

## 1 Abstract

Part of the CIPI 1.2 (Critical Infrastructure Protection Initiative phase 1 stage 2) initiative run by the OpenGIS consortium developed prototype implementations of Translating Web Feature Servers (WFS-X). These feature servers served geographical data in a single common GML application schema with the source data being provided from a number of sources each with a different GML application schema. The implementations were therefore required to translate the data from the various source schemas into the common target schema.

The Snowflake implementation built on components from an existing product, GO Loader. GO Loader enables the translation of GML2 application schema into Oracle database schema.

Setting up translations is labour intensive. The process requires that a person understand the meaning of the data in both schemas and devise a translation between the two. Detailed meta-data is essential to this process. An absence of meta-data makes translation impossible. The informal way in which translations were specified resulted in differences in interpretation of the data and schemas in the different implementations.

GO Publisher is a product currently under development at Snowflake. It is based on the prototype developed in the CIPI 1.2 project and will bring that functionality to industrial strength by Q3 2004. The first release of GO Publisher will support the syntactic approach to translation piloted in CIPI 1.2.

Whilst syntactic translation can provide working solutions to the problem of schema translation it is limited by two factors: The reliability of the translation process is subject to the ambiguities in interpretation of the semantics of the source and target schemas; The cost of creating syntactic translations is high since each translation must be “hand made” by a human who has learned the semantics of both source and target schemas.

A semantic approach to translation would provide a means to address both of these issues.

A semantic approach would involve creating a formal statement of the semantics associated with the source and target schemas. Formal statements can then be made about the relationships between the semantics of each model. If these statements are sufficiently complete then it will be possible to automatically derive the syntactic translation from the source to the target schema from the statement of semantics.

This approach makes the relationship between the source and target models explicit instead of being implied through a syntactic relationship. This reduces the risk of errors arising from misinterpretation of the models.

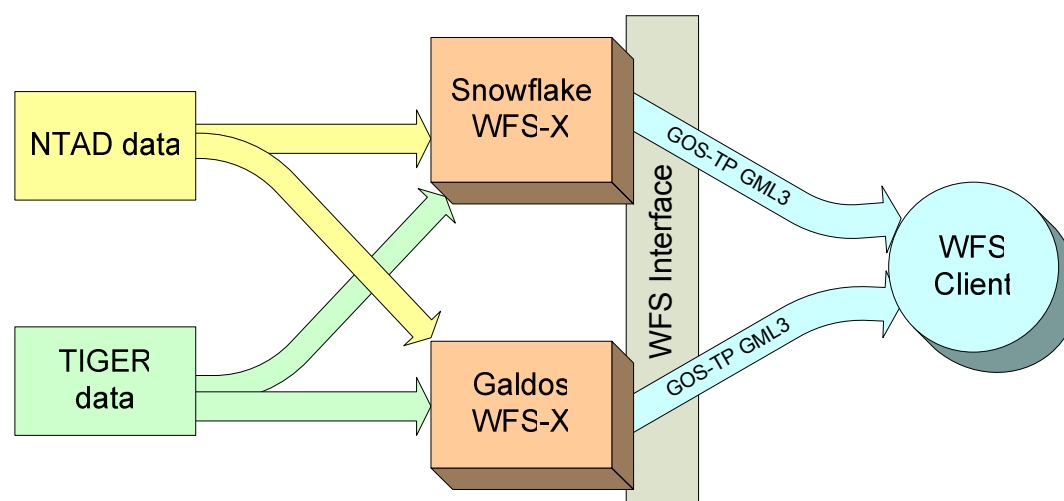
Automating the creation of the syntactic translation means that the investment of time and effort put into translation is spent on the semantics rather than the syntax. The definition of the semantics is not tied to one schema or technology and is therefore more widely applicable. It can be used to create, for example, a translation between two GML schemas, between two relational database schemas, or between a relational database schema and a GML schema.

## 2 Background

### 2.1 The CIPI 1.2 WFS-X Test Bed

Part of the CIPI 1.2 (Critical Infrastructure Protection Initiative phase 1 stage 2) initiative run by the OpenGIS consortium developed prototype implementations of Translating Web Feature Servers (WFS-X). The aim of the exercise was to demonstrate the exchange of information from heterogeneous data stores.

The data stores each had a different data model (schema) and were implemented using different database technologies and the data used for the exercise was supplied in a variety of formats and schema. The aim was to eliminate the differences in source data schema and data store schema in the data provided by the servers to the client application. The client application should therefore have been able to transparently switch between servers and data from different sources without any change of schema.



