

Transforming multidimensional models into OWL-DL ontologies

Nicolas Prat
ESSEC Business School
Avenue Bernard Hirsch – BP 50105
95021 Cergy cedex – France
prat@essec.edu

Jacky Akoka
CEDRIC-CNAM & TEM Research
2 rue Conté
75003 Paris – France
akoka@cnam.fr

Isabelle Comyn-Wattiau
CEDRIC-CNAM &
ESSEC Business School
2 Rue Conté – 75003 Paris – France
wattiau@cnam.fr

Abstract— Business intelligence is based on data warehouses. Data warehouses use a multidimensional model, which represents relevant facts and their measures according to different dimensions. Based on this model, OLAP cubes may be defined, enabling decision makers to analyze and synthesize data. Ontologies (and, more specifically, OWL ontologies) are a key component of the semantic Web. This paper proposes an approach to represent multidimensional models as OWL-DL ontologies. To this end, it presents the multidimensional metamodel, the concepts of OWL-DL, and transformation rules for mapping a multidimensional model into an OWL-DL ontology. It then illustrates application to a case study with a simplified example of a spatiotemporal data warehouse. The transformation rules are refined to deal with spatiotemporal data warehouses, applied step by step, and the resulting ontology is implemented in the Protégé ontology tool. As illustrated by the case study, our approach enables better formalization and inferencing, thanks to OWL-DL. The ontology may also be used to represent OLAP cubes on the semantic Web (with RDF), by defining these cubes as instances of the OWL-DL multidimensional ontology.

Keywords: *business intelligence; multidimensional model; OLAP; ontology; OWL-DL; spatiotemporal data warehouse*

I. INTRODUCTION

Business intelligence software “*is a collection of decision support technologies for the enterprise aimed at enabling knowledge workers such as executives, managers, and analysts to make better and faster decisions*” [1]. Business intelligence facilitates decision making, by providing managers with user-friendly and integrated tools. It is based on data warehouses, which can be defined as databases specifically aimed at decision making. Data warehouses use a multidimensional model, which represents relevant facts and their measures according to different perspectives (dimensions). Based on this model, OLAP (On-Line Analytical Processing) cubes may be defined, enabling decision makers to analyze and synthesize data according to different perspectives.

The semantic Web (also sometimes referred to as “Web 3.0”) is progressively becoming a reality. It is the Web of

machine-interpretable data. Two components are central to the semantic Web: (i) RDF (Resource Description Framework) [2], which represents data as subject-property-object triples, and (ii) ontologies, which convey meaning by explicitly formalizing “*conceptualizations of domain models*” [3]. The semantic Web, also sometimes referred to as the Web of linked data [4], is progressively becoming a reality.

Business intelligence and the Web 3.0 are major issues for IS research [5], and more specifically for design research. These two topics are also complementary to some extent. It is therefore natural to combine business intelligence (and more specifically multidimensional modeling and OLAP) with the semantic Web (more specifically ontologies).

In the semantic Web, the standard language for representing ontologies is OWL [3]. OWL has several variants, including OWL-DL, based on description logic [6]. Description logic facilitates formalization and reasoning on ontologies. In the literature, some research has already explored the relationship between multidimensional modeling/OLAP and the semantic Web (and more specifically OWL ontologies). Niemi and Niinimäki [7] provide an RDF model of an OLAP cube. This paper stresses the key role of aggregation knowledge. However, it is not based on OWL, and the RDF model lacks some important concepts like dimension levels and hierarchies within dimensions. Cyganiak, Reynolds and Tennison propose the RDF Data Cube vocabulary (<http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/html/cube.html>). This vocabulary is meant to be generic and lacks some important concepts for multidimensional modeling. Nebot et al. [8] present a framework for designing semantic data warehouses. To this end, they use a Multidimensional Integrated Ontology, but the structure of this ontology is not shown. Finally, Xie et al. [9] use a multidimensional ontology based on OWL-DL, but, in this paper also, the structure of the ontology remains implicit.

Summing up, we found no publication presenting a complete OWL ontology of multidimensional concepts. This is

clearly a gap in the literature, especially when compared to the extant literature dedicated to mapping relational databases into OWL ontologies (for example, [10], [11], [12], [13]). In this paper, we aim at filling this gap, by presenting an approach to define an OWL-DL ontology from a multidimensional model. To achieve this, we present a multidimensional metamodel (simplified from our model in [14]), and the OWL-DL ontology language (slightly modified from [15]). Both models are presented as UML class diagrams [15]. We then present the transformations, which map a multidimensional model (instance of the multidimensional metamodel) into an OWL-DL ontology (instance of the OWL-DL language). Finally, we present a case study with a simplified example of spatiotemporal data warehouse. The application to spatiotemporal data warehouses leads us to refine our multidimensional metamodel and our transformation rules; we then illustrate the step-by-step application of the rules and implementation with the Protégé ontology tool (www.protege.stanford.edu/). As illustrated by the Protégé implementation, our approach, which shows how to represent a multidimensional model as an OWL-DL ontology, enables better formalization and inferencing, thanks to OWL-DL. The ontology may also be used to represent OLAP cubes on the semantic Web (with RDF), by defining these cubes as instances of the OWL-DL multidimensional ontology.

The paper is organized as follows. Section II presents the multidimensional metamodel. Section III summarizes the OWL-DL language. Section IV presents the transformation rules mapping a multidimensional model (instance of the multidimensional metamodel) into an OWL-DL ontology (instance of the OWL-DL language). Section V illustrates application to spatiotemporal data warehouses: after specifying our approach for this type of data warehouse, we apply the transformation rules step by step (by showing the resulting OWL-DL axioms and facts), and implement and test the resulting ontology in Protégé. Section VI concludes and points to further research.

II. THE MULTIDIMENSIONAL METAMODEL

The multidimensional metamodel is central to our approach. It is the source model for the transformations.

Figure 1 represents the multidimensional metamodel. Due to length constraints, this model is simplified from our conceptual multidimensional metamodel presented in [14].

The two main concepts of the model are dimension and fact. A dimension is composed of hierarchies. A hierarchy is an aggregation of dimension levels. Dimension levels have attributes and are related by rollups. A rollup links a child dimension level to a parent dimension level. For the child and parent, several lower and upper multiplicities are possible (e.g. the model allows a dimension level instance to roll up to several dimension level instances, for example a professor may belong to several departments). Facts are composed of measures and dimensioned by dimensions. The relationship between a fact and its dimensioning dimensions is called a dimensioning and is also characterized by its multiplicities. Usually, the lower and the upper multiplicity of a dimensioning are equal to one (e.g. a sale is made at one and only one date, concerns one and only one product...).

For a given measure along a given dimension, applicable aggregation functions may be specified. The model distinguishes several aggregation types [16]. In aggregation type 1, all aggregation functions (sum, avg, min, max, count) may be used. In aggregation type 2, only avg, min, max and count may be used. In aggregation type 3, only count may be used. Among other factors, applicable aggregation functions depend on the semantics of the measure and the semantics of the dimension.

It should be noted that a special dimension level, called All (with only one instance, called "all"), is often used in multidimensional models. This ensures that all dimension levels with no parent rollup to this terminal dimension level.

