Schema Translations in a Web Service Based SDI

Lassi Lehto
Finnish Geodetic Institute
Department of Geoinformatics and Cartography
PO Box 15, FI-02431 Masala, Finland
Lassi.Lehto@fgi.fi

INTRODUCTION

Emerging network service -based spatial data infrastructures (SDI) are creating new challenges for spatial data providers. One of them is the ever-increasing number of different application areas in which digital geospatial data are being used, together with the diverse requirements these set for the structure and terminology used in the data content.

The Web is undergoing a significant transformation from a decentralized document repository to a platform for distributed applications. This new approach to distributed computing is widely known as Service-Oriented Architecture (SOA). Web applications are also becoming more sophisticated with the introduction of the AJAX computing model (Garret, 2005). The increased local computing power available in Rich Internet Applications (RIA) (Wikipedia, 2007) enable tighter integration of geospatial content with other thematic content on the data level – a task much more complicated than the plain visual overlay of map images. All this calls for a flexible way to satisfy the various demands regarding the data models the geospatial content is served in.

Harmonization of data models among geospatial data providers is a common interoperability goal with a long and rather unsuccessful history. Various organizations have developed their geospatial data structures over the years, carefully tailoring them to provide the best possible support for the core business processes and requirements internal to the organization. Outside demand for data model harmonization, based on common goals and causes, is frequently met with stubborn resistance – for good reasons. It can even be argued that an organization should not change its internal data model on the basis of outside requirements, as these external demands are inherently diverse, possibly contradictory, and change over time.

How then, could a geospatial data provider satisfy these varying external requirements without losing the benefits that a carefully designed data model offers for the internal processes of the organization? A possible answer lies in the principle of achieving data harmonization only on the service level. This implies a significant role for on-the-fly data transformations. A real time transformation can be organized as an additional conversion layer on top of the content service, or transformations could be carried out by a separate Web service, external to the data provider's environment.

At least two basic transformation types are important in the context of geospatial content: coordinate transformation and data model, or schema translation. While coordinate transformation is a well-understood and widely supported process, schema translation is a much more obscure concept and its semantics are poorly defined. Even the name is misleading. A schema translation process actually transforms data content expressed in one schema into data content expressed in another schema – it does not transform the schema itself. The term 'schema mapping' refers to the process of determining the correspondence between the data items in the source and target schemas prior to the actual schema translation process. This mapping can be greatly facilitated by ontology-based semantic matching of related concepts (Visser, 2001).

In this paper the concept of schema translation is discussed in the context of geo-spatial Web services. First, the data transformation requirements set by the envisaged INSPIRE service infrastructure are introduced. The following section identifies the basic schema translation process components and discusses alternatives related to the process flow. The section 'Schema Translation as a Service' approaches schema translation as a Web service concentrating especially on service capabilities and parameterization. Some generic service architecture issues are dealt with in the section 'Service Architecture'. Three practical implementations of the schema translation service are presented in the section 'CASE studies'. The paper ends with a conclusion and some prospects for future work.

RELATED RESEARCH

Integration of heterogeneous data sources through schema translation is a well-established field in computer science. It has a long tradition in various applications, such as federated databases, data warehousing and data mining (Marotta, 1999). However, most of the related studies only consider schema translation in the context of relational database technology, and rarely take into account specific issues related to spatial data.

A specific field of research is schema translation between two modeling languages. The most discussed processes in this area are the translations between UML and XML models and translations between a relational database schema and the corresponding XML content model (Lee, 2003). These studies also include some examples from the domain of geospatial problems (Snowflake, 2004) (Portele, 2005).

Schema translation services have been specifically studied in the context of metadata. These studies relate for instance to Web-based digital libraries (Godby, 2003) or to bibliographic references (Llavador, 2006). In the latter example a three-step procedure for schema translation has been developed: 1. Automatic extraction of concepts from the source and target schema, 2. User-guided mapping of the concepts, 3. Automatic generation of the corresponding transformation script in the form of an XSLT file. The three mapping types identified are: direct mapping for semantically equivalent concepts, function-based mapping and constant value mapping.

A good example of a schema transformation service in the domain of geospatial applications is provided by the geodetic control point service of the Wisconsin Land Information Clearinghouse (WiscLINC, 2006). The service architecture includes a schema mediator process capable of two-way schema translations. In addition to the standard XSLT process, the research project also investigated a streaming variation of the XML-to-XML transformation process, called Streaming Transformations for XML (STX) (Becker, 2003).

