Towards an Ontology of Multidimensional Data Structures for Analytical Purposes

Christian Kurze, Peter Gluchowski, Martin Böhringer
Chemnitz University of Technology, Germany
{christian.kurze | peter.gluchowski | martin.boehringer}@wirtschaft.tu-chemnitz.de

Abstract
Multidimensional data are the foundation for OLAP applications. They can be provided in several ways: relational OLAP, multidimensional OLAP, or hybrid OLAP. The usage of the underlying technology, which is well understood and in most cases formally defined, does not resolve the issue of a missing vocabulary for multidimensional data on a conceptual level. Some basic definitions are broadly used; for example cube, dimension, and the operations like slice, dice, roll-up, and drill-down. When it comes to more sophisticated constructs like irregular hierarchies, different vocabulary exists.

For the integration of different OLAP applications as well as for an easier development (each stakeholder has to use the same vocabulary in order to reduce misunderstandings and projects failures), we provide an approach for a comprehensive ontology of multidimensional data. It defines the vocabulary used during the design of OLAP and data warehouse applications. The ontology can be seen as a basis for (i) notation assessment by evaluating the notation against the Bunge–Wand–Weber model, (ii) the ontological engineering of new data warehouse and OLAP applications, and (iii) ontological model integration.

1. Introduction

During the last decade, Business Intelligence (BI) has become more and more important to different kinds of enterprises. BI is used in almost any department of a company; for example controlling, marketing, and production.

Since these systems grow over time and many heterogeneous systems have to be integrated, they are often hard to maintain if sufficient documentation is missing. Some approaches introduce the usage of ontology to overcome these problems. The authors of [3] present a hybrid framework for the ontological integration of the conceptual, the analytical, and the physical view of a BI system, which are matched against user interfaces, data warehouses, and operational information systems. As a basis for integration, there has to be an accepted upper ontology which is the foundation for the creation of application domain specific ontologies [26, 31]. As outlined in later sections, there is no such general OLAP ontology by now. We aim to close this gap with our ontology but are aware that this is research in progress and still has to be evaluated by a large community of researchers and practitioners.

The approach of the Common Warehouse Metamodel (CWM), which tried to close this gap, is too generic to support all peculiarities of multidimensional data modeling and too complex to be understood by both end-users and developers [27]. Therefore, the multidimensional and OLAP package of the CWM should be integrated into the ontology; but it is not satisfactory to use the CWM for a common understanding of data warehousing terms.

We create this ontology in order to define the core of the data warehousing discipline, not the whole body of knowledge, which is what the CWM tries to. The body of knowledge is characterized by the diversity and complexity of development, implementation, operation, maintenance, and management of data warehouse systems. In terms of existing literature, it is almost clear that neither a single data warehouse researcher nor a single practitioner can be competent in all relevant areas. The core of the discipline, however, may be small. The ontology defines this core of the data warehousing discipline and is therefore relevant to three areas: data warehouse teaching, research, and practice. The definition of the core of a discipline then helps to derive powerful, general theories and paradigms [45, pp. 70–71].

Moreover, we see an ontology as a possibility to enhance conceptual data warehouse modeling. Among others, the answer to complex summarizability issues is currently not supported by any toolsets [28]. We assume ontologies to be suitable to solve these issues because of their ability to conduct reasoning and inference on a semantic level.
2. Preliminaries and related work

This section outlines the preliminaries for the remainder of this paper. It discusses the topics of ontology, ontology levels, ontology engineering, and the distinction between ontology engineering and conceptual modeling. Related work is presented each time a term/concept is defined.

2.1. Ontology

In the context of computer science, the term ontology has largely come to two related things [5]: a representation vocabulary that may be specialized to a specific domain or subject matter, and a body of knowledge describing a particular domain by using a representation vocabulary. The authors of [13, pp. 50-52] summarize the four most important features of ontologies. First, they relate ontologies to vocabulary for referring to terms of a subject area. It is not the vocabulary itself that qualifies an ontology but the conceptualization its terms are intended to capture [5]. Therefore, “an ontology specifies terms with unambiguous meanings” [13, p. 52], independent of reader and context. Since the translation into other languages does not change the ontology conceptually, “an ontology provides a vocabulary and a machine-processable common understanding of the topics that the terms denote” [13, p. 52].