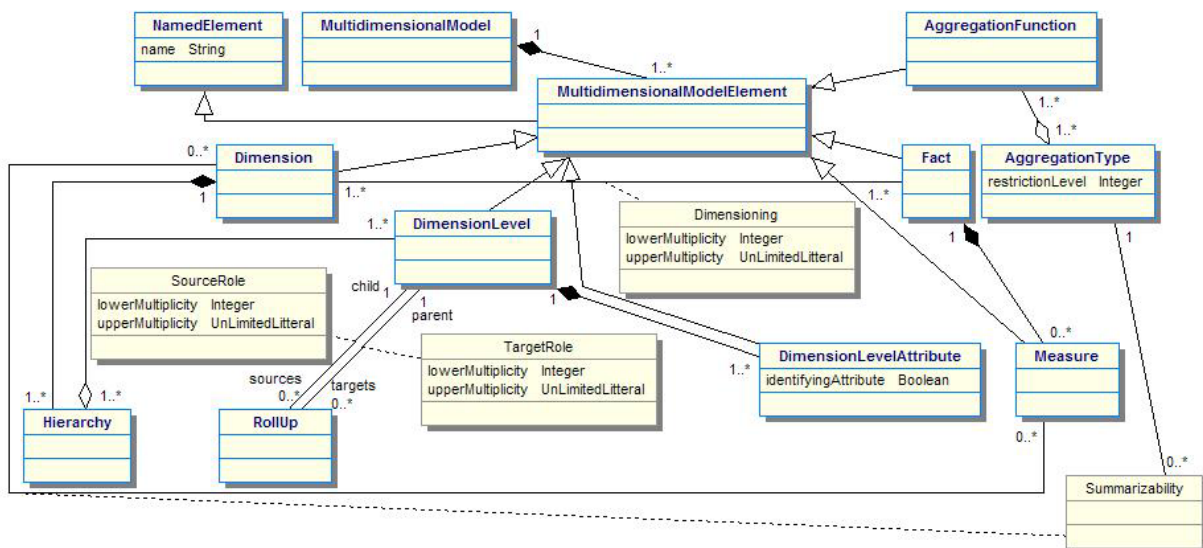


Figure 1. Multidimensional metamodel

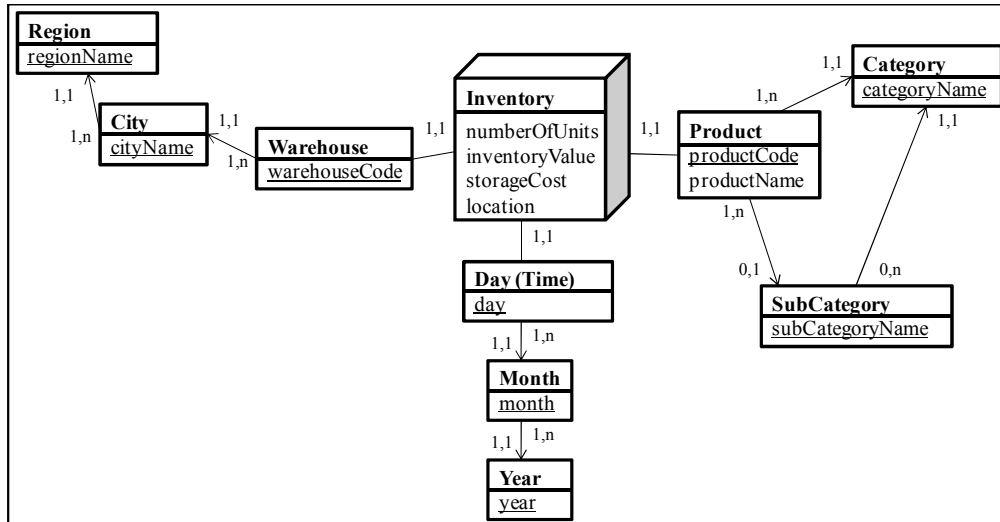


Figure 2. Example multidimensional model

Figure 2 illustrates a multidimensional model (instance of the multidimensional metamodel). This model is a simplified model of an inventory system. We will use the model to illustrate the mapping transformations in section IV. Figure 2 uses the formalism that we propose in [17]. It clearly shows the dimensions, dimension levels and their attributes, facts and their attributes (measures), rollups and dimensionings. The multiplicities for the rollups and dimensionings are also shown. In particular, the model shows that some products have no

subcategory. For the dimension Product, there are two hierarchies: one relates Product to SubCategory to Category; the other relates Product directly to Category. The fact Inventory is characterized by four measures: the number of units of a given product stored in a given warehouse at a given date, the corresponding value of the inventory, the storage cost, and the exact location (e.g. GPS coordinates) where the product is stored in the warehouse at this date.

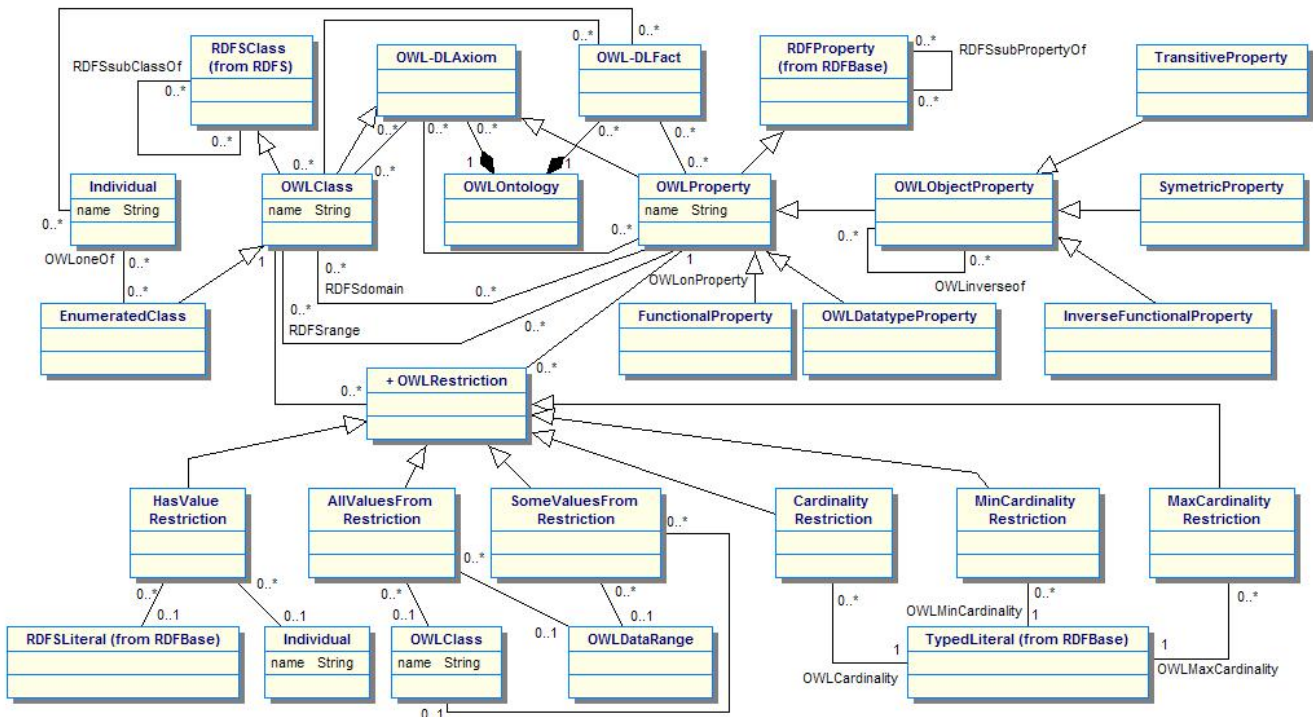


Figure 3. The OWL-DL ontology language (adapted from [15])

III. THE OWL-DL WEB ONTOLOGY LANGUAGE

OMG [15] proposes a metamodel of OWL as part of the Ontology Definition Metamodel. Figure 3 summarizes the main concepts. In this paper, we focus on OWL-DL.

