SAFE SOFTWARE

Technology Brief

Semantic Translation

This paper gets at the core of the theoretical foundation of FME. FME is a Semantic Translator, which allows it to do a much better job of translation than any other approach.

Traditional Translation

To best understand the significance of semantic translation, it is appropriate to review first what is meant by traditional translation.

Traditional translation is sometimes compared to a thin pipe connecting one data source to another. The intent is to provide a simple and straightforward approach to accessing foreign data, based on support for a limited set of constructs at both ends of the pipe. Traditional translators are usually one way only, and in some cases, they are oriented towards particular data sets.

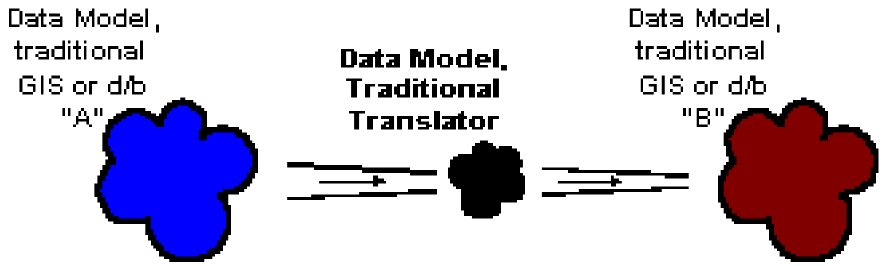
The following points help characterize traditional translation:

- maps geometric primitives from A to B
- simple point & line concepts supported
- traditional attributes often handled separately
- metadata not defined
- demands that input data conform to output model

Additionally, because the models of different systems may differ significantly, a certain degree of information loss typically results.

The underlying model within the translation software is weak, resulting in only basic concepts moving across appropriately. The diagram on the next page indicates the mapping of the respective data models during translation.

When the translator reads data, constructs from a given system ("A") are mapped into an in-memory representation. When the translator writes data, constructs from the in-memory representation are mapped into a second system ("B"). Because the emphasis is principally on reformatting the data, the overall modeling approach tends to be that of the lowest common denominator. In the diagram, the size of the bubbles is an indication of the semantic content of the respective data models.



In some cases it is necessary to restructure the data within a given system such that it meets the criteria for input into the translator. That is, the user of System A may have to redefine the data within her or his system, such that it conforms to a canonical form - the particular form of data which the translator is designed to accept as input.

The inability to handle traditional (non-graphic) attributes means that a user may have to do further processing in order to bring them into conformance with the way they are addressed in her or his system. Unfortunately, no tools are provided to assist this process.

The result of these various deficiencies impacts both data providers and consumers. Frequently both parties must massage the data for the end-to-end transfer to work. This can make the translation process onerous, expensive and time consuming. In practice, translation of given data sets often requires a considerable amount of attention by experienced personnel.

Often with traditional translation, it is desirable to archive the translated data, since the translation exercise is seen as such an impediment; other people can then access this archive without having to go back to the source and subsequent translation headaches. This can lead to the same data being managed multiple times, a situation which otherwise would be avoided.

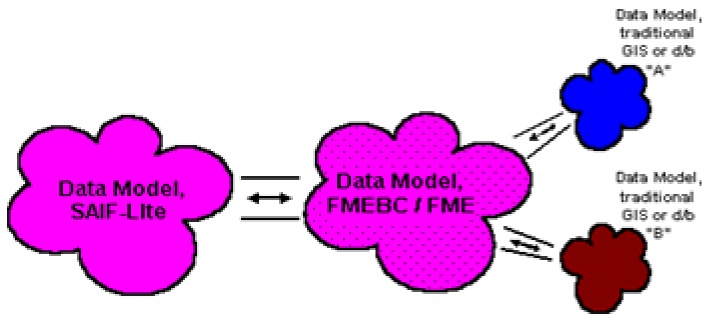
Semantic Translation

Semantic translation differs in a number of important ways. The overall result is that the effort required by both providers and consumers can be considerably less and the associated data management costs can be less as well. The software provides significantly advanced capabilities compared to those described above. In fact, such fat pipe translators represent a different genre of software altogether.

The major points of semantic translation are as follows:

- Defines mapping between intentions in A and those in B
- Uses feature manipulation language
- Allows schema redefinition, e.g., georelational1 to/from georelational2, georelational to/from object-oriented, object-oriented1 to/from object-oriented2
- Provides functions for lookup, counting & some calculations
- Provides factories to create "processed" data from raw data

Semantic translation provides an engine (with a language) which enables the redefinition of the data, either on input or output. Underlying the engine is a rich data model, which is internally consistent and inherently extensible. Constructs within the models of the input or output formats or systems, are mapped to constructs in the engine's model; however, in this case there is no requirement to apply the lowest common denominator. The engine provides a series of methods to carry out model to model transformations, applicable to data either on input or output. This functionality ensures that neither the data provider nor data consumer feels constrained; they can use their respective systems however they wish.



The mapping of the respective data models during translation can be envisioned as shown in the diagram. The data models for the FME and SAIFLite are nearly identical; they are much richer than what is supported in the proprietary systems, allowing for a high degree of redefinition using the built-in FME capabilities. Not only is the mapping highly configurable, but it is bi-directional as well.

The problem of potential information loss is replaced by the decision of how best to maximize information content or how best to meet user needs. A number of options become available. With standard requests against standard data sets, mapping files (defined by the FME language and used to control the translation/transformation process) can be defined in advance and used by anyone.

With non-standard situations, some training in the language is required in order to use the software to its potential. As well, the user or developer of the new mapping files must understand the actual specifications of the input data and the desired specifications of the output data.