Spatio-temporal ontology based model for Data Warehousing

ALBERTO SALGUERO, FRANCISCO ARAQUE, CECILIA DELGADO
Department of Software Engineering :: ETSIIIT
University of Granada (Andalucía)
C/ Periodista Daniel Saucedo Aranda s/n
SPAIN

Abstract: A Data Warehouse (DW) is a database that stores a copy of operational data with an optimized structure for query and analysis. There are many facets due to the number of variables that are needed to consider in the integration phase design. It is not easy to generate and maintain the integrated scheme. In this paper we describe a spatio-temporal extension of an ontology language which facilitates the generation of the scheme of the DW as well as the design of the processes which extract, transform and load the data from the sources in the DW according to the temporal characteristics of the data sources. The proposed model allows the user to concentrate in the problem itself and not in the issues of dealing with the temporal and the spatial concepts found in many of the data sources usually used in the enterprises information systems.

Keywords: Data warehouse, spatial model, temporal model, data integration, ontology.

1 Introduction

There are many problems, from the computational point of view, whose solution involves the use of temporal or spatial concepts. In the case of information systems, which are often used by companies in their strategic level, sometimes occurs that many of the data sources from which the information is obtained contain, in a way, this kind of spatial and temporal information. Moreover, because of the proper nature of information systems based on DWs (which we are interested in) data sources contain an intrinsic temporal characteristics that establish the proper manner to extract their data.

A DW is a database that stores a copy of operational data with an optimized structure for query and analysis. The data coming from the sources are integrated in the data warehouse to satisfy the necessities of collective access to the data, providing an historical vision of the data of the different operational systems.

Usually, whenever a company is faced with the task of including a new data source in its information system, they often face the process virtually from scratch. There are very few tools for assisting in the process of designing information systems based on DW [1] and the inclusion of new data sources may involve the re-design of the scheme of DW.

This paper describes an ontology based model serving as the pivot which the different data sources schemes are translated to (Canonical Data Model). From the data source schemes it is possible to generate an integrated DW straightforwardly. This data model defines the spatial and temporal concepts used more frequently so they can be reused in each specific problem and the efforts are focused on the problem itself, and not in the definition of spatial and temporal information concepts in the source.

On the other hand, this data model, heavily oriented to work with information systems based on DW, also has the ability to annotate each of the data sources with information about their own temporal characteristics, making it possible to generate semi-automatically a set of rules that determine the most efficient way to extract information according to the level of detail required by the DW designer without having to query the data sources more than necessary.

The use of a data model based on ontologies is proposed as a common data model to deal with the data sources schemes integration. This idea is based on [2], a project being developed by members of the University of Granada. Although it is not the first time the ontology model has been proposed for this purpose [3], in this case the work has been focused on the integration of spatio-temporal data. Moreover, to our knowledge this is the first time the metadata storage capabilities of some ontology definition languages has been used in order to improve the DW data refreshment process design.

The remainder of the paper is organized as follows. In Section 2 some basic concepts are revised. Section 3 introduces our spatio-temporal model. In Section 4 the usage of the spatio-temporal model is shown. Finally, in Section 6 the conclusions are presented.
2 Data Warehouses and ontologies

Inmon [4] defined a Data Warehouse as “a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management’s decision-making process”. A DW is a database that stores a copy of operational data with an optimized structure for query and analysis [5]. The scope is one of the issues which defines the DW: it is the entire enterprise. In terms of a more limited scope, a new concept is defined: a Data Mart (DM) is a highly focused DW covering a single department or subject area. The DW and data marts are usually implemented using relational databases which define multidimensional structures. A federated database system (FDBS) is formed by different component database systems; it provides integrated access to them: they co-operate (inter-operate) with each other to produce consolidated answers to the queries defined over the FDBS. Generally, the FDBS has no data of its own as the DW has. Queries are answered in the FDBS by accessing the component database systems.

We have extended the Sheth & Larson five-level FDBS architecture [6], which is very general and encompasses most of the previously existing architectures. In this architecture three types of data models are used: first, each component database can have its own native model; second, a canonical data model (CDM) which is adopted in the FDBS; and third, external schema can be defined in different user models.