An OWL-DL ontology is composed of axioms (at the type level) and facts (at the instance level). At the type level, the main concepts are the concepts of class and property. At the instance level, the main concept is the concept of individual (instance of one or several classes). Properties may be defined on classes (called their domain). The values of properties are specified in their range. The range of object properties consists in classes; the range of data properties consists in classical data types (e.g. integer...). Several restrictions may be defined on classes and properties (e.g. cardinalities).

For more details on OWL, the reader is referred to [3], [15], and to the W3C specification (<http://www.w3.org/TR/owl-features/>).

IV. FROM MULTIDIMENSIONAL MODELS TO OWL-DL ONTOLOGIES

The transformations map a multidimensional model (represented as an instance of the multidimensional metamodel presented in section II) into an OWL-DL ontology (represented as an instance of the OWL-DL ontology language presented in section III). We present the transformations (the step-by-step application to a case study will be illustrated in section V).

Since the source and target metamodels are represented with UML, the transformations may be represented with the QVT language (Query/View/Transformation [15]). Complementary to the natural language specification, QVT is a powerful language for formally representing mapping transformations. Due to space limitations, we illustrate this QVT representation for transformation T2.2 only.

Importantly, OWL makes the “open world assumption”: all that is not stated as true is assumed unknown. For example, the subclasses of a given class are not assumed disjoint unless this disjointness is explicitly stated. In our case, although we don't say it explicitly in the different transformations, all dimensions should be specified as disjoint, as well as all dimension levels, all measures, etc. This is necessary to guarantee correct OWL-DL Inferences.

The transformations map:

- dimensions, hierarchies, dimension levels, dimension level attributes, and rollups (transformations T1.1 to T1.7)
- facts, measures and dimensionings (transformations T2.1 to T2.3)
- aggregation functions and aggregation types (transformation T3).

In order to group concepts of the same type, OWL-DL classes with prefix “A_” are created (e.g. A_DimensionLevel, A_Measure...).

Transformation T1.1: Each dimension of the multidimensional model becomes a class in the OWL-DL ontology.

Transformation T1.2: Each hierarchy of the dimension (created in transformation T1.1) becomes a class in the OWL-DL ontology. The association between the dimension and the hierarchy becomes an OWL-DL object property, whose domain is the class corresponding to the dimension and whose range is the class corresponding to the hierarchy.

Transformation T1.3: Each dimension level of the hierarchy (created in transformation T1.2) becomes a class in the OWL-DL ontology. The association between the hierarchy and the dimension level becomes an OWL-DL object property, whose domain is the class corresponding to the hierarchy and whose range is the class corresponding to the dimension level.

Transformation T1.4: Each dimension level attribute becomes a data type property in the OWL-DL ontology. The domain of this property is the class corresponding to the dimension level.

Transformation T1.5: An object property named rollup and its inverse property named drilldown are created in the OWL-DL ontology. For every rollup R in the multidimensional model with dimension levels DL_1 as child and DL_2 as parent: (i) for the class corresponding to DL_1 , some values of the rollup property are from the class corresponding to DL_2 , and (ii) for the class corresponding to DL_2 , some values of the drilldown property are from the class corresponding to DL_1 .

Transformation T1.6: For every rollup R in the multidimensional model with dimension levels DL_1 as child and DL_2 as parent: (i) for the class corresponding to DL_1 , a minimum and a maximum cardinality restriction are created on the rollup property with (respectively) the lower and the upper multiplicity of the rollup parent (i.e. DL_2) in the multidimensional schema, and (ii) for the class corresponding to DL_2 , a minimum and a maximum cardinality restriction are created on the drilldown property with (respectively) the lower and the upper multiplicity of the rollup child (i.e. DL_1) in the multidimensional schema.

It should be noted that OWL-DL does not allow specifying a cardinality restriction without specifying an exact value. Consequently, if we just want to specify a maximum cardinality of “several”, we cannot represent this information.

When the minimum and maximum cardinality are the same, a single cardinality restriction can be used.

Transformation T1.7: A class named All (a dimension level with a single instance “all”) is created in the OWL-DL ontology. For each dimension level DL_1 with no parent in the multidimensional model (i.e. for each terminal dimension level), a restriction is created on the rollup property for the class corresponding to DL_1 . This restriction states that all values of this property are from the class All. For all terminal dimension levels, the rollup property has the value all.

Transformation T2.1: Each fact of the multidimensional model becomes a class in the OWL-DL ontology.

Transformation T2.2: Each measure of the fact (resulting from transformation T2.1) becomes a class in the OWL-DL ontology. The relationship between the fact and the measure becomes an OWL-DL object property, whose domain is the class corresponding to the fact and whose range is the class corresponding to the measure.

Figure 4 illustrates the QVT representation for transformation T2.2 (in this QVT representation, “mult” represents the source multidimensional model, and “onto” the target OWL-DL ontology).

Transformation T2.3: An object property named dimensionedBy is created in the OWL-DL ontology. For every dimensioning between a fact F and a dimension D in the multidimensional model: (i) for the class corresponding to F, some values of the dimensionedBy property are from the class corresponding to D, and (ii) for the class corresponding to F, a minimum and a maximum cardinality restriction are created on the dimensionedBy property with (respectively) the lower and the upper multiplicity of the dimensioning in the multidimensional model.

Transformation T3: The object properties Summable, Averageable, Maxable, Minable and Countable are created in the OWL-DL ontology. Averageable, Maxable and Minable are equivalent properties; Summable is a subproperty of Averageable; Averageable is a subproperty of Countable. For

each summarizability S in the multidimensional model relating a measure M with a dimension D: (i) if the restriction level of the aggregation type is 1, for the class corresponding to M, some values of the Summable property are from the dimension D, (ii) if the restriction level of the aggregation type is 2, for the class corresponding to M, some values of the Averageable property are from the dimension D and no values of the Summable property are from the dimension D, and (iii) if the restriction level of the aggregation type is 3, for the class corresponding to M, some values of the Countable property are from the dimension D and no values of the Averageable property are from the dimension D.

V. APPLICATION TO SPATIOTEMPORAL DATA WAREHOUSES (SDW)

In this section, we apply our approach to the case of spatiotemporal data warehouses (SDW). In these data warehouses, the special semantics of time and space directly influences the ways in which information may be aggregated. Consequently, when dealing with spatiotemporal data warehouses, it is relevant to (i) specialize our multidimensional metamodel (specializing some concepts of the metamodel) and (ii) specify in more details some of the transformations described previously. Thanks to OWL-DL, special semantics related to time and space can be represented and inferences can be performed based on this semantics. In this section, we review the specificities of spatiotemporal data warehouses, adapt our metamodel and transformations rules for this type of data warehouses, illustrate the transformation rules with an example, and implement the resulting ontology in Protégé. With his implementation, we illustrate some of the benefits of OWL-DL inferencing.

A. Spatiotemporal data warehouses

It is generally estimated that 80% of data stored in spatiotemporal databases have a spatial component [18]. These geographic data are described by:

- a spatial component representing the form and the location on the earth surface of complex objects,
- a semantic component representing the descriptive properties of the objects such as names, size of the population in a given region, etc.