The second feature of an ontology is a taxonomy or concept hierarchy providing a hierarchical categorization or classification of entities within a domain. “The vocabulary and the taxonomy of an ontology together provide a conceptual framework for discussion, analysis and information retrieval of a domain. [...] With ontologies, the subclassing is strict, is formally specified, includes formal instance relationships, and ensures consistency in deduction of the ontology” [13, p. 52].

Third, ontologies identify classes of objects, relations, and concept hierarchies of a domain, so they can be seen as content theories [5]. Ontologies represent knowledge in a very structured way using specific ontology representation languages. An overview of these languages can be found in [13, p. 59]. However, ontologies clarify the structure of domain knowledge. Ontological analysis reveals the concepts of the domain knowledge, their taxonomies, and the underlying organization.

The major purpose of ontologies is not representing vocabulary and taxonomy, it is knowledge sharing and knowledge reuse by applications as their fourth feature [13, p. 52]. The captured intrinsic conceptual structure of a domain [5] can be used to create rich domain-specific knowledge representation languages for building knowledge bases in that domain.

Given that definition, it can be further distinguished between different forms the term “ontology” is used in current IS research. Fonseca [11] outlines two different meanings of the term “ontology” in IS research: Ontology (singular, capital “O”) is seen as the branch of philosophy dealing with basic descriptions of things in the world and their relationships to each other. The usage of this philosophical science in IS research is twofold: first, there are ontologies of information systems. They are seen as descriptions of information systems; Ontology is applied in order to understand the basic constructs of information systems. Its purpose is to support the development of conceptual-modeling tools. Second, there are ontologies for information systems. In this context, Ontologies are seen as part of an information system. Their purpose is the creation of conceptual schemas. Fonseca underpins his work [11] with several examples.

Another example of the distinction between ontologies of and ontologies for information systems is given in [37]. The authors’ semantic framework to support business intelligence includes both understandings of ontologies: the “Domain Ontology” provides a formal description of terminology of the business domain supported by the BI framework and thus can be seen as an ontology for information systems. The “BI Ontology” models the concepts to describe how data are organized in source systems and how to map these data to the concepts described in the “Domain Ontology”. Therefore, it is an ontology of information systems.

Our work is aligned to the “BI Ontology”, since we develop a vocabulary for the basic constructs used in BI systems, especially data-oriented components. We differ from [37] since the ontology presented there is not integrated into an upper ontology and no rigorous construction method is used to develop the ontology.

2.2. Ontology levels

In practice, knowledge sharing by the use of ontologies is not easy – even if a readily constructed ontology for that particular domain exists. Reasons may be seen in the different languages for representing ontologies, different competing approaches and working groups developing different technologies, traditions, and cultures, and different ontologies for a certain domain. One of the major problems is knowledge maintenance, since knowledge evolves over time [13, p. 55].

In order to overcome some of these problems, ontological levels provide different abstraction levels. An “Upper Ontology” is on top of this stack and
describes the most abstract and generic concepts. It can be said that they provide an ontological metamodel [34]. There are different approaches to defining these uppermost concepts. A summary can be found in [13, p. 74]. One of the most discussed works in the context of information systems is the upper ontology of Bunge, Wand, and Weber, often referred to as the BWW-Ontology, which goes back to the works of Bunge [1, 2] and has been made more public by Wand and Weber [41, 42, 43]. For a long time there has been only a description in natural English language for this upper ontology as well as two metamodels defined in the notation of eERM [36] and UML using object oriented concepts [20]. A recent work of Evermann [10] makes use of the Web Ontology Language and UML to formally define the BWW upper ontology. Since a lot of research has proven this upper ontology to be useful for the description of information systems [36], we make use of its OWL based representation based on [10]. There are other upper ontologies as well, for example [23, 24, 32, 38]. A working group has been established at IEEE in order to create a standard upper ontology [16]. However, all approaches have in common that they address a broad range of domain areas. For the purpose of our paper it is more appropriate to use BWW since it is proven to be suitable for ontological analyses of information systems.