Donaubauer et al provides a good introduction to the topic of schema translation services (Donaubauer, 2006). The paper proposes an approach based on Model Driven Architecture (MDA) in which the schema mapping is defined at the conceptual level, preferably aided by an interactive graphical tool. The schema translation rules will be automatically derived from this conceptual-level mapping. Future work within the project involves developing a prototype model-driven Web Feature Service (mdWFS). The prototype service will implement a schema translation process guided by user-defined schema mapping.

Misund and Valerhaugen present research in which datasets encoded in GML according to heterogeneous application schemas are integrated together via schema analysis and mapping (Misund, 2004). The fundamental properties common to GML application schemas are utilized during the XSLT-based schema analysis process. Subsequent schema translations are based on the schema mapping information obtained. The final proof-of-concept result of the project is presented as a generic schema-independent GML viewer.

INSPIRE REQUIREMENTS FOR DATA TRANSFORMATIONS

Most of the geospatial network service categories listed in the EU's recently approved INSPIRE (Infrastructure for Spatial Information in Europe) Directive (European Parliament and the Council, 2007) are rather well defined and already supported by existing service standards and software solutions. These include a geospatial data discovery service, view service and download service. More obscure is the definition of the fourth service category: a transformation service. According to the Directive, this service is supposed to be combined with other service types to make them compatible with the common European-level data specifications being developed in the INSPIRE process.

In the context of geospatial applications, data transformation can be identified as a special type of service able to transform data from its original form into another form, often to achieve conformance with certain common specifications. The need for this kind of service definitely exists, because of the two basic INSPIRE principles: a) Pan-European spatial data is supposed to be available via INSPIRE services as a seamless, harmonized geospatial content, b) these services are supposed to rely mostly on currently existing data sources, without significant new data collection or re-engineering of already established databases.

In the INSPIRE process, transformation services are seen as a generic means to make the other service types compliant with common European-wide specifications. Thus, in addition to the geospatial data content transformations, one also has to consider two other types of transformations: map style -related transformations and metadata content translations. However, the following discussion concentrates entirely on issues related to schema translation applied to geospatial data.

SCHEMA TRANSLATION

Translation Types

The data translation process could be generally discussed at three different levels, i.e. as a syntactic, schematic or semantic translation. In translation on the syntax level, the main task is to change the language of the data representation; generally this also implies change of the related schema definition language. An example of this kind of process would be a translation from a relational database structure, defined in the data definition language of the database, into an XML representation, defined in XML Schema (W3C, 2004). As OGC-defined Web services widely use XML-encodings, a reasonable presumption in the case of the envisioned Schema Translation Service is restriction of the input and output representations into XML syntax. In the case of geospatial content this means Geography Markup Language (GML) Application Schemas (OGC, 2004). Consequently, the role of syntactic translations in the Web service environment is regarded as minor.

A schematic translation means modifying the structure and the schema vocabulary of the data model used in the input dataset. Generally there is a clear, well-defined one-to-one, one-to-many or many-to-one mapping from constructs in the input schema to the corresponding entities in the target schema. This translation category is well suited to the current Web environment, and would be feasible to implement as an independent Web service. The source and target schemas could easily be indicated by referring to the respective XML Schema documents. The main thing missing is a standardized way to indicate the schema mapping, i.e. how schema constructs in the source schema relate to constructs in the target schema.

Semantic translations involve cases where exact mapping from source to target schema cannot be defined. Such translation would require in-depth analysis to determine the semantic match between entities in the related schemas. The translations may also involve significant changes in the way the spatial information is represented. In many cases exact correspondence between the source and target data entities simply does not exist, and a reasonable approximation has to be used instead. Semantic translations require an ontology-based approach and presumably cannot be fully automated.

Schema Translation Components

Schematic translation is the most relevant translation category that can be implemented as a Web service in the current situation. In principle, complex translation process could always be defined as a set of atomic translation steps. In the case of schema translation, a set of basic translation components could also be identified. In the following a categorization of the schema translation process components is presented in the context of a standards-based Web-service environment. The translation process is intended to transform data from one GML application schema into another. The number of possible transformation components is greatly reduced, as the same fundamental modeling concepts are applied on both sides of the process. The main conceptual levels of the GML feature model are presented in Figure 1.