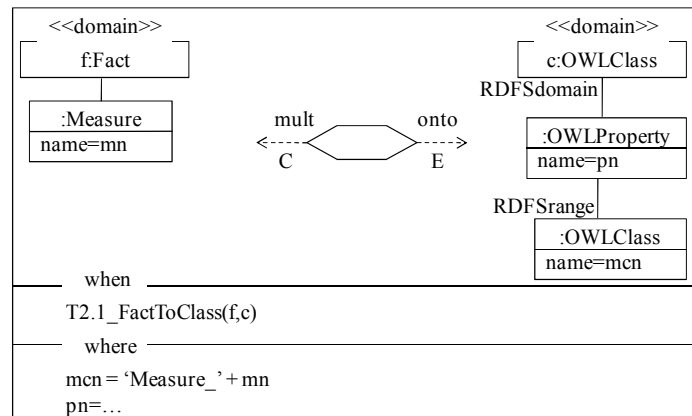


Figure 4. QVT representation for transformation T2.2

The spatial dimension has been integrated in data warehouse and OLAP systems which can be combined to form spatial data warehouses [19]. This integration leads to Spatial OLAP (SOLAP). The concept of SOLAP has been introduced by Rivest, Bédard, and Marchand [20], enabling the exploration of spatial data by drilling on maps. SOLAP concepts have been implemented in JMAP, a commercial tool (<http://piet.exp.dc.uba.ar/piet/index.jsp>). SDW such as SOLAP allow users to analyze spatiotemporal databases to infer hidden relations. They have been applied in several domains such as health, environment, etc. They can be constructed using a spatial model defined by:

- spatial dimensions
- spatial measures
- aggregation functions.

Spatial dimensions are described by spatial hierarchies characterizing the spatial attributes at different spatial levels. An example of a spatial hierarchy is the US geographic organization grouping cities into counties and counties into states. Bédard, Rivest, and Proulx [21] classify spatial dimension hierarchies into three categories: (i) non-geometric, representing only descriptive dimensions, (ii) geometric to non-geometric, and (iii) fully geometric. For the last two categories, geometry is associated with members of the hierarchies. Malinowski and Zymani [22] extend this classification considering that a dimension level is spatial if it is represented as a spatial data type (point, region, etc.). Da Silva, Times, and Salgado [23] provide a framework, called GeoDWFrame, classifying dimensions as geographic data or geographic and non spatial data. A dimension is considered to be spatial if it contains at least one spatial dimension level [24].

Measures in SDW are generally defined as the results of metric operators. They are described by alphanumeric attributes and geometry. More precisely, these *measures* can be alphanumeric, *spatial*, or temporal. In their example, Boulil, Bimonte, and Pinet [24] distinguish between number of injuries (numerical measure), and fire zone (spatial measure). Malinowski and Zymani [22] represent spatial measures either as a geometry which can be aggregated along the dimensions or as a numerical value using metric operators.

Spatial aggregations are classified by Shekar et al. [25] as distributive, algebraic, and holistic. Spatial aggregation is

supported by the MapCube [25] operator using geometric union. Moreno, Arango, and Fileto [26] extend the MapCube operator enabling spatial aggregate functions other than geometric union. They extend the operator in order to support simultaneous spatial aggregate functions. Bimonte, Villanova-Olivier, and Gensel [27] extend the logical multidimensional model GeoCube, decomposing alphanumeric aggregation functions into two aggregation functions and two constraints applied to them to take into account the additivity of measures as well as the dependency of spatial and alphanumeric aggregation functions.

B. Specialization of the multidimensional metamodel and transformation rules

In order to apply our approach to SDW, we need to modify our multidimensional metamodel of Figure 1, as well as the transformation rules described previously. We specialize the concepts of dimension and measure in the metamodel, and we will take into account the specificities of spatiotemporal aggregation in the mapping transformations (and, more specifically, in transformation 3 dealing with the mapping of aggregations and aggregation types).

In the multidimensional metamodel, we introduce two subtypes of the concept of dimension: spatial dimensions and temporal dimensions. These two categories are exclusive. However, these are not the only possible categories of dimensions.

Similarly, we introduce two exclusive subtypes of measure: spatial measures, and non-spatial measures. Spatial measures are subtyped into measures expressing a special situation (e.g. a GPS location), and measures representing a special quantity (distance, surface, volume...). Similarly, non-spatial measures are subtyped into stocks and flows [28]. These distinctions will be useful when determining applicable aggregation functions.

Figure 5 represents the specializations of the concepts of dimension and measure in the multidimensional metamodel. Based on these specializations, we specialize the mapping transformations as follows:

In *Transformation T1.1*, when defining a dimension, the type of this dimension is specified.

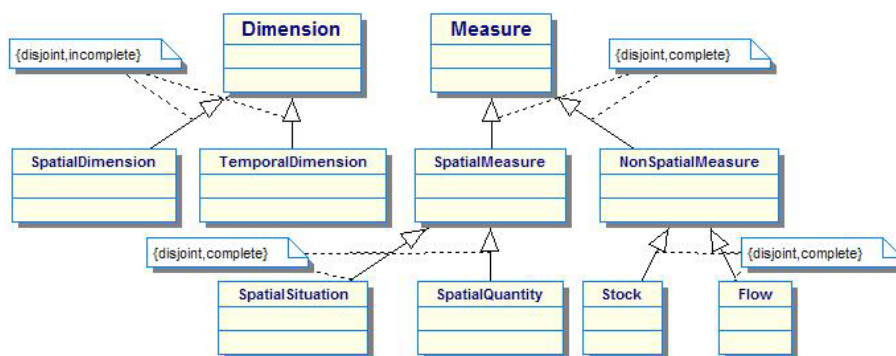


Figure 5. Spatiotemporal extensions to the multidimensional metamodel

Similarly, in *Transformation T2.2*, when defining a measure, the type of this measure is specified.

In *Transformation T3*, we add the following two rules:

-Measures of type stock are not summable along temporal dimensions [28].

-Measures expressing a spatial situation are not averageable along any dimension (in other words, they may only be counted).

C. Application to the inventory example

We now apply our mapping transformations, specialized for SDW, to the inventory example presented previously in Figure 2. Although simplified, this example contains a spatial dimension, a temporal dimension, and different categories of measures. The resulting OWL-DL ontology is represented as axioms and facts in description logic.

Transformation T1.1: Mapping of dimensions.

- $A_Dimension$
- $A_SpatialDimension \sqsubseteq A_Dimension$
- $A_TemporalDimension \sqsubseteq A_Dimension$
- $Dimension_Product \sqsubseteq A_Dimension$
- $Dimension_Warehouse \sqsubseteq A_SpatialDimension$
- $Dimension_Time \sqsubseteq A_TemporalDimension$

Transformation T1.2: Mapping of hierarchies.

- $A_Hierarchy$
- $Hierarchy_H1 \sqsubseteq A_Hierarchy$
- $dimensionToHierarchy_ProductH1$
- $T \sqsubseteq \forall dimensionToHierarchy_ProductH1.Hierarchy_H1$
- $T \sqsubseteq \forall dimensionToHierarchy_ProductH1^-.Dimension_Product$
- $Hierarchy_H2 \sqsubseteq A_Hierarchy$
- $dimensionToHierarchy_ProductH2$
- $T \sqsubseteq \forall dimensionToHierarchy_ProductH2.Hierarchy_H2$
- $T \sqsubseteq \forall dimensionToHierarchy_ProductH2^-.Dimension_Product$
- $Hierarchy_H3 \sqsubseteq A_Hierarchy$
- $dimensionToHierarchy_WarehouseH3$
- $T \sqsubseteq \forall dimensionToHierarchy_WarehouseH3.Hierarchy_H3$
- $T \sqsubseteq \forall dimensionToHierarchy_WarehouseH3^-.Dimension_Warehouse$
- $Hierarchy_H4 \sqsubseteq A_Hierarchy$
- $dimensionToHierarchy_TimeH4$
- $T \sqsubseteq \forall dimensionToHierarchy_TimeH4.Hierarchy_H4$
- $T \sqsubseteq \forall dimensionToHierarchy_TimeH4^-.Dimension_Time$

Transformation T1.3: Mapping of dimension levels.