The approach presented by Bräuer and Lochmann [26] defines a stack of ontologies which we are going to adopt here; see Figure 1. As discussed earlier, our upper ontology is taken from BWW. Technically, on top of this level there is another meta-ontology, namely OWL, which is used to describe the BWW upper ontology. Thus, all concepts of the BWW ontology are derived from owl:Thing.

![Ontology Levels Diagram](image)

**Figure 1. Different ontological levels [26]**

The main purpose of our paper is to create a Core Ontology (highlighted in Figure 1) which conceptualizes the vocabulary and taxonomy of multidimensional data modeling – this can be seen as an ontology of multidimensional information systems. For example, we will define concepts like Cube and Dimension as well as their relationship Cube has Dimensions. A particular Domain Ontology – an ontology for multidimensional information systems – would be a Sales Cube having the Dimensions Customer, Product, and Time. Application-specific ontologies are further enriched by peculiarities of the concrete application, which is out of the scope of this paper. Ontologies can be integrated by referring to concepts in upper-level ontologies [10].

### 2.3. Ontology engineering

Developing an ontology requires plenty of engineering effort, discipline, and rigor. The life cycle of an ontology is supported by a set of design principles, development processes and activities, and supporting technologies as well as systematic methodologies. They are subsumed under the term ontology engineering [13].

The ontology life cycle may be supported by different approaches; see, for example, the survey in [7]. We selected the DOGMA approach [39] since it is an integrated methodology for ontology engineering based on various scientific disciplines, in particular database semantics and natural language processing. Furthermore, DOGMA is described by a formal process and is therefore repeatable. One important reason for choosing DOGMA has been its collaborative characteristic, which is especially important for further development and refinement of our initial ontology by a large group of researchers and practitioners. The DOGMA approach consists of two main steps, but is not a mere linear one [39]:

1. **Preparatory steps:**
   a. Formulate vision statement
   b. Conduct feasibility study
   c. Project management
   d. Preparation and scoping

2. **Ontology engineering steps:**
   a. Domain conceptualization
   b. Application specification

*Formulate vision statement* summarizes a “compelling and inspiring view of a desired and possible future” [39, p. 16]. The *feasibility study* refines the vision statement and reveals whether a project is worthwhile in the sense of costs and benefits, technological feasibility, and needed (committed) resources. *Project management* can be seen as the initialization of management activities. They are ongoing during the whole capture and development process. Furthermore, a project plan should be created. The activity *preparation and scoping* defines the domain of the future ontology more sharply. Sufficient attention should be paid to this step since it determines the quality of the future ontology. We took three steps out of five in order to prepare and scope our ontology:
define purpose which states what our aim is, compile knowledge resources which gives a summary of input documents for our ontology, and scope knowledge resources in which we select relevant passages out of the source material and define scenarios which further structure the material.

The domain conceptualization step is the most crucial one since the domain is analyzed and represented in an ontology. There are several ways of collecting knowledge: knowledge elicitation (consultation of experts), knowledge breakdown (only a few authoritative experts define the knowledge), knowledge negotiation (many stakeholders in a distributed setting), and knowledge discovery (from textual sources). We make use of knowledge discovery from textual sources identified in the preparation and scoping activity as well as knowledge negotiation in future development steps by providing means of collaborative ontology development to a broad public. After collecting the knowledge there is a lexon engineering phase. A lexon is a quintuple \((\lambda, \zeta), \text{term}_1, \text{role}_1, \text{role}_2, \text{term}_2)\), where \(\zeta\) denotes the context used to group lexons that are related to each other. In our case, the context refers to a particular section of the input document. The quintuple expressed in the natural language \(\lambda\) holds in the specified context \(\zeta\) in the following way: \(\text{term}_1\) has \(\text{term}_2\) occur in \(\text{role}_1\) with it. Inversely \(\text{term}_2\) occurs in \(\text{role}_2\) with \(\text{term}_1\). They form a lexon base grouped by language and context. After the creation of lexons, they have to be refined in order to be highly reusable, as simple as possible, representing the correct information, and binary. The grounding of lexons provides clear definitions of each term and role. The creation of meta-lexons makes the grounded lexons language-neutral and context-independent.

