## Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2009 Proceedings

Americas Conference on Information Systems (AMCIS)

2009

# Towards Principles for Structuring and Managing Very Large Semantic Multidimensional Data Models

Christian Kurze Technische Universitat Chemnitz-Zwickau, christian.kurze@wirtschaft.tu

Peter Gluchowski *Technische Universitat Chemnitz-Zwickau*, peter.gluchowski@wirtschaft.tu-chemnitz.de

Follow this and additional works at: http://aisel.aisnet.org/amcis2009

#### **Recommended** Citation

Kurze, Christian and Gluchowski, Peter, "Towards Principles for Structuring and Managing Very Large Semantic Multidimensional Data Models" (2009). *AMCIS 2009 Proceedings*. 315. http://aisel.aisnet.org/amcis2009/315

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

## **Towards Principles for Structuring and Managing Very Large Semantic Multidimensional Data Models**

### Christian Kurze, Peter Gluchowski

Chemnitz University of Technology, Thueringer Weg 7, D-09126 Chemnitz, Germany {christian.kurze | peter.gluchowski}@wirtschaft.tu-chemnitz.de

#### ABSTRACT

The management of semantic multidimensional data models plays an important role during the phases of development and maintenance of data warehouse systems. Unfortunately, this is not done with the necessary stress by now. Reasons might be seen in the plethora of semantic notations or the insufficient tool support for multidimensional modeling. The paper on hand provides experiences gained within a project with an industry partner of the telecommunications industry. Their problem is a very huge data warehouse with more than 400 data cubes and several hundred key performance indicators. We developed a repository-based solution for managing the semantic data models. Our lessons learned show that especially for very large data models there has to be a repository based solution as well as a clear concept on how to break them up into their component pars. The aim of our principles is to increase the understandability as well as the maintainability of semantic multidimensional data models.

#### Keywords

Application Design for Analytical Processing Technologies, ADAPT, multidimensional data modeling, semantic data model, repository, model management

#### INTRODUCTION

This paper has been inspired by an industry partner from the telecommunications industry. They are presently redesigning their data warehouse solution. Due to more than 400 implemented data cubes, they decided to establish a semantic modeling approach in order to reduce redundancy and inconsistency. This solution helps to manage user requirements on a semantic level, i.e. a management of semantic multidimensional data models. To cope with this enterprise-wide initiative, a new department will be established. This department is responsible for gathering user requirements, check their feasibility, model the requirements in alignment to already existing structures, and handling the specified requirements on to the IT department which is responsible for implementing the new solution.

The basis of this idea is an underlying repository, which stores all semantic data objects and associations between them and thus facilitates analyses on the data models. Questions like "Which cubes are affected if we change our customer dimension?" can be answered. Furthermore, precise definitions of measures, dimensions, cubes, etc. are given. The repository will be the single point of truth for multidimensional data structures within the whole enterprise. Our practical experience has shown that a simple structuring advice of (Bulos and Forsman, 2006) is not deep enough for very large diagrams. That is the reason for deducing principles for breaking up large diagrams into its component parts in order to increase understandability and maintainability.

During the development of data warehouses numerous steps must be taken to accomplish a highly productive realization of user requirements. Therefore, the principles and its underlying tool support outlined in the present paper should be aligned into a development framework for data warehouses presented by (Kimbal, Ross, Thorntwaite, Mundy and Recker, 2008). According to these authors, one of the most important steps is *Business Requirements Definition*. Subsequently, there are three parallel tracks: *Technology, Data*, and *Business Intelligence Applications*. Our work is aligned into the data track, mainly the *Dimensional Modeling* step. Within this step, modelers identify dimensions, their granularity, attributes and numeric facts. Furthermore, our outcomes are useful in the phases of *Growth* and *Maintenance* since semantic data models are an ideal basis for analyzing new requirements on a business level (cf. figure 1).

The remainder of the paper is structured as follows: First of all, we will summarize related work. Second, we give a short introduction to the ADAPT notation and outline our repository solution as well as issues that occurred during its

implementation. In a third step we derive our structuring principles for ADAPT diagrams. Finally, we give an outlook to further research questions.