- $A_DimensionLevel$
- $DimensionLevel_Product \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H1Product$
- $T \sqsubseteq \forall hierarchyToLevel_H1Product.DimensionLevel_Product$
- $T \sqsubseteq \forall hierarchyToLevel_H1Product^-.Hierarchy_H1$

- $DimensionLevel_SubCategory \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H1SubCategory$
- $T \sqsubseteq \forall hierarchyToLevel_H1SubCategory.DimensionLevel_SubCategory$
- $T \sqsubseteq \forall hierarchyToLevel_H1SubCategory^-.Hierarchy_H1$
- $DimensionLevel_Category \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H1Category$
- $T \sqsubseteq \forall hierarchyToLevel_H1Category.DimensionLevel_Category$
- $T \sqsubseteq \forall hierarchyToLevel_H1Category^-.Hierarchy_H1$
- $hierarchyToLevel_H2Product$
- $T \sqsubseteq \forall hierarchyToLevel_H2Product.DimensionLevel_Product$
- $T \sqsubseteq \forall hierarchyToLevel_H2Product^-.Hierarchy_H2$
- $hierarchyToLevel_H2Category$
- $T \sqsubseteq \forall hierarchyToLevel_H2Category.DimensionLevel_Category$
- $T \sqsubseteq \forall hierarchyToLevel_H2Category^-.Hierarchy_H2$
- $DimensionLevel_Warehouse \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H3Warehouse$
- $T \sqsubseteq \forall hierarchyToLevel_H3Warehouse.DimensionLevel_Warehouse$
- $T \sqsubseteq \forall hierarchyToLevel_H3Warehouse^-.Hierarchy_H3$
- $DimensionLevel_City \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H3City$
- $T \sqsubseteq \forall hierarchyToLevel_H3City.DimensionLevel_City$
- $T \sqsubseteq \forall hierarchyToLevel_H3City^-.Hierarchy_H3$
- $DimensionLevel_Region \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H3Region$
- $T \sqsubseteq \forall hierarchyToLevel_H3Region.DimensionLevel_Region$
- $T \sqsubseteq \forall hierarchyToLevel_H3Region^-.Hierarchy_H3$
- $DimensionLevel_Day \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H4Day$
- $T \sqsubseteq \forall hierarchyToLevel_H4Day.DimensionLevel_Day$
- $T \sqsubseteq \forall hierarchyToLevel_H4Day^-.Hierarchy_H4$
- $DimensionLevel_Month \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H4Month$
- $T \sqsubseteq \forall hierarchyToLevel_H4Month.DimensionLevel_Month$
- $T \sqsubseteq \forall hierarchyToLevel_H4Month^-.Hierarchy_H4$
- $DimensionLevel_Year \sqsubseteq A_DimensionLevel$
- $hierarchyToLevel_H4Year$
- $T \sqsubseteq \forall hierarchyToLevel_H4Year.DimensionLevel_Year$
- $T \sqsubseteq \forall hierarchyToLevel_H4Year^-.Hierarchy_H4$

Transformation T1.4: Mapping of dimension level attributes.

- $dimensionLevelAttribute_productCode$
- $T \sqsubseteq \forall dimensionLevelAttribute_productCode^-.DimensionLevel_Product$
- $dimensionLevelAttribute_productName$
- $T \sqsubseteq \forall dimensionLevelAttribute_productName^-.DimensionLevel_Product$
- $dimensionLevelAttribute_subCategoryName$
- $T \sqsubseteq \forall dimensionLevelAttribute_subCategoryName^-.DimensionLevel_SubCategory$
- $dimensionLevelAttribute_categoryName$
- $T \sqsubseteq \forall dimensionLevelAttribute_categoryName^-.DimensionLevel_Category$
- $dimensionLevelAttribute_warehouseCode$

- $T \sqsubseteq \forall \text{dimensionLevelAttribute_warehouseCode}^-$.
DimensionLevel_Warehouse
- *dimensionLevelAttribute_cityName*
- $T \sqsubseteq \forall \text{dimensionLevelAttribute_cityName}^-$.
DimensionLevel_City
- *dimensionLevelAttribute_regionName*
- $T \sqsubseteq \forall \text{dimensionLevelAttribute_regionName}^-$.
DimensionLevel_Region
- *dimensionLevelAttribute_day*
- $T \sqsubseteq \forall \text{dimensionLevelAttribute_day}^-$.
DimensionLevel_Day
- *dimensionLevelAttribute_month*
- $T \sqsubseteq \forall \text{dimensionLevelAttribute_month}^-$.
DimensionLevel_Month
- *dimensionLevelAttribute_year*
- $T \sqsubseteq \forall \text{dimensionLevelAttribute_year}^-$.
DimensionLevel_Year

Transformation T1.5: Mapping of rollups.

- $\text{rollup} \equiv \text{drilldown}^-$
- $\text{DimensionLevel_Product} \sqsubseteq \exists \text{rollup}.\text{DimensionLevel_SubCategory}$
- $\text{DimensionLevel_SubCategory} \sqsubseteq \exists \text{drilldown}.\text{DimensionLevel_Product}$
- $\text{DimensionLevel_SubCategory} \sqsubseteq \exists \text{rollup}.\text{DimensionLevel_Category}$
- $\text{DimensionLevel_Category} \sqsubseteq \exists \text{drilldown}.\text{DimensionLevel_SubCategory}$
- $\text{DimensionLevel_Product} \sqsubseteq \exists \text{rollup}.\text{DimensionLevel_Category}$
- $\text{DimensionLevel_Category} \sqsubseteq \exists \text{drilldown}.\text{DimensionLevel_Product}$
- $\text{DimensionLevel_Warehouse} \sqsubseteq \exists \text{rollup}.\text{DimensionLevel_City}$
- $\text{DimensionLevel_City} \sqsubseteq \exists \text{drilldown}.\text{DimensionLevel_Warehouse}$
- $\text{DimensionLevel_City} \sqsubseteq \exists \text{rollup}.\text{DimensionLevel_Region}$
- $\text{DimensionLevel_Region} \sqsubseteq \exists \text{drilldown}.\text{DimensionLevel_City}$
- $\text{DimensionLevel_Day} \sqsubseteq \exists \text{rollup}.\text{DimensionLevel_Month}$
- $\text{DimensionLevel_Month} \sqsubseteq \exists \text{drilldown}.\text{DimensionLevel_Day}$
- $\text{DimensionLevel_Month} \sqsubseteq \exists \text{rollup}.\text{DimensionLevel_Year}$
- $\text{DimensionLevel_Year} \sqsubseteq \exists \text{drilldown}.\text{DimensionLevel_Month}$

Transformation T1.6: Mapping of rollup multiplicities.