The application specification step is also crucial, since semantic constraints on lexons are defined and an evaluation of the ontology by means of competency questions is done. Defining competency questions forms a base line reference of domain knowledge that the ontology should be able to express from an application point of view. They should be structured in a hierarchical way proceeding from general to more specific. The definition of semantic constraints outlines, for example, internal/external uniqueness, mandatory constraints, subsets, equality, and exclusion. Therefore, a reflection of the meta-lexons according to their textual resources has to be done. The validation of the ontology is done by answering competency questions. These answers form the basis of further definition of semantic constraints.

As a last step, the meta-lexons and their semantic constraints have to be implemented in an ontology language. The paper at hand makes use of the Web Ontology Language (OWL). Terms are represented by Subclasses of owl:Thing, roles by Properties, and semantic constraints by Conditions on Classes.

The approach presented in [33] also takes already existing ontologies into account. We included this step in the preparation and scoping phase. The integration into the upper ontology has been done in the refine and ground lexons steps. This activity gives further hints about what kind of terms are defined and implicitly creates new lexons based on the properties of the upper ontology’s classes.

### 2.4. Supplementing conceptual modeling with ontological engineering

As outlined in the previous sections, ontological engineering and conceptual modeling interrelate with each other. The question arising is whether it is still necessary to model on a conceptual level if there exists an ontology of and/or an ontology for information systems. Guarino [15] describes the ontology driven development of information systems. Domain Ontologies could be transformed to domain conceptual-modeling scripts which, in turn, can be further transformed into software artifacts, that is, other scripts and/or code. Existing ontologies can be used to create conceptual-modeling scripts. The work of Wand and Weber tries to evaluate conceptual-modeling grammars against Core Ontologies [42]. Therefore we can see differences in ontological engineering and conceptual modeling: ontological engineering tries to gather the knowledge of a certain domain on different levels, that is, the construction of ontologies of and for information systems. These ontologies are, in turn, used to evaluate conceptual-modeling grammars and scripts [44]. They form the basis for concrete software/system development. The main difference is that reasoning can be done over ontologies, but not over conceptual data models. A human is not able to anticipate all implications of a new restriction within a complex conceptual model; but a reasoner can be used within a formally grounded ontology. In other words, the impact of changes on single models can be automatically examined by a reasoner over the ontology. Therefore, the semantic quality (i.e. which contents should be depicted in the conceptual model) can be improved significantly by automated reasoning. Conceptual models are, in turn, an adequate way to visualize the domain conceptualization. Issues that might arise from small changes within the conceptual model concerning the summarizability of measures are discussed in [28].
3. An ontology of multidimensional data structures for analytical purposes

We would like to provide a first step towards an ontology of multidimensional data. Future research will include publishing the ontology and providing means for collaborative improvement by both the scientific and practitioners’ communities. By using the Bunge–Wand–Weber upper ontology we will provide rigorous definitions of common terms used in modeling for analytical purposes.

3.1. Upper ontology

We base our upper ontology on the works given in [10, 20, 36, 42]. They all give descriptions of Bunge’s ontology [1, 2] in a more or less formal way. The work of Evermann [10] gives a complete representation in UML and OWL of Bunge’s ontology. Because this work is very complex, we refine the upper ontology to the concepts used in [20, 36, 42]. Our work is therefore aligned to the Bunge–Wand–Weber ontology. All concepts defined later on will be child classes to this upper ontology in order to enable the integration of different ontologies via the chosen upper ontology.