One of the fundamental characteristics of a DW is its temporal dimension, so the scheme of the warehouse has to be able to reflect the temporal properties of the data. The extracting mechanisms of this kind of data from operational system will be also important. In order to carry out the integration process, it will be necessary to transfer the data of the data sources, probably specified in different data models, to a common data model, that will be the used as the model to design the scheme of the warehouse. In our case, we have decided to use an ontological model as canonical data model.

An ontology is the specification of a conceptualisation of a knowledge domain. It is a controlled vocabulary that describes objects and the relations between them in a formal way, and has a grammar for using the vocabulary terms to express something meaningful within a specified domain of interest. They allow the use of automatic reasoning methods. OWL is a language for defining ontologies. OWL ontologies may be categorised into three species or sub-languages: OWL-Lite, OWL-DL and OWL-Full.

OWL-Lite is the syntactically simplest sub-language. It is intended to be used in situations where only a simple class hierarchy and simple constraints are needed. OWL-Full is the most expressive OWL sub-language but is not able to guarantee the decidability or computational completeness of the language, so it is not always possible to perform automated reasoning on it. OWL-DL is much more expressive than OWL-Lite and allows the use of automated reasoning restricting some OWL-Full constructions.

We have extended OWL with temporal and spatial elements. We call this OWL extension STOWL. This is also our proposal: use an ontological-oriented model as CDM, enhanced with temporal and spatial features, for the definition of the data warehouse and data mart schema to define the refreshment of the data warehouse and data marts.

3 STOWL: The Spatio-Temporal OWL

According to the integration architecture proposed in [7], [8], [9] it is necessary the selection of a CDM which serves as pivot which all the schemes that specify the structure of the different information sources and the DW will be translated to. The obtained DW scheme will generate, in turn, the schemes which will be used by the query and the data analysis tools. The requirements to be performed by this model are:

- Correctly treatment of temporal information. Being an architecture geared towards building data warehouses it must be able to provide high-quality support for this type of information because it is the basis of this kind of architecture (a data warehouse is basically a set of historical data). This is achieved by defining temporal primitives.
- Support for the adequate representation of spatial information: geographical, topological… The model must define spatial primitives allowing the representation and the proper treatment of information from the different data sources.
- Semantic data annotation to support their integration. It should includes all the available information in the data sources to make easy the design of the extracting, transforming, loading and refreshing data tasks from the DW.

The data models based on ontologies are a good option as a CDM but too generals. All studied language proposals for defining ontologies used ultimately a very limited set of basic data types to define underlying concepts. This set of basic types
are usually very general and does not support the spatial information types. Extending any of the languages for defining ontologies to cover these requirements seem the most suitable option. Among all of them, OWL language is selected as the base for defining the CDM.

Starting from the OWL basic data type set we can express in form of ontology the common spatio-temporal concepts required for the schema definition of data sources to perform the integration phase. Thus, when describing the scheme of the data source, the design process will focus on the data source characteristics and not in the process of specifying spatio-temporal concepts.

Firstly an ontology in the OWL language will be built to define the generic temporal primitives found of interest. Then the spatial primitives will also be incorporated. Finally, the information that describes the characteristics of data to integrate, i.e. the metadata which will be useful to design the data extracting, loading and refreshing processes, will be incorporated to the data source scheme using the annotation properties of OWL. Annotation properties are a special kind of OWL properties which can be used to add information (metadata— data about data) to classes, individuals and object/datatype properties.

The result will be an ontology, expressed in OWL, defining the spatial and temporal primitives which will be used to build the schemes of the data sources and a set of properties which will allow the addition of information about the data sources characteristics. We call STOWL (Spatio-Temporal OWL) to this base ontology.

Basically, STOWL is an application of OWL. The primitives on which STOWL rests come basically from GML (spatial information) and ODMG-T (temporal information) data models.