- $\text{DimensionLevel_Product} \sqsubseteq \geq 0$
 $\text{rollup}.\text{DimensionLevel_SubCategory}$
- $\text{DimensionLevel_Product} \sqsubseteq \leq 1$
 $\text{rollup}.\text{DimensionLevel_SubCategory}$
- $\text{DimensionLevel_SubCategory} \sqsubseteq \geq 1$
 $\text{drilldown}.\text{DimensionLevel_Product}$
- $\text{DimensionLevel_SubCategory} \sqsubseteq = 1$
 $\text{rollup}.\text{DimensionLevel_Category}$
- $\text{DimensionLevel_Category} \sqsubseteq \geq 0$
 $\text{drilldown}.\text{DimensionLevel_SubCategory}$
- $\text{DimensionLevel_Product} \sqsubseteq = 1$
 $\text{rollup}.\text{DimensionLevel_Category}$
- $\text{DimensionLevel_Category} \sqsubseteq \geq 1$
 $\text{drilldown}.\text{DimensionLevel_Product}$
- $\text{DimensionLevel_Warehouse} \sqsubseteq = 1$

- $\text{rollup}.\text{DimensionLevel_City}$
- $\text{DimensionLevel_City} \sqsubseteq \geq 1$
 $\text{drilldown}.\text{DimensionLevel_Warehouse}$
- $\text{DimensionLevel_City} \sqsubseteq = 1$
 $\text{rollup}.\text{DimensionLevel_Region}$
- $\text{DimensionLevel_Region} \sqsubseteq \geq 1$
 $\text{drilldown}.\text{DimensionLevel_City}$
- $\text{DimensionLevel_Day} \sqsubseteq = 1$
 $\text{rollup}.\text{DimensionLevel_Month}$
- $\text{DimensionLevel_Month} \sqsubseteq \geq 1$
 $\text{drilldown}.\text{DimensionLevel_Day}$
- $\text{DimensionLevel_Month} \sqsubseteq = 1$
 $\text{rollup}.\text{DimensionLevel_Year}$
- $\text{DimensionLevel_Year} \sqsubseteq \geq 1$
 $\text{drilldown}.\text{DimensionLevel_Month}$

Transformation T1.7: Special dimension level All.

- $\text{All} \sqsubseteq \text{A_DimensionLevel}$
- $\text{All} \equiv \{\text{all}\}$
- $\text{DimensionLevel_Category} \sqsubseteq \forall \text{rollup}.\text{All}$
- $\text{DimensionLevel_Region} \sqsubseteq \forall \text{rollup}.\text{All}$
- $\text{DimensionLevel_Year} \sqsubseteq \forall \text{rollup}.\text{All}$
- $\text{A_TerminalDimensionLevel} \equiv \text{A_DimensionLevel} \sqcap (\text{A_DimensionLevel} \sqsubseteq \forall \text{rollup}.\text{All})$
- $\text{A_TerminalDimensionlevel} \sqsubseteq \text{rollup}:\text{all}$

Transformation T2.1: Mapping of facts.

- A_Fact
- $\text{Fact_Inventory} \sqsubseteq \text{A_Fact}$

Transformation T2.2: Mapping of measures.

- A_Measure
- $\text{A_SpatialMeasure} \sqsubseteq \text{A_Measure}$
- $\text{A_NonSpatialMeasure} \sqsubseteq \text{A_Measure}$
- $\text{A_SpatialSituation} \sqsubseteq \text{A_SpatialMeasure}$
- $\text{A_SpacialQuantity} \sqsubseteq \text{A_SpatialMeasure}$
- $\text{A_Stock} \sqsubseteq \text{A_NonSpatialMeasure}$
- $\text{A_Flow} \sqsubseteq \text{A_NonSpatialMeasure}$
- $\text{Measure_NumberOfUnits} \sqsubseteq \text{A_Stock}$
- $\text{factToMeasure_InventoryNumberOfUnits}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryNumberOfUnits}.\text{Measure_NumberOfUnits}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryNumberOfUnits}^-.\text{Fact_Inventory}$
- $\text{Measure_InventoryValue} \sqsubseteq \text{A_Stock}$
- $\text{factToMeasure_InventoryInventoryValue}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryInventoryValue}.\text{Measure_InventoryValue}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryInventoryValue}^-.\text{Fact_Inventory}$
- $\text{Measure_StorageCost} \sqsubseteq \text{A_Flow}$
- $\text{factToMeasure_InventoryStorageCost}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryStorageCost}.\text{Measure_StorageCost}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryStorageCost}^-.\text{Fact_Inventory}$
- $\text{Measure_Location} \sqsubseteq \text{A_SpatialSituation}$
- $\text{factToMeasure_InventoryLocation}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryLocation}.\text{Measure_Location}$
- $T \sqsubseteq \forall \text{factToMeasure_InventoryLocation}^-.$

Fact_Inventory

Transformation T2.3: Mapping of dimensionings.

- *dimensionedBy*
- $Fact_Inventory \sqsubseteq \exists dimensionedBy. Dimension_Product$
- $Fact_Inventory \sqsubseteq= 1 dimensionedBy. Dimension_Product$
- $Fact_Inventory \sqsubseteq \exists dimensionedBy. Dimension_Warehouse$
- $Fact_Inventory \sqsubseteq= 1 dimensionedBy. Dimension_Warehouse$
- $Fact_Inventory \sqsubseteq \exists dimensionedBy. Dimension_Time$
- $Fact_Inventory \sqsubseteq= 1 dimensionedBy. Dimension_Time$

Transformation T3: Mapping of aggregation functions and aggregation types.

- *summable*
- *averageable*
- *minable*
- *maxable*
- *countable*
- $minable \equiv averageable$
- $maxable \equiv averageable$
- $summable \sqsubseteq averageable$
- $averageable \sqsubseteq countable$
- $Measure_NumberOfUnits \sqsubseteq (\exists countable. Dimension_Product) \sqcap \neg(\exists averageable. Dimension_Product)$
- $Measure_NumberOfUnits \sqsubseteq \exists summable. Dimension_Warehouse$

- $Measure_NumberOfUnits \sqsubseteq (\exists averageable. Dimension_Time) \sqcap \neg(\exists summable. Dimension_Time)$
- $Measure_InventoryValue \sqsubseteq \exists summable. Dimension_Product$
- $Measure_InventoryValue \sqsubseteq \exists summable. Dimension_Warehouse$
- $Measure_InventoryValue \sqsubseteq (\exists averageable. Dimension_Time) \sqcap \neg(\exists summable. Dimension_Time)$
- $Measure_StorageCost \sqsubseteq \exists summable. Dimension_Product$
- $Measure_StorageCost \sqsubseteq \exists summable. Dimension_Warehouse$
- $Measure_StorageCost \sqsubseteq \exists summable. Dimension_Time$
- $Measure_Location \sqsubseteq (\exists countable. Dimension_Product) \sqcap \neg(\exists averageable. Dimension_Product)$
- $Measure_Location \sqsubseteq (\exists countable. Dimension_Warehouse) \sqcap \neg(\exists averageable. Dimension_Warehouse)$
- $Measure_Location \sqsubseteq (\exists countable. Dimension_Time) \sqcap \neg(\exists averageable. Dimension_Time)$
- $A_Stock \sqsubseteq \neg(\exists summable. A_TemporalDimension)$
- $A_SpatialSituation \sqsubseteq \neg(\exists averageable. A_Dimension)$

D. Implementation in Protégé

In order to test our approach and the inferencing possibilities offered by ontologies, we implemented the ontology of the inventory example (resulting from the transformations applied in the previous section). We chose the Protégé ontology tool (www.protege.stanford.edu/), coupled with the Pellet reasoner (www.clarkparsia.com/pellet/protege/). Figure 6 shows some of the classes in the resulting ontology.

