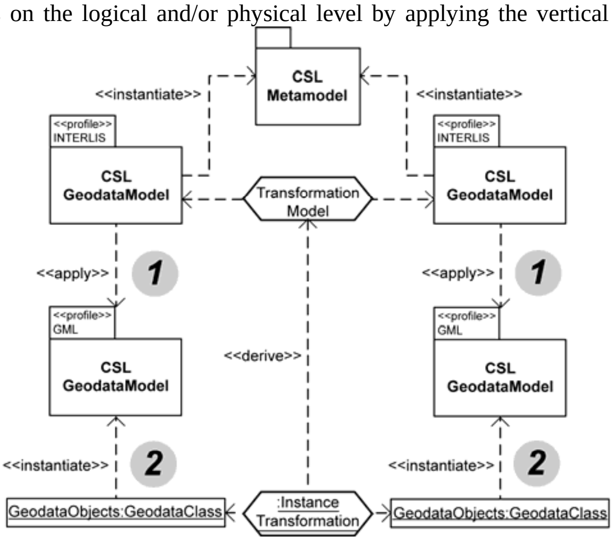

Figure 4: Actual horizontal and vertical model mapping and encoding combined

In Figure 4, we use the symbolism of *UMLX* introduced by Willink, 2003, for schematic display of the model mappings and object transformations. The OMG MOF (Meta Object Facilities OMG, 2003, 2004) is the language used to describe metamodels.

The combination of mappings PIM-PIM, PIM-PSM and PSM-PSM (as a data transformation) provides, together with the integration of independent or class-related functionality, a methodology that enables effective semantic interoperability in the field of geodata. An adequate separation of a conceptual schema is proposed by Olivé, 2004: the Domain Conceptual Schema (DCS) and the Functionality Specification (FS). Due to the preceding analysis, we focus on the DCS to establish semantic interoperability.

GENERIC MODEL DRIVEN TRANSFORMATION SOFTWARE (STEP IV)

The feasibility of the theoretical work is currently being verified by implementing prototype software modules:

Metamodel infrastructure for geodata models

An appropriate management of data about models is crucial for a transformation between these models. As our work is based on the UML modelling language the according UML metamodel is used as a common platform for storing data describing models. The exact specification of a profile of the UML version 2 metamodel for geodata-modelling is a part of this step and will eventually provide the metamodel of the object-oriented textual conceptual schema language INTERLIS. The modelling elements provided by INTERLIS have been proved to be useful and appropriate for practical needs by the resulting models of different large applications (Kaufmann et al. 1993, Balanche et al. 2001, Bruhin et al. 1998, 2005). An INTERLIS-compiler already exists to parse textual conceptual models. In addition, its INTERLIS-UML2 version will fill the data about correct models into the metamodel profile. As a result, metadata about models are available in XMI-format and may therefore be stored and queried by an XML-database.

Transformation-Language metamodel

The intended transformation engine consists of a parser for transformation instructions. An example of such instructions is given in Figure 3. The metamodel for storing these instructions is implemented according to the Query/View/Transformation approach of the OMG (OMG, 2006).

Transformation of Geodata

The intrinsic transformation of objects finally is a matter of transcription of the transformation metamodel to an actual software-platform like a GIS or a database. As a prototypical proof of concept on the PSM-Level we decided to use SQL3 with spatial extensions.

POSITION OF OUR WORK IN THE ONTOLOGY SPECTRUM

After having described the method of model mapping for conceptually modelled data in order to establish a semantic interoperability framework, let us have a look at the possibility and the consequence of positioning the conceptual data modelling within the ontology spectrum described by Obrst, 2006. An “ontology” defines the terms used to describe and represent an area of knowledge (subject matter) or reality selection (see figure 1). An ontology also is the (semantic) model (set of concepts) of these terms. The “ontology spectrum” describes a range of semantic models (i.e. ontologies) of increasing expressiveness and complexity: taxonomy, thesaurus, conceptual model, logical theory.