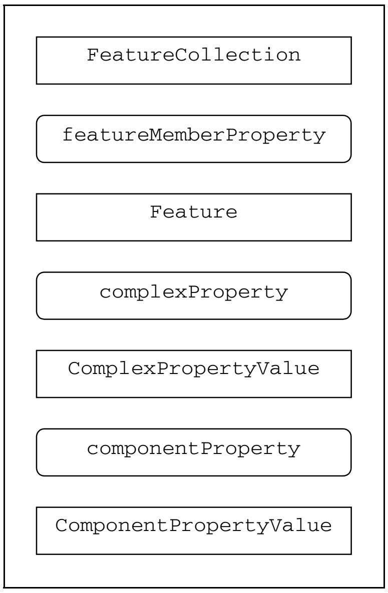

Figure 1: Conceptual levels of the GML feature model. The rectangles represent object-type levels and the rounded rectangles property-type levels.

The schema translation components identified are categorized in the following list. Each main translation type is further divided into subcategories, based on the role the component has on the different levels of the GML feature model.

1. Filtering

1.1. Filtering of `FeatureCollections` (only some of the source `FeatureCollections` are represented in the target schema)

1.2. Filtering of `Features` (only some of the `Features` in the source `FeatureCollection` are translated to the target `FeatureCollection`, typically basing the selection on the value of a given property; because the `featureMemberProperty` and the `Feature` itself form an inseparable unit, the `Feature` works as the criterion for filtering of the `featureMemberProperty`)

1.3. Filtering of feature `properties` (only some of the source feature `properties` are translated to the target feature; concerns both complex and simple `properties`)

2. Renaming

2.1. Renaming of `FeatureCollections` and `Features`, their `properties` and property values (e.g. translation between two natural languages)

3. Reclassification

3.1. Reclassification of `FeatureCollections`

3.1.1. Converging (based on a property value, `FeatureCollections` from two or more source `FeatureCollection` types are translated to `FeatureCollections` in a single target `FeatureCollection` type)

3.1.2. Diverging (based on a property value, `FeatureCollections` from a single source `FeatureCollection` type are translated to `FeatureCollections` in two or more different target `FeatureCollection` types)

3.2. Reclassification of `Features`

3.2.1. Converging (based on a property value, `Features` in two or more source `FeatureCollections` are translated to `Features` in a single target `FeatureCollection`)

3.2.2. Diverging (based on a property value, `Features` in a single source `FeatureCollection` are translated to `Features` in two or more different target `FeatureCollections`)

3.3. Reclassification of property values

3.3.1. Converging (a coarser classification system in the target property value domain)

4. Merge / Split

4.1. Merging or splitting `FeatureCollections`

4.1.1. Merging (`Features` in two or more source `FeatureCollections` are translated to a single target `FeatureCollection`; requires instance-level matching)

4.1.2. Splitting (`Features` in a single source `FeatureCollection` are translated to `Features` in two or more different target `FeatureCollections`)

4.2. Merging or splitting `Features`

4.2.1. Merging (properties in two or more source `Features` are translated to properties of a single target `Feature`; requires instance-level matching)

4.2.2. Splitting (properties of a single source `Feature` are translated to properties in two or more target `Features`)

4.3. Merging or splitting complex `property`

4.3.1. Merging (two or more complex `property` values in the source `Feature` are translated to a single complex `property` value in the target `Feature`)

4.3.2. Splitting (a single complex `property` value in the source `Feature` is translated to two or more complex (or single) `property` values in the target `Feature`)

4.4. Merging or splitting simple `property` values

4.4.1. Merging (two or more simple `property` values in the source `Feature` are translated to a single simple `property` value in the target `Feature`; requires specific knowledge of the method (concatenation, sum etc.))

4.4.2. Splitting (a single simple `property` value in the source `Feature` is translated to two or more simple `property` values; requires specific knowledge of the method (break on space, division etc.)

5. Reordering

5.1. Changing the order of `featureMemberProperties` inside a `FeatureCollection`

5.2. Changing the order of properties inside a `Feature`

5.3. Changing the order of component `properties` inside a complex `property`

6. Value conversions

6.1. Spatial conversion

6.1.1. Simplification

Surface to linear outline conversion
Surface to point conversion

6.2. Unit of measure (UOM) conversions of `property` values (e.g. converting miles to kilometers)

7. Morphing

7.1. Morphing down (translating a `FeatureCollection` to a `Feature` or a `Feature` to a complex `property` value)

7.2. Morphing up (translating a complex property value to a `Feature` or a `Feature` to a `FeatureCollection`)

8. Augmentation

8.1. Adding derived spatial properties

8.2. Deriving values for target schema properties missing in source schema – based for instance on the values of other properties existing in source schema

8.3. Filling in default property values in target schema

In the following table the identified schema translation components are shown in the context of the GML content model. The rows in the table correspond to the different conceptual levels of the GML feature model, going from the root `FeatureCollection` via the Feature level to the value of a complex property and finally to the value of an individual complex property component. The corresponding property name levels are shown interleaving these GML object types.