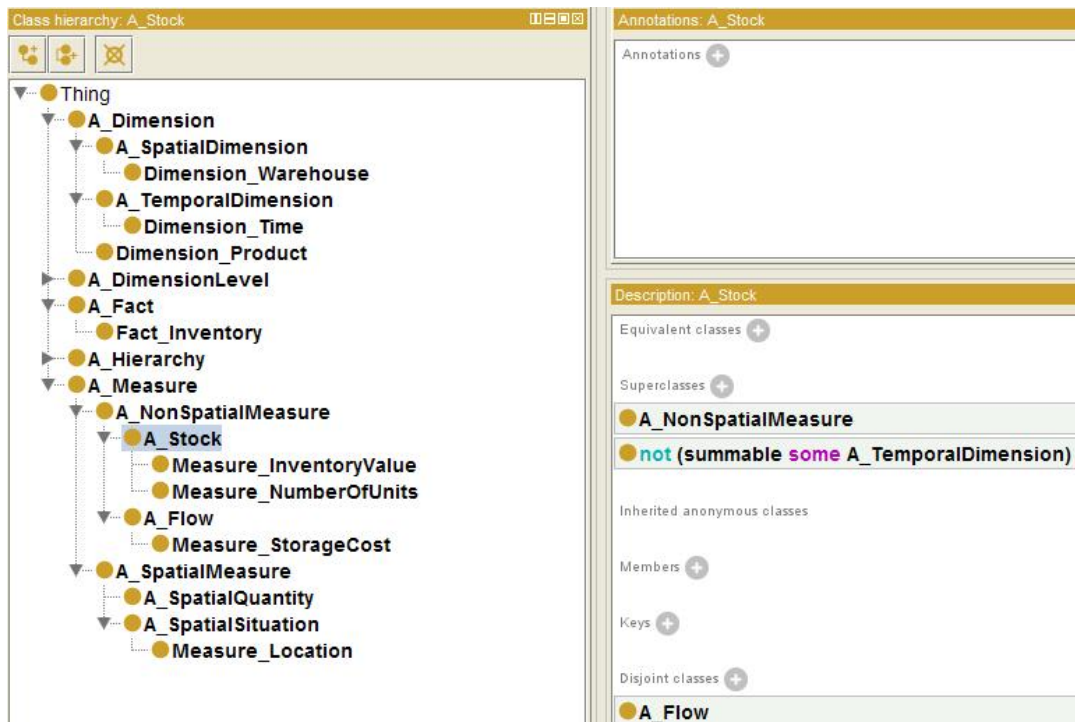


Figure 6. Excerpt of OWL-DL ontology for the inventory example

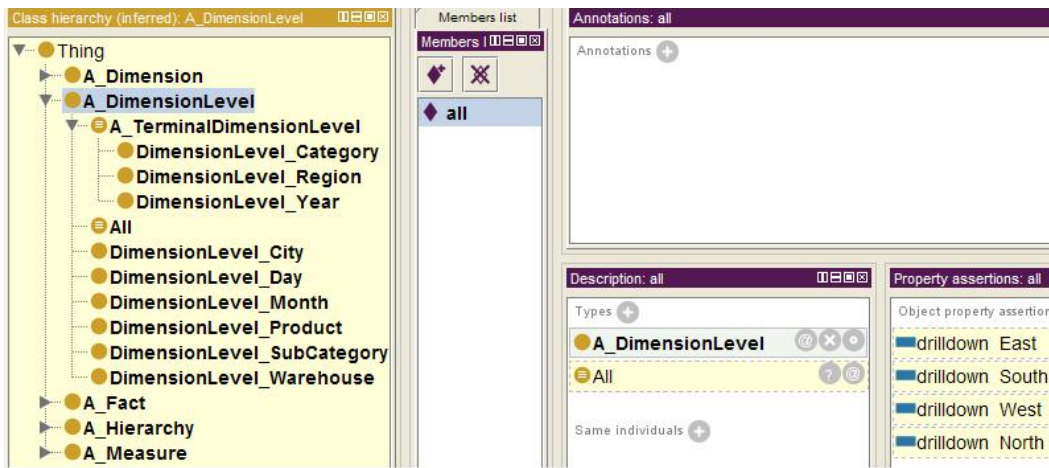


Figure 7. Inferences on the OWL-DL ontology

OWL-DL facilitates inferencing at the type level (classes and properties) as well as at the instance level (individuals). In particular, the reasoning capabilities may be used for the following purposes (see also [29]):

- *consistency checking* (e.g. by checking if some axioms contradict each other)
- *class subsumption*, and more generally *classification* (of classes and individuals)
- *class equivalences* (determining what classes are equivalent to each other)
- and, more generally, inferencing of *implicit consequences*.

In our case, consistency checking performed with the Pellet reasoner indicates that the ontology is consistent.

Figure 7 illustrates some of the inferencing enabled by the OWL-DL ontology. In this figure, the class hierarchy inferred

by the reasoner is shown. The reasoner has determined that the dimension levels Category, Region and Year are terminal dimension levels. As shown in Figure 7, it has also inferred that “all” (instance of the dimension level All) drills down to North, South, East and West. North, South, East and West were defined as individuals (instances of the dimension level Region), and the reasoner combined this information with the axioms and facts defined as a result of mapping transformation T1.7.

Figure 8 illustrates consistency checking. In this example, we assume that the measure “location” has been erroneously characterized as averageable along the dimension Warehouse. When checking consistency, the reasoner identified a contradiction with the axiom that says that spatial-situation measures are not averageable along any dimension (transformation T3). Consequently, an error is signaled for the measure “location”.

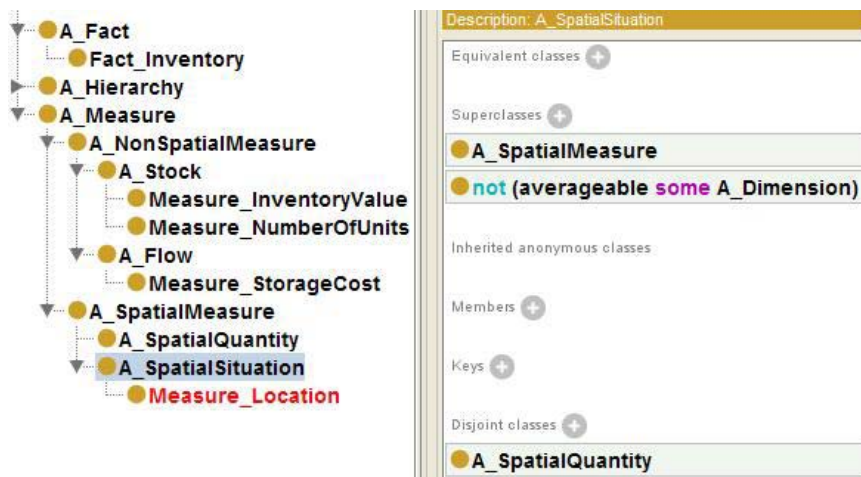


Figure 8. Inconsistency detection

VI. DISCUSSION AND CONCLUSION

In this paper, we have proposed an approach for representing a multidimensional model as an OWL-DL ontology, thus filling a gap in the literature. To achieve this goal, we have proposed a set of mapping transformations. The transformations map a multidimensional model (represented as an instance of the conceptual multidimensional metamodel) into an OWL-DL ontology (represented as an instance of the OWL-DL language). The OWL-DL ontology is a collection of axioms and facts, represented with description logic. We have specialized our approach to the case of spatiotemporal data warehouses, and illustrated our transformations with an example. We have implemented the ontology resulting from the transformations, using the Protégé ontology tool, coupled with the Pellet reasoner. OWL-DL facilitates inferencing and enables a concise and formal representation of knowledge. Transformation T3 illustrates inferencing possibilities offered by description logic; transformation T1.7 illustrates that OWL-DL nicely represents knowledge that, in the multidimensional model, is not easy to express (e.g. representing the fact that a type has a single specified instance) and is often left implicit. The example and implementation also illustrate inferencing and inconsistency-detection possibilities provided by description logic.