Our experiences confirm the classification of conceptual modelling in the ontology spectrum proposed by Obrst. Usually a basic set of predefined object properties – the “data types” – is selected to describe the attributes of (geo) data structures. Together with data types describing the properties, the other elements of a conceptual schema language like class, association etc. allow to describe structure, relations, constraints and rules of the concepts represented by a data model. Conceptual models correspond to the ontology definition given above. Clearly they are ontologies of much more expressiveness than taxonomies or thesauri. By a conceptual model the basic properties can be used to build more complex data structures and data types. Therefore, one has to be aware of the generation of a set of attributes which may be re-used for the description of geometric data types out of classes and properties in the Web Ontology Language (OWL) (WWW Consortium, 2004, 2004a).

Geographic data are in fact compatible with the concept of ontologies. Winter, 2001 asks if ontologies are merely a temporary fashion or if a paradigm shift in the GI science occurred thereby. He states that the work with ontologies exceeds conceptual modelling. According to our experience we can say, that conceptual models are ontologies. A precise analysis of these similarity aspects is introduced by Olivé, 2004. Machine processing is possible and different automatic services can be based upon. But they need human interpretation as we have seen in step (iii) of defining semantic mappings on the conceptual level. Therefore, conceptual models are only “weak” ontologies according to the ontology spectrum. Additional machine reasoning and the use of inference engines is provided by logical theories, which therefore are classified as “strong” ontologies. In addition, it seems that the only way to provide exact enough descriptions of reality selections being as well system independent as machine treatable consists in modelling conceptually. The adaptation of “non-geographic” weak ontologies (like OWL-ontologies) or their conceptual schema languages (like OWL), respectively, to geographic information is therefore an important step towards a truly semantic interoperability.

Comparing the four layer concept of data modelling with the ontology spectrum, we see that the lowest level of expressiveness can be used on the first level, where we need to describe the relevant selection of the real world using “natural language”. To clarify the natural language describing the reality selection, a weak taxonomy can be used with arbitrary subclassification relation and without generalisation/specialisation (example: a folder/directory structure), or a strong taxonomy with consistent semantics for generalisation/specialisation (e.g.: Linnés biological taxonomy), or even a thesaurus of terms with the four term-semantic relationships between synonyms, homonyms, broader/narrower-than terms and associated terms (example: controlled vocabulary). To add a taxonomy or a thesaurus to the natural language used, allows to structure the content of a future data model in a "language neutral" way. This facilitates communication about reality selections across language borders practically all over the world. Proceeding to the conceptual level, we need more than a hierarchical taxonomy or a thesaurus of terms because here we need in addition to define properties (attributes and values), relations, constraints and rules about concepts referenced by terms. Therefore weak ontologies (conceptual models) and finally strong ontologies (logical theories) are needed.

"Spatialising" OWL, a non-geometric CSL for weak ontologies

Let's have a look at the OWL metamodel illustrated in Figure 5. In extension of W3C's Resource Description Framework (RDF) and RDF Schema (RDFS), all OWL language elements are specified. An ontology given in OWL represents a static structure in a similar way as a geodata model given in an arbitrary other conceptual schema language. We describe the metamodel (i.e. the language elements of OWL) by means of Meta Object Facility (MOF) classes and MOF associations (OMG, 2002a). Like Chang & Kendall, 2004, we use URI references to denominate the elements. In this way the RDF schema acts as a meta language that defines both itself and OWL. An OWL class selects a set of instances (called *individuals*) from a collection based on their common characteristics. Property aspects that declare the relationship between the OWL classes respectively their individuals come along with OWL classes. The properties define the static framework discussed above, that characterises the ontology. For instance the `owl:intersectionOf` an OWL class instance with another OWL class instance provides the possibility to describe the above mentioned intersection functionality of geometric objects, if these geometric objects, i.e. subsets of a Euclidean plane or 3-dimensional space, are treated as classes of point individuals. An additional construct (that is only provided by OWL DL and OWL Full, as the `owl:intersectionOf` – see WWWConsortium, 2004) can be used to describe geometric functionalities, namely `owl:unionOf`. The definition of geometric properties based on the geometric interpretation of these two constructs allows OWL also to provide weak ontologies for reality selections with spatial aspects. The topological relationships like *disjoint*, *meets*, *covers* and *overlaps* according to the *4/9i Models* by Egenhofer, 1993 and Egenhofer, Sharma & Mark, 1993, can easily be introduced as consistency constraints using the two OWL-constructs introduced above. The case of spatial fuzzy set operations is not concerned at the