The columns represent the main schema translation component types. The order of the columns from left to right indicate the most reasonable sequence for carrying out these translation components in an individual transformation process. The numbers in the cells refer to the corresponding subcategory in the listing above.

Operation Level	Filt.	Ren.	Recl.	M/S	Reor.	Conv.	Mor.	Aug.
FeatureCollection	1.1	2.1	3.1.1 3.1.2	4.1.1 4.1.2			7.1	
featureMemberProperty	1.2	2.1			5.1			
Feature	1.2 [crit.]	2.1	3.2.1 3.2.2	4.2.1 4.2.2			7.1 7.2	
complexProperty	1.3	2.1			5.2			
ComplexPropertyValue	1.3 [crit.]	2.1		4.3.1 4.3.2		6.1	7.1	8.1
componentProperty	1.3	2.1			5.3			
ComponentPropertyValue	1.3 [crit.]	2.1	3.3.1	4.4.1 4.4.2		6.2		8.2 8.3

Table 1: Applicability of the schema translation components in relation to the levels of the GML feature model. [crit.] indicates that the corresponding level only works as a criterion for processing at the parent level. Numbers refer to the listing above.

SCHEMA TRANSLATION AS A SERVICE

Schema Translation Service Capabilities

When considering schema translation as a Web service it is important to define clearly the functional capabilities a given service can offer. The aspects that could be taken into account include support for two-way translations, pre-configurability, capabilities regarding spatial transformations and support for external thesauri.

A basic property of a schema translation service is whether it works as a one-way or a two-way translation. One-way translation can only process data content. A two-way translation service is able to transform both data queries and the resulting datasets. In the layered service architecture a schema translation service must be able to run both query and data translations (Figure 2 A), whereas in the case of chain-based service orchestration this depends on how the service interface is specified (Figure 2 B).

Another important consideration is whether the service only works in a real-time mode or whether it can be configured beforehand. In the case of real-time service, the schema mapping definition is given to the service as a part of the request to perform the translation. A configurable service allows translation definitions to be stored in the service before the actual processing request (configuration time). In the translation request only a reference to a preconfigured process definition is given, together with the dataset to be transformed (run time). Substantial performance and robustness benefits can be achieved with the pre-configurable service approach.

A relevant capability issue is the level of sophistication the service offers regarding spatial processes in the translation. One aspect is the ability of the service to carry out spatial analysis operations on the source dataset to match data structures present in the target schema. As a translation definition might also include processes that transform geometry objects from one geometry type into another (surface to point, surface to line etc.), the ability of the service to carry out such spatial type conversions is another important capability consideration.

When one of the most obvious translation tasks – translation between two different natural languages – is considered, the support for thesaurus-based term mappings becomes crucial. The ability of the service to consult an external thesaurus instead of a fixed setting in the schema mapping adds considerably to the service’s flexibility.

Parameterization of the Schema Translation Service

When schema translation of geospatial data as a Web service is discussed, an important question arises concerning parameterization of the service. How can a calling application direct the service to run the translation the way that is desirable considering the intended usage?

The source and target schemas can be easily identified by a means of a URI reference to the corresponding XML Schema document, thus enabling the process to validate the input and output datasets against a well-defined schema. A more difficult task is to instruct the service about the schema mapping.

There are examples of applications relying on sophisticated graphical user interfaces that let the user define schema mappings in an interactive, software-guided manner. However, these solutions are tightly integrated into the translation process and do not follow any standards in the way the mapping is conveyed to the process.

XSLT as a Schema Translation Language

In the present Web processing environment a standard mechanism has been defined for XML data transformations, called Extensible Stylesheet Language Transformations (XSLT) (W3C, 2006). When designed as a set of template rules, an XSLT-based process can be applied well to schema translation of geospatial data. Other relevant transformations where the same mechanisms can be applied include generalization (Lehto, 2005) and multi-purpose publishing of geospatial data (Lehto, 2003).

Being a declarative process, XSLT transformation adapts flexibly to varying schema constructs in the input dataset. As a well-defined and widely recognized specification, XSLT is supported by various software implementations and is also readily available as free, open source tools.