We acknowledge that some limitations have to be imposed on the OWL-DL language, in order for description logic to remain decidable. In particular, it is not possible, with this language, to express a maximum cardinality constraint of N (without specifying the value of N), as customary in entity-relationship or UML class diagrams. We also acknowledge that further experimental evaluation would be needed to assess more fully the correctness of our approach, as well as its scalability.

In future research, we plan to refine our mapping transformations, based on special semantics of measures and dimensions, and the cardinalities of rollups. These refinements will allow more powerful inferencing and consistency checking on the OWL-DL ontology. We will also conduct further experimentation, to assess most precisely the benefits and costs (including overhead) of our approach. The ontology may also be used to represent OLAP cubes on the semantic Web (with RDF), by defining these cubes as instances of the OWL-DL multidimensional ontology. Finally, we plan to develop an architecture enabling decision makers to query semantic Web data through the familiar OLAP view. The transformation rules presented in this paper are a first step in this direction.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their helpful comments.

REFERENCES

- [1] S. Chaudhuri, U. Dayal, and V. Narasayya, "An overview of business intelligence technology", *Communications of the ACM*, vol. 54, no. 8, August 2011, pp. 88-98.
- [2] W3C, "RDF Primer", <http://www.w3.org/TR/rdf-primer/>, February 2004.
- [3] G. Antoniou and F. van Harmelen, "Web Ontology Language: OWL", in S. Staab and R. Studer (eds.), *Handbook on Ontologies*, Springer, 2009, pp. 91-110.
- [4] C. Bizer, T. Heath and T. Berners-Lee, "Linked data – the story so far", *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, 2009, pp. 1-22.
- [5] G. Davis, P. Gray, S. Madnick, J. Nunamaker, R. Sprague, and A. Whinston, "Ideas for the future of the IS field", *ACM Transactions on Management Information Systems*, vol. 1, no. 1, December 2010.
- [6] B. Grosz, I. Horrocks, R. Volz, and S. Decker, "Description logic programs: combining logic programs with description logic", *Proc. of WWW 2003*, Budapest, Hungary, May 2003.
- [7] T. Niemi and M. Niinimäki, "Ontologies and summarizability in OLAP", *Proc. of SAC'10*, Sierre, Switzerland, March 2010.
- [8] V. Nebot, R. Berlanga, J. Pérez, M. Aramburu, and T. Pedersen, "Multidimensional integrated ontologies: a framework for designing semantic data warehouses", *Journal on Data Semantics XIII*, 2009, pp. 1-36.
- [9] G. Xie, Y. Yang, S. Liu, Z. Qiu, Y. Pan, and X. Zhou, "EIAW: towards a business-friendly data warehouse using semantic Web technologies", *Proc. of ISWC/ASWC 2007*, Busan, Korea, November 2007.
- [10] I. Astrova, "Rules for mapping SQL relational databases to OWL ontologies", *Proc. of MTSR 2007*, Corfu Island, Greece, October 2007.
- [11] G. Bumans and K. Cerans, "RDB2OWL: a practical approach for transforming RDB data into RDF/OWL", *Proc. of I-SEMANTICS 2010*, Graz, Austria, September 2010.
- [12] Z. Qu and S. Tang, "Research on transforming relational database into enriched ontology", *Proc. of ICACTE'08*, Phuket, Thailand, December 2008.
- [13] S.H. Tirmizi, J. Sequeda, and D. Miranker, "Translating SQL applications to the semantic Web", *Proc. of DEXA 2008*, Turin, Italy, September 2008.
- [14] N. Prat, I. Comyn-Wattiau, and J. Akoka, "Combining objects with rules to represent aggregation knowledge in data warehouse and OLAP systems", *Data & Knowledge Engineering*, vol. 70, no. 8, August 2011, pp. 732-752.
- [15] Object Management Group, *OMG Formal Specifications*, <http://www.omg.org/spec/>.
- [16] T. Pedersen and C. Jensen, "Multidimensional data modeling for complex data", *Proc. of ICDE 1999*, Sydney, Australia, March 1999.
- [17] N. Prat, J. Akoka, and I. Comyn-Wattiau, "A UML-based data warehouse design method", *Decision Support Systems*, vol. 42, no. 3, December 2006, pp. 1449-1473.
- [18] A. Tripathy, L. Mishra, and P. Patra, "A multi dimensional design framework for querying spatial data using concept lattice", *Proc. of IACC 2010*, Patiala, India, February 2010.
- [19] L. Gomez, S. Haesevoets, B. Kuijpers, and A. Vaisman, "Spatial aggregation: data model and implementation", *Information Systems*, vol. 34, no. 6, September 2009, pp. 551-576.
- [20] S. Rivest, Y. Bédard, and P. Marchand, "Toward better support for spatial decision making : defining the characteristics of Spatial On-Line Analytical Processing (SOLAP)", *Geomatica*, vol. 55, no. 4, 2001, pp. 539-555.
- [21] Y. Bédard, S. Rivest, and M. Proulx, "Spatial On-Line Analytical Processing (SOLAP): concepts, architectures and solutions from a geomatics engineering perspective", in R. Wrembel and C. Kocilica (eds.), *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, IRM Press 2007, pp. 298-319.
- [22] E. Malinowski and E. Zymani, "Representing spatiality in a conceptual multidimensional model", in *Proc. of GIS'04*, Washington, DC, USA, November 2004, pp. 12-22.
- [23] J. da Silva, V. Times, and A.C. Salgado, "An open source and Web based framework for geographic and multidimensional processing", *Proc. of SAC'06*, Dijon, France, April 2006, pp.63-67.
- [24] K. Bouilil, S. Bimonte, and F. Pinet, "Un modèle UML et des contraintes OCL pour les entrepôts de données spatiales : de la représentation conceptuelle à l'implémentation", *Ingénierie des Systèmes d'Information*, vol.16, no. 6, 2011.

- [25] S. Shekar, C. Lu, X. Tan, S. Chang, and R. Vastrai, "Map Cube : a visualization tool for Spatial data warehouses", in H. Miller and J. Han (eds.), *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis, London, 2001, pp. 74-90.
- [26] F. Moreno, F. Arango, and R. Fileto, "Extending the map cube operator with multiple spatial aggregate functions and map overlay", *Proc. of Geoinformatics 2009*, Fairfax, VA, USA, August 2009.
- [27] S. Bimonte, M. Villanova-Olivier, and J. Gensel, " A multidimensional model for correct aggregation of geographic measures", in P.N. San-Banto Furtado (ed.), *Evolving Application Domains of Data Warehousing and Mining*, Information Science Reference, N.Y., 2010.
- [28] H.J. Lenz and A. Shoshani, "Summarizability in OLAP and statistical data bases", *Proc. of SSDBM 1997*, Olympia, Washington, USA, August 1997.
- [29] D. Berardi, D. Calvanese, and G. de Giacomo, "Reasoning on UML class diagrams", *Artificial Intelligence*, vol. 168, no. 1-2, October 2005, pp. 70-118.