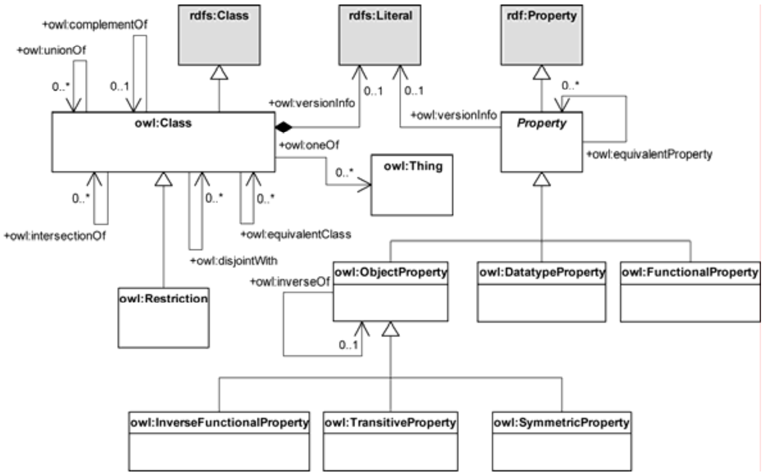

Figure 5: OWL Metamodel

moment. Thus, the fundamentals are given to extend the static attribute value structure of OWL ontologies with geometric properties.

FUTURE WORK

Transformation-Enabled Geoweb Service: Model-Driven WFS

A Web Feature Service (WFS) (OGC, 2005), which implements the current specifications of the OGC provides geodata of an underlying database in GML-encoded form according to an associated application schema (W3C XML schema). From the client's point of view, these data and corresponding models are static. If necessary, he is responsible for a conversion, if other kinds of data models and data formats are needed on the target system. A service framework including the modelling and transformation possibilities presented in the chapters before may be called "Transformation Enabled Model-Driven WFS (mdWFS)". It is able to transform the requested data into the desired target format before delivering, if mapping instructions are given. The following workflow arises for a service user:

- Service provides a catalogue of available models and corresponding data
- Client creates a target model as well as a transformation model. This model describes the mapping from the service model to the desired target model
- These Models are transmitted to the service for processing
- Target model and corresponding, transformed data can be obtained by the client

The realisation and prototype implementation of the preceding concept including corresponding supplemental proposals to the OGC WFS specifications is currently a cooperation project with the Technical University of Munich (TUM).

Eclipse GMT

The Generative Model Transformer project of the Eclipse initiative1 will provide a basic infrastructure for the implementation of the Model Driven Approach (MDA) of the OMG. Atlas Transformation Language (ATL) (Eclipse, 2005) is a promising candidate model transformation language, whereas still no public specification is available (see above). It is also still unclear, what role ATL will play in relationship with regards to a final "MOF QVT" OMG standard. To establish a prototype implementation of all presented concepts, GMT libraries are used as a base upon which extensions to handle spatial aspects are built. Particularly software modules performing model based data transformations are developed. These modules allow the transformation of hierarchically built geometric data types. Whereas the aim of the GMT is primarily to optimise model based software and application development respectively, we intend to contribute extensions to the transformation language that allow the handling of large spatial data sets. On the other hand, the slightly different focus of GMT certainly provides valuable input about applying common patterns from software engineering to geo-information systems.

Integration of geometry into OWL

The conditions allowing the introduction of 2- or 3-dimensional Euclidean point sets as OWL-classes to use the OWL (meta) associations `owl:intersectionOf` and `owl:unionOf` for the definition of geometric properties have to be clarified thoroughly.

CONCLUSIONS

In this paper, we introduced how knowledge about the interrelations between models can be described on conceptual level *and* in a machine processable form. Therefore, a system independent transformation description language on the metamodel basis of a conceptual schema language has been presented. Besides pure modelling, the formal description of the interrelations between data models is a big step towards semantic interoperability in a first phase. According to the ontology section, this furthermore makes up the potential to connect UML facilities used for geodata modelling with ontology-based or driven approaches, respectively.