Figure 1. Data Warehouse Lifecycle diagram (Kimball et al., 2008)

#### **RELATED WORK**

This section outlines related work relevant for our paper: semantic multidimensional notations, repository implementations, meta-CASE toolsets, and some literature about measuring the quality of data models. Semantic notations for modeling multidimensional data can be clustered as follows: extensions to the Entity-Relationship model, extensions to the UML and ad hoc models (Rizzi, Abelló, Lechtenbörger and Trujillo, 2006). Extensions of the Entity-Relationship model are, for example, (Sapia, Blaschka, Höfling and Dinter, 1998) and (Malinowski and Zimámyi, 2006). Neither is able to model dynamic structures, e.g. calculation rules between measures. The work of (Luján-Mora, Trujillo and Song, 2006) serves as an example for UML-based notations by providing a profile for multidimensional modeling. The notation used in our paper, ADAPT (Bulos and Forsman, 2006), as well as the Dimensional Fact Model (Golfarelli, Maio and Rizzi, 1998) rank among the ad hoc approaches. Several methodologies have been presented to support the multidimensional design of data warehouses. Since there is a very good survey on these methodologies in (Romero and Abelló, 2009), we do not want to elaborate on these.

A repository is needed in order to store the metadata along the whole data warehousing process (Kimball et al., 2008). The Common Warehouse Metamodel (CWM) of the Object Management Group (OMG) is a much discussed standard for storing business intelligence metadata (OMG, 2003). Version two of this standard, the *Information Management Metamodel*, should have been published in September 2007 (OMG, 2005), but has not been published to date. For this reason, we are not concentrating on this standard for our repository. Nevertheless, there are two general standards for metadata repositories: Information Resource Dictionary System (ISO/IEC, 1990) and the Metadata Registry Standard (ISO/IEC, 2004).

A great deal of work has been done to create meta-CASE toolsets for developing modeling environments for domain specific visual languages. An outstanding work has been presented by (Zhu, Grundy, Hosking, Liu, Shuping and Mehra, 2007). They summarize plenty of related work and created a toolset which simplifies the development of modeling environments. All models are persistently saved in an XML format so that they can also be stored in a database. Other important works include, for example, MetaEdit+ (Steven, Kalle and Matti, 1996), Meta-MOOSE (Ferguson, Parrington, Dunne, Hardy, Archibald and Thompson, 2000), GMF (Eclipse, 2009), and DSL Tools (Cook, Jones, Kent and Wills, 2007). Those frameworks often require considerable effort to be understood and to be used effectively.

Concerning the quality of models in general, there has been presented a great deal of work. The Guidelines of Modeling (GoM) derive six principles to enhance the quality of information models: construction adequacy, language adequacy, economic efficiency, clarity, systematic design, and comparability (Schuette and Rotthowe, 1998). Considerable effort has been made in order to analyze the quality of E/R diagrams; one of the latest works is (Genero, Poels and Piattini, 2008). The authors assess indicators of how structural properties determine conceptual data model understandability. The third interesting work we would like to present here is (Wand and Weber, 2002). They constructed a framework for conceptual modeling consisting of a conceptual-modeling grammar, conceptual-modeling method, conceptual-modeling scripts, and a conceptual-modeling context. They introduced the term "script" for what is often called a "diagram" or "model". For assessing the quality of scripts, they use a semiotic approach presented by (Lindland, Sindre and Sølvberg, 1994). The latter work assesses the quality of information models by distinguishing the semiotic perspectives of syntax, semantics and

pragmatics. Their goals have been made realistic by introducing the notion of feasibility. The pragmatic quality of conceptual models, i.e. the audience interpretation, is especially interesting for us.

Our work is therefore aligned to (Wand and Weber, 2002) as well as (Lindland et al., 1994), since we would like to offer principles for enhancing conceptual-modeling methods especially for very large models (or scripts in the vocabulary of Wand and Weber). These principles should enhance and clarify the audience interpretation, especially the understanding of large models by separating these models into smaller chunks which can be understood more easily.

#### PREREQUISITES

#### Modeling Multidimensional Data with ADAPT

The ADAPT notation emerged in a practical consulting environment during the 1990s (Bulos, 1996). Due to its pragmatic roots, the notation is not entirely founded in a formal way. A proposal of a metamodel is provided by (Gluchowski, Kurze and Schieder, 2009). Further, the modeling language had largely been ignored in scientific publishing on conceptual modeling. The following paragraphs will outline the basic building blocks of this – from our point of view – sophisticated notation. Since ADAPT has already been established at our project partner and its staff is experienced in using ADAPT, we made use of this notation. To answer the open issue if ADAPT outperforms other notations, a survey of semantic multidimensional modeling notations as well as an evaluation and comparison of each notation against the Bunge-Wand-Weber model (Roseman and Green, 2002) should be conducted.