3.2. Preparatory steps

As the first step in the DOGMA approach, we defined a vision statement given in Table 1. Second, we did a short feasibility study in order to outline the benefits of this ontology. A simple project management has been established. It consists of task definitions, responsibilities, and clear timeframes.

The preparation and scoping step is very important to describe the domain of the future ontology more sharply. The resulting scoping form is given in Table 2. We added an additional section to cover related ontologies. The results of our literature search are also given in Table 2.

Based on their distribution in the area of business intelligence, we have chosen two textual sources for our core ontology. Reference [6] has been cited more than 1000 times (Google Scholar on 30 May 2009) and can therefore be seen as a base for a large amount of other scientific works. Reference [19] is a source used by many practitioners and has been cited more than 350 times (Google Scholar on 30 May 2009). The selected sections introduce dimensional modeling.

The work presented in [25] provides an ontological model based on MetaCube which is a conceptual multidimensional data model. Their core ontology is integrated directly into OWL, that is, their upper ontology is OWL. The authors also provide means for instantiating a domain ontology.

Reference [30] does not focus on the dimensional model itself, that is, its structure and behavior, but on how to extract and integrate information from given BI applications. The authors provide heuristics for extracting facts (called cubes in our vocabulary), measures, and dimensions. The identified concepts are then further standardized. In summary, this approach describes the automated construction of a domain ontology.

The authors of [31] give an overview of using ontologies to describe the domain of OLAP applications. They also provide an ontology for OLAP. Unfortunately, the given ontology is not broad enough for a comprehensive description of multidimensional data structures. It only consists of four classes and is – from our point of view – strongly related to logical star schema modeling since it uses terms like FactRows which constitute OLAPCubes.

All three approaches have in common that they do not use an upper ontology which has been proven to be suitable for the description and integration of information systems. To meet this shortcoming, we make use of Bunge–Wand–Weber as an upper ontology which gives our work rigor and helps to further integrate BI application by using a common upper vocabulary. Ontological analyses based on this upper ontology are expected to have a high maturity since it has been used for more than twenty years now [14].

Table 1. Ontology vision statement [39]

<table>
<thead>
<tr>
<th>Ontology vision statement form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
</tr>
<tr>
<td>Domain</td>
</tr>
<tr>
<td>Ontologies</td>
</tr>
<tr>
<td>Applications</td>
</tr>
<tr>
<td>Technologies</td>
</tr>
</tbody>
</table>
Table 2. Ontology scoping [39]

<table>
<thead>
<tr>
<th>Ontology scoping form</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Purpose</strong></td>
<td>The purpose of this ontology is (i) to give a clear definition of vocabulary used in multidimensional data modeling for analytical purposes, (ii) to formally define the semantics of each term, and (iii) to provide a basis for ontological (semantic) integration of different BI applications.</td>
</tr>
<tr>
<td><strong>Sources</strong></td>
<td>The ontology will be built on the following sources: [6, sections 4 and 5], [19, chapter 6]; future work has to include a broad literature study of scientific as well as practitioners’ works.</td>
</tr>
<tr>
<td><strong>Related ontologies</strong></td>
<td>[25], [30], [31]</td>
</tr>
</tbody>
</table>

3.3. Ontology engineering steps

Within the domain conceptualization phase we used the approach of knowledge discovery from textual sources. Therefore, we outlined the important settings of each source and further detailed each setting in accordance to the narratological schemata presented in [39]. These detailed settings formed the base for creating lexons for each source. A shortened sample of [6] is given in Table 3. The analysis of [19] resulted in three settings (with subsettings) and 16 lexons.

The refinement step is based on a hermeneutical approach. It was provided by [12, 35] and is well accepted in IS research, for example [4, 29]. The core of this approach is a cycle that iterates over repeated interpretations of a text in different backgrounds of understanding. Generally, this approach has been used to understand parts of a text and to put these parts into the context of the whole text. The interpretation by a reader is complete when the reader’s interpretation does not change from previous readings of the text. With this theory in mind, the authors conducted several discussions until no change in the refined lexons was needed and the lexons were as simple as possible and reflected the problem area as well as possible. Six lexons have been used for integrating the terms and roles into the BWW upper ontology.