**Figure 1 - Components of the CIPI 1.2 test bed**

Communication between the data stores and client applications was carried out through an interface conforming to the OpenGIS Web Feature Server (WFS) standard. The transport format for the data was Geography Markup Language (GML) 3.0. Road data was used for the exercise and data was sourced from the US Census TIGER database and from the US NTAD (National Transportation Atlas Database). The common schema used to communicate between the servers and client was the GOS-TP (Geospatial One-Stop – Transport Pilot)-Roads schema developed during a previous OpenGIS test bed.

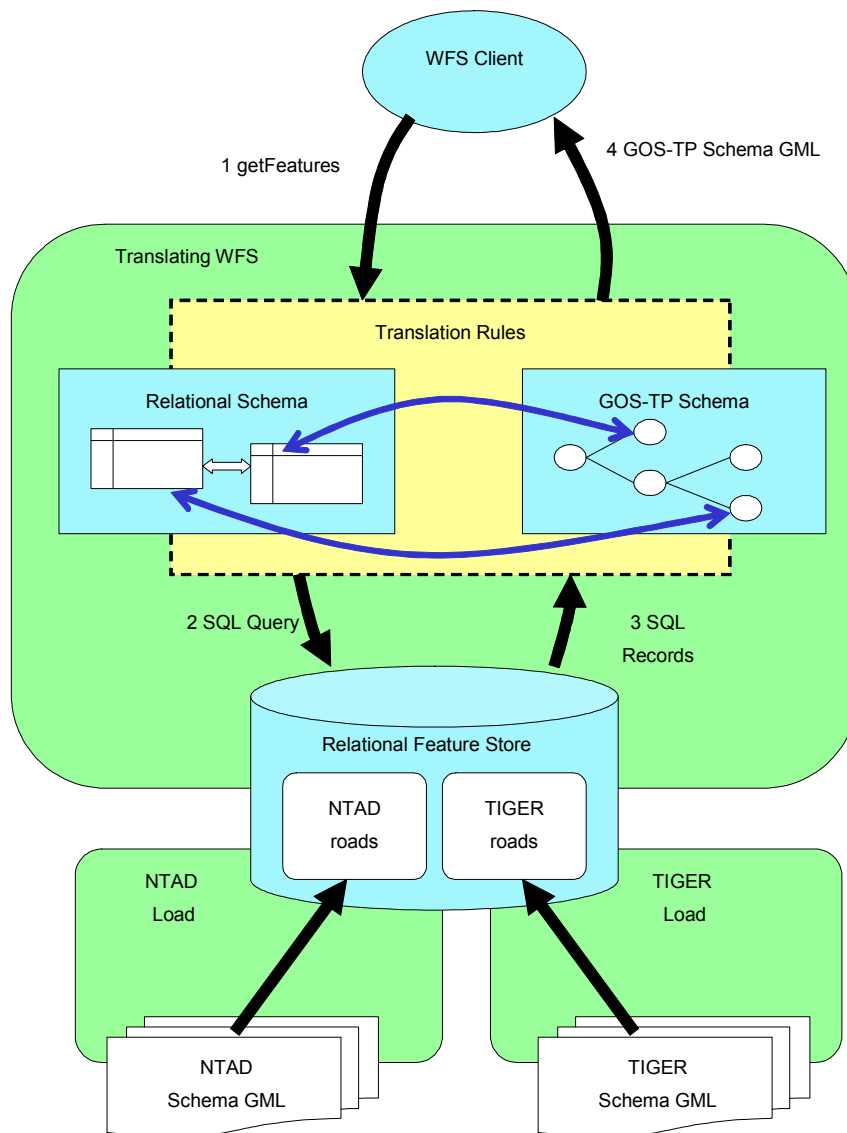
### 2.2 The Snowflake WFS

The Snowflake translating web feature server (WFS-X) translated data from a relational database model into the GML which was served to the client. The database was initially populated from GML2 data. The overall process of translating the data from the source GML2 to output GML3 is therefore a two stage process with the data residing in a relational model between the processes.

The loading and feature serving processes are entirely decoupled. The feature server could therefore be used to serve data entered into the database via any process regardless of its

original source or format. This makes the feature server suitable for serving any data which is stored and maintained in Oracle Spatial.

In this exercise the loading took place as a one time activity to prepare the database for feature serving. (In general the database could be under continuous maintenance from systems connected to the Oracle database.) The translation from the relational model to the output GML is done each time a request is received by the WFS-X



**Figure 2 - The Snowflake WFS-X architecture**

## 2.3 Lessons Learned

The prototypes were successful in translating the source data into a common target schema. The resulting data was read by the client application and data from the different sources could be used together as a single dataset. However, there were limitations to the process.

Whilst the two source data sets superficially appeared to contain the same kind of information in detail there were significant differences in information content.