IN considering the schema translation components listed above, it can be noted that XSLT readily supports many of the operations identified. The following simple XSLT instruction code snippet includes seven translations components (1.2, 1.3, 2.1, 3.2.1, 3.3.1, 5.2, 8.3).

```
<xsl:template match=″in:Parcel[area &lt; 2000] | in:Lot″>
   <out:CadastralUnit>
      <out:areaClass>
         <xsl:choose>
            <xsl:when test=″in:area &lt; 1000″>A1</xsl:when>
            <xsl:otherwise>A2</xsl:otherwise>
         </xsl:choose>
      </out:areaClass>
      <out:id_number>
         <xsl:value-of select=″in:idNro″/>
      </out:id_number>
      <out:status>registered</out:status>
   </out:CadastralUnit>
</xsl:template>
```

A geospatial schema translation service can be envisaged as one that is parameterized by XML Schema document-based source and target data model definitions and XSLT document-based schema mapping instructions. Further work would be needed to introduce spatial concepts into the XSLT rule set by standardizing them via the XSLT extension mechanism, applying either extension elements or extension functions.

SERVICE ARCHITECTURE

There are two fundamentally different approaches on how a schema translation service could be understood. Firstly, a translation service can be seen as an individual layer in a stack of service layers. Secondly, schema translation can be regarded as a link in a chain of services. The two architectures are depicted in Figure 2.

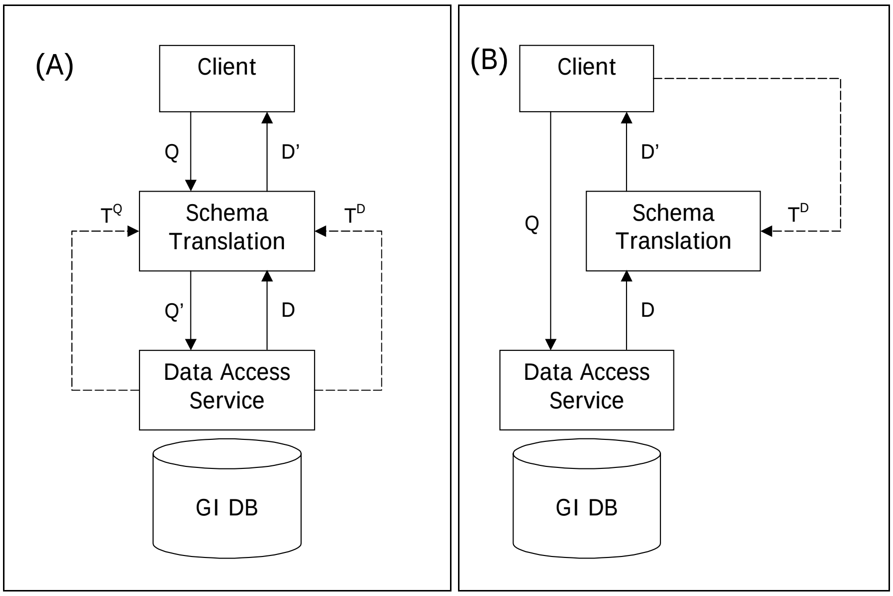

Figure 2: (A) A schema translation service as a layer in a stack. Schema translation is a two-way process; both query and data translations (TQ, TD) are controlled by the server. (B) A schema translation service as a link in a chain. Schema translation is a one-way process controlled by the client (TD).

Layered Service Architecture

In the case of services as layers in a stack, each layer makes use of the service provided for it by the service layer below and in turn provides services to the layer above. The stack is preconfigured and rather static. In this approach a schema translation service could be seen as a front end to the actual data content service, completely hiding the background service from the calling client application or service layer. In this case the schema translation service is fully controlled by the content provider and the service interface is actually presented as a conventional data access interface. The pros and cons of this approach can be listed as follows.

Pros

- The solution is easy for the calling application; it does not need to care about the details related to the translation process.
- The solution is robust and works well in cases where the target data model is pre-defined and rather stable.

Cons

- There is no way for the client application to fine-tune the outcome of the translation.
- The data provider must tailor the translation to the various client needs; consequently, it must be familiar with each of the data models used on the client side.

Chained Service Architecture

In this approach the translation service is seen as a link in a chain of individual services. This architecture enables more flexible service configuration. The chain of services can be set up in an ad hoc manner, and thus the best possible combination of services can be chosen for the particular task in hand. The interface of a schema translation service is in this case seen as a type of data processing interface. The pros and cons of the approach include the following.