An introductory example, which will be used again in a later section of our paper, summarizes the application of the basic ADAPT elements (Gluchowski et al., 2009). In the simple business case, a company wants to analyze its product sales. Therefore, one cube "Sales" is modeled and connected with five dimensions ("Organizational Unit", "Customer", "Product", "Time", and "Measures"). The measures are encapsulated in the dimension "Measures" by modeling one member for each measure. The sample cube is depicted in figure 2 (left). We would like to outline only an exemplary product dimension. The dimension has a parallel hierarchy for aggregating products into their categories as well as their suppliers. The dimension itself has two attributes: a description in English and Spanish referring to all elements in the different hierarchy levels; the product level has three attributes which further describe each single product and are only valid for each single product – a product category, for example, will not have a weight. Figure 2 (right) shows the entire dimension. Due to space limitations we cannot go into further detail about ADAPT. The reader is referred to the relevant literature on this topic (Bulos and Forsman, 2006; Gluchowski et al., 2009).



Figure 2. Exemplary sales cube and product dimension

#### Project Scope of Industry Partner

This section outlines the requirements of our partner as well as problems that we were faced with while implementing the solution. Since metamodeling is a common approach in designing Domain Specific Languages (DSL), the requirements have been defined in a metamodel. The work of (Gluchowski et al., 2009) was the starting point of our work; but had to be enhanced by specific extensions.

There had to be a representation of multidimensional structures known as cubes, as well as flat reporting structures. The name "report" for flat structures had already been established in the environment of our partner. A report is the same as a cube, but it is represented in a flat way, e.g. as a spreadsheet, whereas cubes are displayed in Online Analytical Processing

(OLAP) applications. Another important fact was a clear distinction between a "regular" dimension element and measures as elements of a measure dimension. Therefore, we introduced a new modeling element which is to represent systems of performance indicators. This new element may contain measures and calculation rules between these measures. If modeled in a neatly arranged way, one can easily explore the hierarchical character of complex calculation structures. This distinction between different dimension types has been part of the initial ADAPT notation (Bulos, 1996).

For our repository solution, which is an integrated repository for all the requirements to the data warehouse solution, the requirements align with the aims of an integrated metadata repository stated by (Kimball et al., 2008). Our project partner wanted to undertake impact analyses on a semantic level, audit and (business) user documentation was given, and the metadata quality should be improved by executing queries on the repository to check the consistency of the models, for example. Therefore we needed an open repository which could be easily queried.

Additionally, we were not allowed to install a database server due to security reasons. There also should not be the necessity for any additional software licenses. The modelers of our partner enterprise were experienced users of Microsoft Visio and Access, so we decided to use these two products.

During the actual implementation we were faced with some problems of complexity. The entire metamodel consisted of only about ten classes and twenty associations, but due to the separation of cubes, reports, dimensions, and systems of performance indicators in different files, the implementation was difficult since a great deal of interdependencies had to be taken into account. To give an example, the partner required us to create facilities for connecting not only entire dimensions to a cube or report, but to also connect a subset of hierarchies to those dimensions. The actual problem was the insertion and deletion of elements, which resulted in several consistency checks and many lines of source code. We further formalized these issues and summarized them in five principles outlined in the next section.

#### PRINCIPLES FOR STRUCTURING SEMANTIC MULTIDIMENSIONAL DATA MODELS

(Bulos and Forsman, 2006) state that diagrams need to be reasonably sized, preferably on standard-sized sheets of paper. They recommend the division of an ADAPT diagram's layout into sections. Each section contains related objects. They recommend the distinction between sections for cubes and dimensions. One regular-sized page describes exactly one cube or exactly one dimension. As we have seen in the requirements definition above, there has to be a distinction between "regular" dimensions and systems of key performance indicators – the third section. The special requirements of our industry partner led to a fourth section, the "report" section. When developing a metamodel based solution, the sections must be clearly defined. This lead to our first two principles:

#### Principle One: Definition of a Notation Metamodel

In order to support a generic implementation, there has to be a clearly defined metamodel of the underlying notation. Without a metamodel, it is impossible to create a consistent solution. A particular implementation should be aligned to a metamodel framework, e.g. the Object Management Group's Meta Object Facility (OMG, 2006). In order to increase understandability of our principles we refrain from using such a complex framework and only outline the important elements. Further research has to give an alignment into the MOF (or another) architecture. A simplified metamodel is given in figure 3.



Figure 3. Metamodel of principle one

#### Principle Two: Definition of Model Sections

Each model section should contain all modeling elements that are related to each other. A modeling element may be included in more than one model section, which allows the interconnection of different sections. Confer, for example, the "Product" dimension. Its rectangular modeling element appears in the model section "cube" and "dimension" (cf. figure 2). The semantic is that the dimension within the cube section is further defined in its corresponding dimension section.