For example, the target GOS-TP schema requires road features to be connected in a link and node network to allow for navigation through the network. The NTAD data contained a link and node structure and so could be translated into the GOS-TP schema. However, the TIGER data was collected for census purposes and so the emphasis of the attribution of roads is on address, voting and statistical information and not on navigation. The TIGER data did not have a link and node model relating the roads for navigational purposes. The translation process for the TIGER data was therefore required to infer the topological relationships in the TIGER data from the geometry in order to make the links and nodes explicit in the output data. This process is expensive both in terms of processing time and the use of sophisticated software (Laser-scan's Radius Topology software was used within the Snowflake WFS-X to infer the topology). There is also a significant risk of making false inferences and so providing incorrect information in the output.

Inferring the topology is a very complex case of inferring values. There were many simpler cases. In the simplest cases a single value for an attribute in the target schema was deemed to be appropriate for all features from the source data.

Setting up translations was labour intensive. The process requires that a person understand the meaning of the data in both schemas and devise a translation between the two. Even with good quality metadata this is labour intensive. With poor metadata the process could become one of research and guesswork which becomes both time consuming and unreliable.

Semantic information is needed to define the semantics associated with each schema. The GOS-TP schema had well defined syntax but there was little semantic information to guide the developers as to what information should be used to populate each field. The tag structure of XML at least guarantees that every data field has a name and this provides a hint at the meaning associated with each field. The source data had a better quality of metadata provided in the form of data dictionaries providing natural language descriptions of the semantics.

The informal way in which translations were specified resulted in differences in interpretation of the data and schemas in the different implementations. The only formal statements of the translation was in the code and scripts used to implement the translation. These were non-standard and only accessible to the developers implementing the translation. The implementations were informed by an informal statement of the translation which was reviewed and agreed by all participants in the exercise. This specification was broadly successful in achieving consistency between the different implementations of the translation but the informal nature of the specification left scope for differences in interpretation in many points of detail.

### 3 An architecture for semantics driven translation

In order to translate data based on the semantics of the data models a number of kinds of information are needed. Each of these must be formally stated if this process is to be automated.

#### Source and target model syntax

A definition of the syntax for the source and target models.



## Source and target model semantics

A definition of the semantics of the source and target models.

## Semantic mapping

The semantic mapping defines the relationship between elements in the semantic definition of the source and target models. In the simplest cases these mapping could be as basic as identifying equivalent elements in the source and target model semantics although more complex relationships will exist in practical applications.

## Syntax/semantic mapping

A definition of the relationship between the semantics and the syntax of the models.

The relationship between the source data syntax and semantics shows which fields within the data contain the information with each part of the semantics. In the case of the source data this tells the translation process where to find the information in the source data to translate. In the target model this tells the translation process where to put the information when it has been created.

## Syntax mapping

The syntax mapping is a set of rules that define the mapping between values and attributes in the source data and the target model. These rules are executed by the translation engine to produce data in the target model from the source data.