The grounding of lexons has been done by summarizing the references from which the lexons came, that is, the source text and (sub-)settings. Meta-lexons have not been created since we did not establish a connection to definitions like those given in WordNet or elsewhere. Meta-lexons should be established in the future after an extensive literature study in combination with lexon engineering. We are aware that reviewing a larger body of literature might change the structure of our ontology. This is where we see another benefit of using an ontology instead of a metamodel: the methods for ontology evolution are more mature (see for example [9, ch. 4]).

Table 3. Sample narratological schema for reference [6]

<table>
<thead>
<tr>
<th>Settings</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S1 (p. 68)</strong></td>
<td>(First half paragraph of section 4, not shown here for space reasons.)</td>
</tr>
<tr>
<td><strong>S1.1</strong></td>
<td>[…] Set of numeric measures that are the objects of analysis.</td>
</tr>
<tr>
<td><strong>S1.2</strong></td>
<td>Each of the numeric measures depends on a set of dimensions which provide the context for the measure.</td>
</tr>
<tr>
<td><strong>S1.3</strong></td>
<td>The dimensions together are assumed to uniquely determine the measure.</td>
</tr>
<tr>
<td><strong>S1.4</strong></td>
<td>Thus, the multidimensional data view a measure as a value in the multidimensional space of dimensions.</td>
</tr>
<tr>
<td>…</td>
<td>(Five more settings not shown here.)</td>
</tr>
<tr>
<td><strong>Lexons</strong></td>
<td></td>
</tr>
<tr>
<td><strong>L1</strong></td>
<td>S1.1, S1.2; numeric measure; depends on; provide context; dimension.</td>
</tr>
<tr>
<td><strong>L2</strong></td>
<td>S1.3, S1.4; numeric measure; is value in; uniquely determine; set of dimensions.</td>
</tr>
<tr>
<td>…</td>
<td>(Seventeen more lexons not shown here.)</td>
</tr>
</tbody>
</table>

3.4. Ontology overview

In order to present our ontology we follow the approach of [20] and introduce it part by part. The first category contains subclasses of Thing, Property, and State; second, the representational category with subclasses of Schema, Attribute, and StateVariable; and third, relationships. Table 4 summarizes all classes.

**Things** exist to represent that the world consists of things [20]. In our case, DimensionElements form the key objects of business [19] and are therefore the basic things in our world. Correlated DimensionElements form the CompositeThing Dimension. A Measure is
also a CompositeThing since it is clearly determined by Dimensions. It is possible to argue that Measures exist without Dimensions, but in our context this is not useful, since the determining Dimensions of a Measure are important. The same holds for Cube, since it consists of Measures and Dimensions.

We identified Attributes as Properties of DimensionElements. The ComplexProperty HierarchyLevel consists of different basic Attributes (e.g. a key Attribute as well as a descriptive Attribute) and is also a Property itself. The precedence relationship between Attributes is used to characterize hierarchies within the Attributes (parent-child relationships). A precedence states that one Property must hold in order for the succeeding Properties to hold [20]. For example, a product must be contained in a subcategory (the same subcategory-Attribute as other products) in order to be in some category (again the same category-Attribute as other products).

A Fact constitutes the State of a Measure. In other words, Facts are the intersections of the multidimensional array indices of a Cube and therefore represent concrete values. In this way we express that a Measure changes in the sense of filtering along the Dimensions and has different values if the combination of DimensionElements changes. The Fact is determined by the selected instance values of DimensionElements.

The representation of mdm:Attributes (child class of bww:Property) is done via mdm:AttributeValue which is a child of bww:Attribute (consider the same spaces). Since mdm:HierarchyLevel is a child of mdm:Attribute, no further representational classes are needed.