Pros

- As the original content service is visible to the calling application, it can be freely plugged into the best translation service available.
- Being in control of the translation process, the client application can flexibly configure the process to fit the end result to its needs.
- As the main responsibility for configuring the translation is with the calling application, the content provider does not need to cater for the varying needs of different clients.

Cons

- The approach is more demanding for the client-side application.
- Developing a dynamic, user-controllable schema translation process is a demanding task.

A Schema Translation Service as a Component of SOA

In the Service-Oriented Architecture (SOA) approach computing applications are supposed to be built by integrating a set of independent network services (He, 2003). One of the main principles of SOA is that a service does not need to know anything about the calling application (apart what is conveyed by the service request parameters) and that the calling process does not need to know anything about the internal details of the service. A well-defined service interface is the only fixed thing between the actors. Both sides must naturally have common understanding of the semantics related to the operations the service interface supports.

When schema translation service is considered as a component service in an SOA, the basic SOA principles that have to be taken into account include the following: loose coupling, reusability and composability.

The principle of loose coupling means that there should be no tight connections and interdependencies among the services. Regarding the two general service architectures outlined above, the tight integration existing between the data service and the schema translation service in the layered approach (Case A) clearly infringes this requirement. In the service chain approach (Case B), the connection between the data service and translation service is created by the calling application in an ad hoc manner.

Reusability refers to the requirement for a service to be applicable in many different service configurations. In the case of the service chaining approach, the transformation service can easily be configured to support varying requirements as the client is in control of the translation process.

According to the principle of composability, it should be easy to combine services with other services to form new computing processes. This principle emphasizes the requirement for a well-defined service interface that could be used as a contract between services when they are combined dynamically.

Service Interface

As the discussion above indicates, the service chain approach yields more benefits than the more static layered service stack setup. This is especially true in dynamic service environments. One of the main issues in the service chain -based process is the fact that it requires a well-defined service interface for schema translation to be really pluggable in a flexible way. So far no standard interface specification for a schema translation service has been defined by any standardization body.

The OGC has defined a couple of service interface specifications that have some relevance in the context of the envisaged Schema Translation Service. All the aspects that are common to every service type have been collected into a single base document: OpenGIS Web Services Common Specification (OWS) (OGC, 2005a). This specification deals with a service metadata query (`GetCapabilities`) and service exceptions. It also defines some widely used data types, such as bounding boxes, coordinate system references, etc. All the concrete service interface specifications are supposed to make use of the definitions in OWS.

An example of a concrete service specification related to data transformations is the OGC's draft Web Coordinate Transformation Service Implementation Specification (WCTS) (OGC, 2005b). WCTS defines a service capable of transforming a given dataset from one coordinate reference system into another. The source and target coordinate reference systems, and the dataset to be processed are indicated in the service request. The supported coordinate systems and data formats can be requested using a service metadata query.

There are some similarities between the WCTS and the envisaged schema transformation service. Both processes have well-defined source and target systems and the input dataset is transformed between the two. For the process to be possible, clearly specified mapping from the source system to the target system must exists and be available for the transformation engine. The two services are so close to each other that coordinate transformation could easily be run as an integral part of a schema translation process.

The Web Processing Service (WPS) draft specification (OGC, 2005c) defines a generic interface for a service that is able to perform some specific processing of geospatial data. In addition to the metadata query inherited from the OWS interface, the WPS specification declares two new operations: `DescribeProcess` and `Execute`. The `DescribeProcess` operation provides detailed metadata about the nature of the processes the service offers. The `Execute` request is used to carry out the process. The specification also provides mechanisms for indicating the structure and location of the input and output datasets, which can represent either literal or complex types. The WPS interface could well be applied as a base standard in the design of an interface for a schema translation service.

CASE STUDIES

In connection with the research described in this paper, three different experimental implementations of a schema translation service have been developed. All three services are based on the service stack approach. The first service was developed as part of the GiMoDig project (Sarjakoski, 2005) aiming to integrate topographic map data from four different countries in a cross-border use case scenario. The second was implemented as a data integration service between three individual municipalities in the Helsinki metropolitan area in Finland. The third service prototype was developed in connection with the INSPIRE process to test schema translations between two different road data models.

The GiMoDig project concentrated on use cases related to mobile cross-border applications. The architectural approach in the project relied on direct database access, on-the-fly schema translation and real-time generalization. A specific service layer, called a Data Integration layer, was responsible for the schema translations between the agreed common schema and the national data models of the four participating countries: Denmark, Finland, Germany and Sweden (Lehto, 2004a). The common schema developed as one of the major deliverables of the project contains 17 important content themes represented on the national topographic maps at a scale of 1:20000. The required coordinate transformations were also incorporated into the translation process.