An appropriate tool allows the navigation through the different, but interconnected model sections (cf. principle four). Figure 4 depicts the metamodel.



Figure 4. Metamodel of principle two

#### Principle Three: Section Views

Because of the reuse of dimensions in different models sections, the number of implemented dimensions may be reduced. For example, a product dimensions has a parallel hierarchy. In the semantic definition of reports, we might have three options: only one of both hierarchies is used, or both hierarchies are used. We then have three views on our product dimension (the last one with both hierarchies is the same as the dimension itself and exists per default). An abstraction of this example leads to principle three.

Each model section might have views on this section (cf. the given examples in the next part of this paper), whereas each view consists of at least one model element. There is at least one section view that is the same as the given section itself (i.e. the whole product dimension in the example above). The metamodel is provided in figure 5.



Figure 5. Metamodel of principle three

An additional constraint defines that model elements as well as associations used in the section view must be allowed in the related model section. The following OCL constraint expresses an invariant for all instances of the class SectionView (Warmer and Kleppe, 2003). It defines that all elements and associations used within the current section view must be contained in the elements or associations collection of the corresponding model section.

```
context SectionView
inv definedModelElements: section.elements->includesAll(elements)
inv definedModelAssociations: section.associations->includesAll(associations)
```

#### Principle Four: Navigation Support

This principle is necessary for the connection of different model sections and model views. Its realization results in tools that allow efficient navigation between the different sections and views of diagrams, i.e. the users do not have to search for files or database records; the tool just opens the section of interest. It therefore differs from principle five, since principle four explicitly says that there has to be an easy way for navigating through the whole model collection. However, it opens up the possibility to conduct (manual) lineage and impact analyses on the semantic data models (Kimball et al., 2008). They foster the understanding of large models by facilitating their inspection chunk for chunk.

#### Principle Five: Repository-based Implementation

As we have seen in our project, it is necessary to base the implementation on a repository. Otherwise, it would not be possible to handle hundreds of files containing data models. The benefits of using standards-based repository systems

facilitate the implementation of principles one to four. They can be summarized as follows (ISO/IEC, 1990). From a database management system's perspective there are the following aspects: enforcement of constraints, access control, audit trail, limits and defaults, database integrity, query and reporting facilities, and remote data access.

From the point of view of information resource management, the following repository facilities assist the development, introduction, and use of information processing systems. They go beyond general-purpose database management systems (ISO/IEC, 1990): naming (unique identification of objects), status of dictionary content, information system lifecycle management, version control, partitioning, sub setting, copy creation, and impact analysis.

#### **EVALUATION OF PRINCIPLES**

This section outlines the application of the five principles within our project. We will show an extract of the metamodel (we are not allowed to publish the entire metamodel), the defined model sections, the implementation of section views as well as the repository based implementation.

In order to implement principle one, we created a metamodel of the ADAPT notation based on (Gluchowski et al., 2009). The classes shown without attributes are instances of the metaclass "ModelElement", the classes ending with "Ass" are instances of "ModelAssociation". Their member "type" identifies the used arrow symbol. The roles "in" and "out" identify the arrow head's direction. Figure 6 depicts the metamodel.



Figure 6. Adjusted ADAPT metamodel

Five invariants in OCL enrich the metamodel. A dimension may have either hierarchies or members. Furthermore, the top level of a dimension is either connected via the symbol "Connector" (and shows that there is no aggregating level "All Elements") or via the symbol "Strict Precedence" (and thus implying an aggregating level "All Elements"). The relationships between levels also have to be specified in more detail. If "Strict Precedence" or "Loose Precedence" are used, "in" and "out" have to be different instances of "Level". Modeling a recursive relationship via the "Self Precedence" arrow requires "in" and "out" to be the same instance of "Level". The following OCL statements are necessary:

```
context Dimension
inv hierarchiesXORMembers: hierarchies->size() > 0 xor members->size() > 0
context Hierarchy
inv allXORNotAll: toplevel->size() > 0 xor toplevelStrict->size() > 0
context LevelLevelStrictAss
inv inOut: in <> out
context LevelLevelLooseAss
inv inOut: in <> out
context LevelLevelSelfAss
inv inOut: in = out
```

Applying principle two in our project led to the identification of four model sections. For space reasons and the limited metamodel presented above, we only outline two of these section: "Cube" and "Dimension". Table 1 summarizes the model elements and associations in each section.