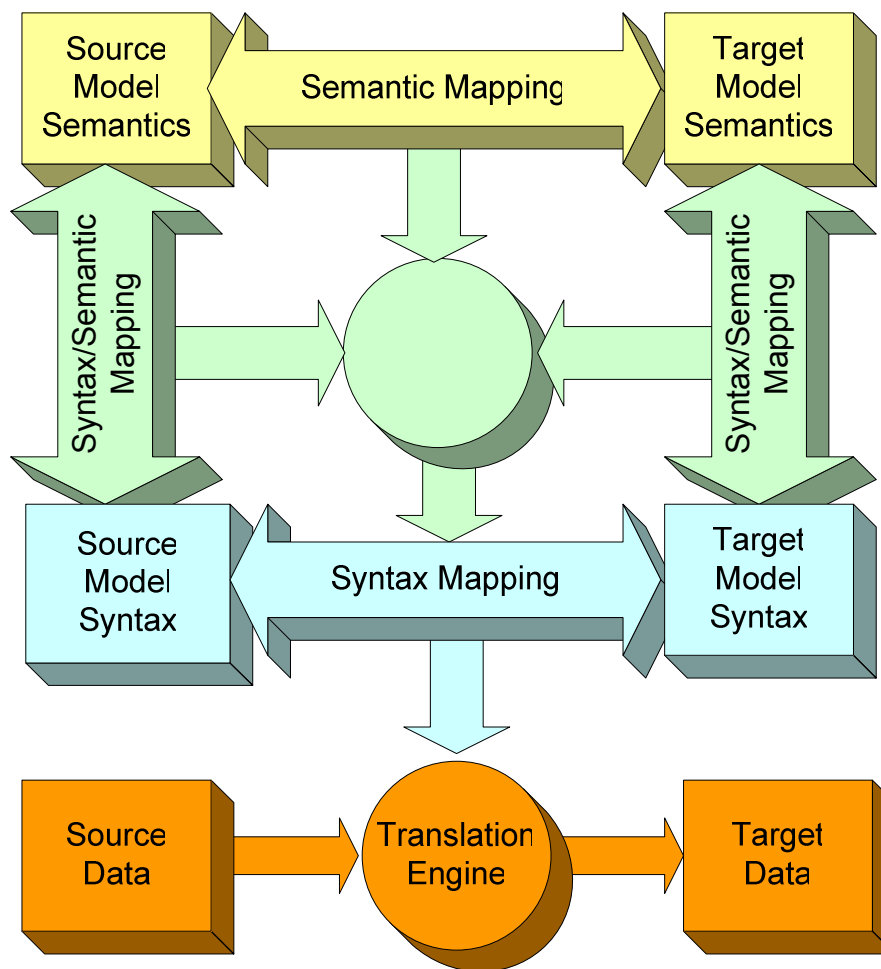


Figure 3 - an architecture for semantic translation

In the case of the CIPI 1.2 project we can consider each of these elements to have been present in some form but for the most part not formally stated:

- The syntax of both source and target models was well defined and formally stated using XML Schema and SQL.
- The principle statement of the syntax mapping was an informal document agreed by the developers. It was stated formally but only in terms of programming languages not generally accessible to anyone other than the developers implementing the translation engines.
- The semantics and the syntax/semantic mapping of the source data was stated informally in the form of data dictionaries and implied in the field names of the target schema.
- The semantic mapping was never explicitly stated in any form although it is implied by the syntax mapping.

With a sufficiently well defined mapping between the source and target model semantics, and a complete mapping between the semantics and syntax for both the source and target models, it should be possible to automate the creation of the syntax based translation rules.

## 4 Future Development

### 4.1 Syntactic Translation

The CIPI 1.2 project showed that a syntactic approach to data translation can be successful in allowing multiple data providers with heterogeneous data stores to collaborate to provide a unified service. For the CIPI 1.2 project the translations were created through software development. For this approach to become practical tool support is needed. Snowflake Software is currently developing an ‘industrial strength’ WFS-X based on the findings from the CIPI 1.2 project to support this process in practical applications.

Appropriate tool support will reduce the effort and skill levels required to create syntactic translations but the process will still require the person developing the translation to gain a detailed working knowledge of both the source and target models.

### 4.2 Semantic Translation

Whilst syntactic translation can provide working solutions to the problem of schema translation it is limited by two factors: The reliability of the translation process is subject to the ambiguities in interpretation of the semantics of the source and target schemas; The cost of creating syntactic translations is high since each translation must be “hand made” by a human who has learned the semantics of both source and target schemas.

A semantic approach to translation would provide a means to address both of these issues.

A semantic approach would involve creating a formal statement of the semantics associated with the source and target schemas. Formal statements can then be made about the relationships between the semantics of each model. This approach makes the relationship between the source and target models explicit instead of implied through a syntactic relationship. This would reduce the risk of errors arising from misinterpretation of the models.

If these statements are sufficiently complete then it will be possible to automatically derive the syntactic translation from the source to the target schema from the statement of semantics. Tools for carrying out a semantics driven translation process can be developed as a progression from syntax based translation. Since all translation must at its lowest level be from one syntax to another, a semantic based translation system can use the semantic information to build the syntactic mapping. Thus it is possible to first build an engine which will carry out syntactic translation and, at a later date add tools which generate the syntactic map from the semantics and utilise the syntactic translation engine to carry out the translation.

Automating the creating of the syntactic translation means that the investment of time and effort put into translation is spent on the semantics rather than the syntax. The definition of the semantics is not tied to one schema or technology. It can therefore equally be used to create, for example, a translation between two GML schemas, between two relational database schemas, or between a relational database schema and a GML schema.



This aids interoperability by making it easier to support many translations. One definition of a translation can be used to support translations using several different technologies. It can be used to translate data between syntaxes of different technologies. The definition of the semantics of the source model can be used as the basis for translating to many target models i.e. the investment in defining the semantics can be re-used. This reduction in the obstacles to translation means that a greater number of translations can be implemented and so barriers to sharing of information are reduced.

## 5 Conclusions

The CIPI 1.2 project showed that a syntactic approach to translation can be used to support interoperability. This will require suitable tool support. The quality of the translation will depend on the understanding of the person setting up the translation of the source and target data models. This, in turn, will be limited by the availability of metadata.

The semantic approach to translation has the potential to reduce cost and increase the reliability of the translation processes. It could result in reusable resources which contribute to an infrastructure to support translation.

Translation tools can be developed in an incremental way by first creating tools to carry out syntactic translation and, at a later date, adding functionality to derive the syntactic mapping from the semantic mapping.