OWL allows classes, properties, individuals and the ontology itself (technically speaking the ontology header) to be annotated with various pieces of information/metadata. These pieces of information may take the form of auditing or editorial information. For example, comments, creation date, author, or, references to resources such as web pages etc.

OWL Full does not impose any restriction in the use of the annotation properties. The problem of using OWL Full is that, under certain restrictions, the resulting ontology still satisfies the OWL DL requirements:

- The filler for annotation properties must either be a data literal, a URI reference or an individual.
- Annotation properties cannot be used in property axioms — for example they may not be used in the property hierarchy, so they cannot have sub properties, or be the sub property of another property. They also must not have a domain and a range set for them.

The ontology itself can be seen as a resource that describes the scheme of a data source. Therefore, this scheme can be annotated with information about the general characteristics of the source. This information will be used to design the extracting, transforming and loading process of the data in the DW.

3.1 Temporal annotation of data

Due to the nature of the DW the annotation properties will usually refer to the temporal characteristics of data, so a set of annotation properties is defined to describe the sources according to some of the temporal concepts studied in [8]. All these properties are associated directly to the ontology viewed as a resource and not individually to each resource defined in the ontology. This means that all the resources of the ontology share the values assigned to these properties (expected if extracted from the same data source).

Figure 1: Extraction Time definition in STOWL.

STOWL defines, for instance, the Extraction Time of a change in a data source as shown in figure 1. The Extraction Time parameter can be defined as the time expended in extracting a data change from the source. Some examples of the temporal parameters [10] that we consider of interest for the integration process are:

- Availability Window (AW). Period of time in which the data source can be accessed by the monitoring programs responsible for data source extraction.
- Extraction Time (ET). Period of time taken by the monitoring program to extract significant data from the source.
- Transaction time (TT). Time instant when the data element is recorded in the data source.
computer system. This would be the data source transaction time.

- **Storage time (ST).** Maximum time interval for the delta file, log file, or a source image to be stored.

- **Temporal Reference System (TRS):** A temporal reference system provides standard units for measuring time and describing temporal length or duration. Gregorian calendar with UTC is often used as the default temporal reference system.

All the temporal concepts explained previously can be applied to a data source as a whole because all data from the same source share these temporal characteristics. On the other hand, there are certain characteristics that should be defined in more detail. Within the scheme of the data source, each of the data fields that can be extracted can be annotated semantically with temporal information to further facilitate their integration into the data warehouse.

As with the case of generic data source features, the annotation properties to describe the characteristics of different data fields will also be used. For the temporal aspects of data STOWL defines the following property:

- **Temporal Granularity (TGr):** It is the extent to which a system contains discrete components of ever-smaller size. Generally speaking, information granules are collections of entities, usually originating at the numeric level, that are arranged together due to their similarity, functional adjacency, indistinguishability, coherency or the like. In our case, because we are dealing with time, it is common to work with granules like minute, day, month…

### 3.2 Spatial annotation of data

Once incorporated the temporal primitives to the CDM it is necessary to define the spatial primitives. After reviewing various spatial data models we have chosen to follow the model proposed by the Open GIS Consortium. The main reason is that it is a standard widely accepted by the majors companies working on GIS systems. This standard, called Geography Markup Language (GML), is an XML grammar written in XML Schema for the modelling, transportation and storage of geographic information. GML is often used as a communication protocol between a large set of GIS applications, both commercial and open source. This spatial data model has been tested and polished over years and it is accepted by most of GIS related companies and associations. It has been necessary the translation of GML, written in XML Schema, to OWL.

GML defines several kinds of entities (such as features, geometrical entities, topological entities…) in form of a object hierarchy. As well as the temporal properties they have been incorporated to STOWL in order to support the inclusion of spatial metadata in the data sources schemes.

There are also spatial characteristics which can be associated to the data source as well as other kind of spatial characteristics which must be associated to each data field of the data source. In the former set we have, for instance, the Coordinate Reference System (CRS) concept. A coordinate reference system consists of a set of coordinate system axes that is related to the earth through a datum that defines the size and shape of the earth. A CRS provides a method for assigning values to a location. All the data in a data source must share the same CRS.