REFERENCES

Balanche, R. et al.: Swiss Official Surveying Interface, Object-oriented Conceptual Schema. Version 24d, Eidg. Vermessungsdirektion, 2005

Brodeur, J.: Interopérabilité des Données Géospatiales: Elaboration du Concept de Proximité Géosémantique. PhD Thesis, Déparement des Sciences Géomatiques, Faculté de Forestière et de Géomatique, Université Laval, Québec, Canada, 2003

Bruhin, G. et al.: Leitungskataster, Object-relational Conceptual Schemas for Gas, Water, Waste Water, Distant Heating, Electricity and Telecommunication. Swiss standard SIA 405 and technical sheets #2015, #2016, 1998.

Bruhin, G. et al.: Leitungskataster, Object-oriented Conceptual Schemas for Gas, Water, Waste Water, Distant Heating, Electricity and Telecommunication. Swiss standard SIA 405 and technical sheets #2015, #2016, Revision 2005.

Chang, D.T., Kendall, E.: Metamodels for RDF and OWL. Proceedings of the 1st international Workshop on the Model-Driven Semantic Web, Monterey, USA, 2004

Eclipse Generative Model Transformer Project: ATLAS Transformation Language Sub-Project. http://www.eclipse.org/gmt/, 2005

Egenhofer, M.J.: A Model for Detailed Binary Topological Relationships. Geomatica, 47(3&4)m 261-273, 1993

Egenhofer, M.J., Sharma, J., Mark, D.M.: A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis. Autocarto 11, R. McMaster and M. Armstrong (eds.), Minneapolis, USA, 1993

ISO International Standards Organisation: ISO19107 Geographic information – Spatial Schema, 2001

ISO International Standards Organisation: ISO19137 Geographic information – Core Profile of the Spatial Schema, 2005

Kaufmann, J. et al.: Swiss Official Surveying Interface, Object-relational Conceptual Schema. Technical Ordnance of Swiss Official Surveying, Annex B, Eidg. Vermessungsdirektion, 1993

Morf, A., Staub, P.: Semantic Transformation by Means of Conceptual Modelling Techniques. Proceedings of the 7th Conference on GI Science (AGILE), Heraklion, Greece, 2004

Obrst, L.: The Ontology Spectrum & Semantic Models. MITRE Information Semantics Group, Information Discovery & Understanding, Center for Innovative Computing & Informatics, January 12 & 19, 2006

OGC Open Geospatial Consortium: Web Feature Service Implementation Specification.OGC Specification No. 04-094, 2005

Olivé, A.: On the Role of Conceptual Schemas in Information System Development. Ada-Europe 2004, LNCS 3036, 2004

OMG Object Management Group: Unified Modelling Language: Action Semantics. OMG Revised Final Submission ad/01-08-06, 2001.

OMG Object Management Group: XML Metadata Interchange v.2.0. OMG Specification formal/02-01-01, 2002

OMG Object Management Group: Meta Object Facility v.1.4. OMG Specification ptc/02-04-03, 2002a

OMG Object Management Group: Unified Modellig Language: Infrastructure. OMG Specification ptc/03-09-15, 2003

OMG Object Management Group: Unified Modelling Language: Superstructure. OMG Specification ptc/04-10-02, 2004

OMG Object Management Group: Query/View/Transformation, RFP (ad/2002-04-10), QVT-Merge Group, version 2.0, ad/2005-03-02, 2005

SNV Swiss Standards Organisation: SN612030 INTERLIS (1) Modelling Language and Data Transfer Method, http://www.interlis.ch, 1999

Willink, E.: UMLX: A Graphical Transformation Language for MDA. Eclipse GMT Consortium (`http://www.eclipse.org/gmt`), 2003

Winter, S.: Ontology: Buzzword or Paradigm Shift in GI Sciences? International Journal of Geographic Science, 15(7), 587-590, 2001

WWW Consortium: OWL Web Ontology Language Reference. W3C Recommendation, http://www.w3.org/TR/owl-ref/, 2004

WWW Consortium: OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation, http://www.w3.org/TR/owl-semantics/, 2004a