Things are represented in different Schemata. If a ROLAP approach is used, then Things are represented via a RelationalSchema; in case of a MOLAP approach, via a MultidimensionalSchema. At this point we see the integrating characteristic of our ontology. Different source and target systems can be linked to the integrating characteristic of our ontology. Dimensions, Measures, and Cubes. For example, a dimension can be represented by a relational table or, for instance, by an XML file describing structures in a multidimensional OLAP server. Other sources can be subclassed in the domain ontology.

A StateVariable for representing a Fact is not given in our ontology, since this is a function which filters and groups Attributes along the determining Dimensions. A subclass should be created in a domain ontology.

Events exist to represent the change from one State into another. Within our sources we found ten operations shown in Table 4 as subclasses of Event.

### Table 4. Classes in the ontology of multidimensional data structures for analytical purposes

<table>
<thead>
<tr>
<th>BWV</th>
<th>MDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thing</td>
<td>DimensionElement</td>
</tr>
<tr>
<td>CompositeThing</td>
<td>Cube</td>
</tr>
<tr>
<td>Property</td>
<td>Attribute</td>
</tr>
<tr>
<td>ComplexProperty</td>
<td>HierarchyLevel</td>
</tr>
<tr>
<td>State</td>
<td>Fact</td>
</tr>
<tr>
<td>Attribute</td>
<td>AttributeValue</td>
</tr>
<tr>
<td>Schema</td>
<td>MultidimensionalSchema</td>
</tr>
<tr>
<td>Event</td>
<td>Aggregation</td>
</tr>
<tr>
<td></td>
<td>Comparison</td>
</tr>
<tr>
<td></td>
<td>ComputingAttributes</td>
</tr>
<tr>
<td></td>
<td>ComputingMeasures</td>
</tr>
<tr>
<td></td>
<td>DrillDown</td>
</tr>
<tr>
<td></td>
<td>Pivoting</td>
</tr>
<tr>
<td></td>
<td>Ranking</td>
</tr>
<tr>
<td></td>
<td>RollUp</td>
</tr>
<tr>
<td></td>
<td>Selection</td>
</tr>
<tr>
<td></td>
<td>SliceAndDice</td>
</tr>
</tbody>
</table>

The relationships between subclasses of Thing inherit from the CompositeThing hasAssociatedThing* Thing relationship: Dimension hasDimension-Elements* DimensionElement, Cube hasDimensions* Dimension, Cube hasMeasures* Measure, Measure isDeterminedBy* Dimension. The precedence relationship between Attributes representing the parent-child order between HierarchyLevels has already been explained. All other relationships are inherited from the BWV upper ontology.

3.5. Application specification

This section outlines our competency questions for evaluating the ontology. Answers are given and open issues are addressed.

3.5.1. Defining competency questions. The definition of competency questions is crucial for the evaluation of the created ontology. The questions presented here arose from a project with an industry partner of the telecommunications industry. Within this project, a repository-based modeling tool has been developed [21, 22]. The following statements should be answerable in accordance with the ontology vision and ontology scope defined in section 3.2:

1. Are the static peculiarities of multidimensional data modeling for analytical purposes defined?
   a. Which dimensions are used in which cubes and vice versa?
b. Which cubes make use of which measures in which combination of dimensions?

c. Are there any cubes with a similar structure; do they differ in only some dimensions?

d. Which attributes are related to dimension elements or groups of dimension elements?

e. Is it possible to retain different versions of structural elements and their relationships by means of historicization?

2. Are the dynamic peculiarities of multidimensional data modeling for analytical purposes defined?

a. Are the operations defined?

d. How to deal with changing dimensions and calculation rules between measures over time?

Since the questions have been taken from one single project, they limit our work in the sense of generalizability. But, on the other hand, they prove a strong practical relevance of our outcomes.

3.6. Applying an example

The example given in this section shows how to apply the ontology. It is based on the sample database Adventure Works DW of Microsoft SQL Server. We integrate the Internet Sales Cube into our ontology. Due to space limitations we outline only two measures and the basic dimensions. The customer dimension is taken as an example for further explanation.