In the latter set of spatial characteristics we consider, for instance, the Spatial Granularity (SGr). In this case, because we are dealing with spatiality, it is common to work with granules like meter, kilometer, country…

As with the temporal data source features, the annotation properties are used to describe the source spatial metadata.

### 4 Defining new data source schemes

We have developed a plug-in for Protégé which allows an easy definition of new data sources schemes using the STOWL language. Protégé is a free, open source ontology editor and a knowledge acquisition system. It is being developed at Stanford University in collaboration with the University of Manchester.

The operation of the plug-in is pretty simple. From all of the properties defined in STOWL, those considered to be of interest for defining the schemes of data sources by users are modified to be also a subproperty of a property called “externalProperty”. All these properties will be available to the users when designing the schemes of the data sources. This approach allows an easy and transparent modification of the classes and basic properties of STOWL for the end user.

Let suppose a company which needs to incorporate tourist information from a database of hotels to its information system. One of the typical properties that share the hotels is the definition of different holiday periods depending on the demand (high, medium and low seasons). In this case, the user should follow the following sequence of steps to define this property using the plug-in we have developed:
1. Create a new class named “Hotel”. The plug-in creates a new class with the name chosen by the user and makes it inherits directly from the class “OWL: Thing”.

2. Create a new attribute for the class “Hotel” called “hasHighSeason”. Internally, the plug-in creates a new property called “hasHighSeason” and assign the class “Hotel” as its domain.

3. Select STOWL property that best matches. Using a drop-down menu, containing the complete list of properties defined as external in STOWL, the user has to select the property which best mach with the new created property. Following with the tourist example, the most appropriate property in this case is the “hasPeriod” property whose range is instances of the class “TimePeriod”. The new property “hasHighSeason” will be created by the plug-in in such a way that it inherits directly from the property “hasPeriod” defined by STWOL, inheriting all its features.

In figure 2 is shown how the new “hasHighSeason” property is defined and how it is linked with the rest of the properties. It is important to note that, because of the inherent reasoning capability of ontology based model, the class “Hotel” would also be automatically assigned to the class of dynamic objects (the domain of the property “hasPeriod” is the class “DynamicObject”) without the user having to take part. Using this ontology feature is possible to make queries involving some kind of reasoning. This feature can also be used in the integration process of the different data sources schemes.

Once defined the “hasHighSeason” property we can use it to describe the data in the data sources, i.e. the ontology instances. In figure 3 it is illustrated how the new “hasHighSeason” would be used to indicate that the high season period for the Hesperia hotel goes from May to September. The objects below the line are the only classes or properties the developer of the data source scheme should create. The classes and properties above the line are objects defined by STOWL and reusable in different applications and data sources scheme.

It is also important to notice that the time period instance “May-September” has a granularity of month. To indicate this fact the designer of the data source scheme has used the annotation property “hasGranularity” of the class “TimePeriod”. Knowing this information it is possible to assure that it is not possible to obtain an integrated scheme where the level of detail of the season time periods will be smaller than a month. The DW refreshing process can take advantage of this knowledge and query the data sources consequently.

5 Conclusions

A data warehouse is a database that stores a copy of operational data with an optimized structure. The data coming from the sources are integrated in the data warehouse to satisfy the necessities of collective access to the data.

To carry out the data integration process, it is necessary to transfer the data from the data sources, probably specified in different data models, to the data warehouse. To overcome these differences and for design the data warehouse schema it is used a common data model.

As common data model, we have suggested an ontology-based model; in particular, we have defined an spatio-temporal extension of the OWL language, and to store metadata about the temporal data sources characteristics we propose the use of the annotation properties of OWL.
Making use of the reasoning capabilities of OWL it is possible to derive rules to integrate the schemes of the different data sources and to design efficient DW refreshing process.

Acknowledgements

This work has been supported by the Research Program under project GR2007/07-2 and by the Spanish Research Program under projects EA-2007-0228 and TIN2005-09098-C05-03.

References