The schema translation process in the GiMoDig project is based on the XSLT -mechanism. The transformations are preconfigured and static. As the prototype setup is based on the service stack approach, the schema translation is a two-way process. The queries first have to be translated from the jointly agreed common schema into the local schema, and the resulting datasets are subsequently translated from a national schema to the common schema. As the Web Feature Service (WFS) access interface was used in the Data Integration service, both the queries and the resulting data content were encoded in XML. This greatly facilitates the translation process, as the same base mechanism (XSLT) can be applied both ways (Lehto, 2004b).

The service platform developed in the GiMoDig project has also been applied on a national level, in a similar cross-border setting between individual municipalities. The three participating cities, Helsinki, Espoo and Vantaa, together form the Helsinki metropolitan area. The surveying departments in all three cities maintain basically identical base-map data content for city planning purposes; however, each city models the information in an individual way. An experimental data integration service was developed as a means to facilitate development of the cross-border service infrastructure in the Helsinki area.

The common data model applied in the Helsinki case study has been developed by the Finnish Association of Local and Regional Authorities as a national recommendation on municipality-level base-map information. This common schema models features with much richer topological structures than are available in the datasets provided by the cities in the Helsinki test case. Consequently, the schema translation service in the prototype must significantly enhance the topology of the source datasets. Sophisticated functions have been integrated into the schema translation process for this purpose. They include, for instance, a process to spatially match individual features and combine them into more complex logical units, as required by the target schema. As such, the service implementation demonstrates the role a schema translation service might play in improving the topological quality of the source data.

In the third case study, the same base mechanisms were applied to carry out a schema translation from the Finnish national digital road database, Digiroad, to the European level standardized road-data model developed by the EuroroadS project (Wikström, 2006). The successfully implemented test service demonstrates the important role a schema translation process can play in a theme-specific Pan-European service infrastructure like INSPIRE.

CONCLUSIONS

Geospatial data resources are becoming an important background information source for many new application areas. Combining disparate data sources together as an innovative information product is a prevailing trend in current Web development. Together with the introduction of increasingly sophisticated Web application frameworks, i.e. Rich Internet Applications, these developments require data producers to provide better support for varying data models in their Web services.

Schema translation is emerging as an indispensable process type in a service infrastructure aiming to satisfy varying application needs based on a single constant data resource. This is exemplified particularly well by the main goal of the INSPIRE process: to build up a seamless, harmonized, European-wide geospatial data service infrastructure largely relying on existing database resources.

When implemented as a Web service, schema translation can be regarded as a data-access interface layered on top of the actual data source providing geospatial content in its original schema. In a rather static environment this architecture can be implemented in a robust manner and the process can be fine-tuned for optimal performance. However, the data provider must manage all the translations required to support diverse client needs, which can easily become a burden. Layered schema translation also necessitates support for a two-way processing model, as both the query and the resulting dataset must be translated. The experience gained in the practical test implementations indicates that it frequently becomes impossible to carry out the translation of a complicated query sentence.

A schema translation service has to be defined appropriately if it is going to be introduced as a service category of its own in the emerging SOA-based computing environment. To be flexibly orchestrated and plugged into a dynamically formed service chain, schema translation must be seen as a type of independent geospatial processing service with a well-defined access interface. Standardized mechanisms for parameterization of the service have to be found. The introduced categorization of identifiable schema translation components will help in specifying unambiguous service capability statements and in formulating the requirements for a schema mapping language.

The mechanisms for parameterization of a schema translation service could be based on common Web techniques, such as XML Schema for source and target schema definition and XSLT, augmented with appropriate spatial extensions, for schema mapping instructions. Further work is required to validate the introduced schema translation component categorization and to test its full applicability using these techniques.

BIBLIOGRAPHY

Becker O., Brown P., Cimprich P., 2003, An Introduction to Streaming Transformations for XML. O'Reilly XML.com, URL: http://www.xml.com/pub/a/2003/02/26/stx.html

European Parliament and the Council, 2007, Directive of the European Parliament and of the Council establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). PE-CONS 3685/06, Brussels.

Donaubauer A., Fichtinger A., Schilcher M., Straub F., 2006, Model Driven Approach for Accessing Distributed Spatial Data Using Web Services - Demonstrated for Cross-Border GIS Applications. Proceeding of the XXIII International FIG Congress, Munich, Germany, October 8-13, 2006.