An additional business value can be seen in the formal description of multidimensional data structures independent of any particular system. A business user might look up the information in question within the ontology and can then link this information to database tables in SQL Server and multidimensional concepts in Analysis Services. Similarly, developers are supported in the same way: they can search for data objects in business terms (as provided by the users) within the ontology and link them to concrete database objects.

We created four subclasses of mdm:Dimension: DimCustomer, DimDate, DimProduct, and DimSalesTerritory. They are represented by subclasses of mdm:RelationalSchema named after their relational table names, for example, advdw:dbo.DimDate. All Dimensions consist of their appropriate DimensionElements. The customer dimension uses two relational tables; thus, there have to be two classes. Their multidimensional representation in MS Analysis Serviced can be found in the project definition file Customer.dim. The relationship between dimensions and their schemata is done via a restriction, for example bww:hasRepresentationSchema only (advdw:dbo. DimCustomer or advdw:dbo.Dim- Geography or advdw:Customer.dim) for the customer dimension.

Accordingly, we created two subclasses of mdm:Measure: MSalesAmount and MUnitPrice. Their state is represented by two mdm: Facts: FSalesAmount and FUnitPrice. Again, restrictions further specify the Dimensions determining both Measures. Together, the created Dimensions and Measures constitute the Cube InternetSales which is again formally expressed via restrictions.

Each attribute of the customer dimension is represented as a subclass of mdm:Attribute or mdm:HierarchyLevel depending on its usage. Their representation is given via classes representing their attributes in the underlying relational tables. Here one clearly sees that each bww:Property can be represented via more than one bww:Attribute. For example, the customer’s city is represented via two key columns (Geography.City, Geography.StateProvinceCode) and a name column (Geography.City). The hierarchical relationships are represented via precedence between
4. Conclusions and further research

Our paper introduced an ontology of multidimensional data for analytical purposes. The approach is rigorous since the ontology is based on an accepted upper ontology of information systems and a proven engineering approach has been used. The work is still research in progress and has to be further developed by researchers and practitioners. We propose a collaborative approach and will provide technical means. Please contact the authors to contribute.

Our ontology is one part of an ontological integration of BI systems. The multidimensional concepts can be identified in different types of systems. A further integration and mapping to business ontologies helps to identify the concept in the whole heterogeneous system environment. Imagine the concept product. If a business ontology and the presented ontology were integrated, finding the dimension element product in different BI components would be possible. This scenario can be seen as a model trace between the ontology and the underlying systems [8]. The integration with a business ontology also shows where to find products in transactional systems. Both identification mechanisms form the basis for an ontology-based creation of extract-transform-load processes. This process tries to map the related ontologies and to (semi-) automatically create transformation rules. Such a mapping approach may be implemented based on the ideas of [40].

Further research efforts have to include a broader literature study in order to ground the concepts of the ontology. There are several modeling aspects not covered in our ontology like temporal aspects (slowly/rapidly changing dimensions) or irregular hierarchy structures. Scientific as well as practitioners’ literature has to be analyzed. Therefore, the proposed DOGMA approach should be used and supported by appropriate software systems.

A second important research area is the formal representation of the Bunge–Wand–Weber ontology. For the purpose of this paper we modeled this upper ontology in OWL in accordance with the sources available and an OWL ontology of the underlying works of Bunge. This upper ontology also has to be further refined and extended.

After this step, the concept hierarchy should be critically analyzed in a collaborative way. The issue is whether all concepts defined in our ontology are classified correctly. For example, is HierarchyLevel a ComplexProperty or a Kind of DimensionElements with characteristicProperties constituting the HierarchyLevel? Such issues can only be answered in a collaborative way by finding consensus. The refinement should also include the formal definition of all operations. This enables reasoning over whether states’ measures, that is, facts, are lawful or not as well as the derivation of statements about their summarizability [28].

5. References