Garret J. J., 2005, AJAX: A New Approach to Web Applications, URL: http://www.adaptivepath.com/publications/essays/archives/000385.php

Godby C.J., Smith D., Childress, E., 2003, Two Paths to Interoperable Metadata, In: Dublin Core Conference, DC-2003: Supporting Communities of Discourse and Practice Metadata Research & Applications, Seattle,Washington (USA).

He H., 2003, What is Service-Oriented Architecture. O'Reilly XML .com, URL: http://www.xml.com/pub/a/ws/2003/09/30/soa.html

Lee D., Mani M., Chu, W., 2003, Schema Conversion Methods between XML and Relational Models. Knowledge Transformation for the Semantic Web, IOS Press, Amsterdam.

Lehto, L., 2003, A Standards-Based Architecture for Multi-purpose Publishing of Geodata on the Web. In: Peterson, M.P. (ed.), Maps and the Internet, Elsevier Science, pp. 221-230.

Lehto L., Sarjakoski, T., Hvas, A., Hollander, P., Ruotsalainen, R. Illert A., 2004a, A Prototype Cross-border GML Data Service. Proceedings of the 7th AGILE Conference on Geographic Information Science, Heraklion, Greece, April 29 - May 1, 2004, pp. 361-367.

Lehto L., Sarjakoski L.T.: Real-time generalization of XML-encoded spatial data for the Web and mobile devices. International Journal of Geographical Information Science, 19(8-9) (2005) 957-973.

Lehto, L., Sarjakoski T., 2004b, Schema Translations by XSLT for GML-Encoded Geospatial Data in Heterogeneous Web-Service Environment. Proceedings of the XXth ISPRS Congress, July 12-23, 2004, Istanbul, Turkey, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXV(B4:IV), pp. 177-182.

Llavador M., Canos J.H., 2006, XSMapper: a Service-Oriented Utility for XML Schema Transformation. ERCIM News, No. 64, 2006.

Marotta A., Ruggia R., 1999, Designing Data Warehouses through schema transformation primitives. Proceedings of the 18th International Conference on Conceptual Modeling (ER'99), Paris, France, November, 15-18, 1999.

Misund G., Valerhaugen H., 2004, Integration of Heterogeneous GML Sources. In Online Proceedings of GML Days 2004, Vancouver, Canada.

Open Geospatial Consortium, 2004, OpenGIS Geography Markup Language (GML) Implementation Specification, URL: http://portal.opengeospatial.org/files/?artifact_id=4700

OGC, 2005a, Open Geospatial Consortium: OpenGIS Web Service Common Implementation Specification, URL: https://portal.opengeospatial.org/files/?artifact_id=8798

OGC, 2005b, Open Geospatial Consortium: Web Coordinate Transformation Service (WCTS) draft Implementation Specification, URL: https://portal.opengeospatial.org/files/?artifact_id=8847

OGC, 2005c, Open Geospatial Consortium: Web Processing Service (WPS) Specification, URL: http://portal.opengeospatial.org/files/?artifact_id=8723&version=2

Portele C., 2005, OGC Web Services (OWS) 3 UGAS Tool, Open GIS Discussion Paper.

Sarjakoski T., Sarjakoski L.T., 2005, The GiMoDig public final report. URL: http://gimodig.fgi.fi/deliverables.php

Snowflake, 2004, Schema Translation between Relational and GML Schemas. Snowflake White-paper.

Visser, U., Stuckenschmidt, H., Wache, H., & Vögele, T., 2001, Using Environmental Information Efficiently: Sharing Data and Knowledge from Heterogeneous Sources. In C. Rautenstrauch & S. Patig (Eds.), Environmental Information Systems in Industry and Public Administration, Hershey, USA & London, UK: IDEA Group, pp. 41-73.

Wikipedia, 2007, Rich Internet application, URL: http://en.wikipedia.org/wiki/Rich_Internet_application

Wikström L., 2006, Road Network Information Model, Deliverable 6.3, EuroroadS project, URL: http://www.euroroads.org/php/Reports/D6.3 Road Network Information Model.pdf

WiscLINC, 2006, Wisconsin Land Information Clearinghouse: Schema Transformation. URL: http://www.sco.wisc.edu/wisclinc/transform.php

W3C, 2004, World Wide Web Consortium: XML Schema Part 0: Primer Second Edition, URL: http://www.w3.org/TR/xmlschema-0/

W3C, 2006, World Wide Web Consortium: XSL Transformations (XSLT) Version 2.0, URL: http://www.w3.org/TR/xslt20