Model Section	Model Elements	Model Associations
Cube	Cube Dimension	CubeDimensionAss
Dimension	Dimension Member Hierarchy Level Attribute	DimensionMemberAss ModelMemberInputAss ModelMemberOutputAss DimensionHierarchyAss HierarchyLevelToplevelAss HierarchyLevelToplevelStrictAss LevelLevelStrictAss LevelLevelSelfAss LevelLevelLooseAss DimensionAttributeAss LevelAttributeAss

 Table 1. Model sections and their elements as well as associations

The following example illustrates the idea of views, i.e. principle three. Therefore, we refer to the example of the introduction to ADAPT. Assuming that we would like to analyze our sales figures in two ways – by aggregating products according to their categories on the one hand, and according to their suppliers on the other hand – a parallel hierarchy is modeled in the product dimension. In order to keep the reports as small and clear as possible, there are two different views on the cube which result from different views on the product dimension. The whole cube and the whole product dimension are shown in figure 2. The views on the cube are presented in figure 7. They only differ in the usage of a different view on the dimension "Product". The example is very simple, in our project the reporting structures differed in several dimensions. Sometimes one dimension had more than five different views.



Figure 7. "Category" and "Supplier" view on the "Sales" cube

Figure 8 clarifies the usage of different elements in different views. We propose to always depict the whole dimension, but to "disable" all elements and associations that are not used. All elements not valid for a particular view are displayed in gray. A suitable tool will help creating new views by showing the whole dimension and offering facilities to "disable" unused elements and associations. Another benefit of displaying the views in this way is the geometry information of each element – only one position has to be saved persistently within the repository. The containment of model elements and associations within a view is a binary relationship between view and each single model element or association.

The repository based implementation assembles principles four and five. We have chosen to implement a very simple architecture due to a short timeframe for the project. According to the requirements, we only made use of standard office tools. The main folder within the file system contains the Access database "repository.mdb", the (adjusted) Visio stencil "adapt.vss" and folders for the ADAPT diagrams according to the defined modeling sections: "cube", "dimension", "measure", and "report". These folders contain the modeled ADAPT diagrams. The whole VBA code is stored in the stencil file. This allows an easier maintenance since not every single document will have to be changed if the code changes. Figure 9 summarizes the basic ideas of the underlying architecture.



Figure 8. "Category" and "Supplier" view on the "Product" dimension

The interconnection between different model sections is done via an extension of the context menu. In order to navigate from a cube definition to the dimension definitions, the user has to click the right mouse button on the dimension object and choose "Show definition..." from the context menu. This solution has proved to be easy to use and fast. As figure 9 also shows, there can be several reporting structures defined on top of the repository. Our industry partner developed several forms in Microsoft Access for analyzing the semantic data models.



Figure 9. Repository implementation architecture

#### **FUTURE WORK**

The present paper represents research in progress. Therefore, an empirical validation of the principles has to be undertaken in the future. We are currently planning a class experiment. The research framework for this study is given in (Maes and Poels, 2007), (Genero et al., 2008) as well as (Jones and Song, 2008). The first paper introduces and validates a model for measuring the semantic and pragmatic quality of semantic models based on the model of information systems success by (Seddon, 1997) and an extension of (Lindland et al., 1994) by (Krogstie, Lindland and Sindre, 1995). The authors propose questions that measure quality aspects of interest. They have to be answered on a 7-point Likert scale with response options ranging from "strongly disagree" to "strongly agree".

The students will take part in the experiment after two courses concerning data warehousing and business intelligence. After that, they should be able to understand multidimensional data models and their level of knowledge should be almost same. In order to answer the questionnaire, the students have to answer some questions addressing the contents of the model and they have to conduct minor changes to the model. Suggestions on which tasks to perform can be taken from (Genero et al., 2008). By randomly selecting students a control group and an experiment group will be created (Jones and Song, 2008). The

experiment group will get further education on ADAPT which introduces our five principles, i.e. they will be provided the metamodel as well as an application assembling principles two to five.

Our hypothesis is that the answers from the experiment group outperform those of the control group, i.e. the tool support raises semantic and pragmatic quality as well as usefulness and user satisfaction of the model. Therefore, we will perform a *t* test (paired variables test) on the answers to determine if the use of our principles does indeed have an impact on the perceived quality of the semantic model as done in a similar study by (Jones and Song, 2008).

More research has to be done in order to embed our work into the context as well as the method for conceptual modeling (Wand and Weber, 2002). This will include an analysis of development methodologies for data warehousing (Romero and Abelló, 2009) and the deduction of a project management guideline for the development of multidimensional data models. Concerning the actual implementation, we should further align our principles to a metadata framework like MOF (OMG, 2006).

#### REFERENCES

- 1. Bulos, D. (1996) OLAP Database Design, A new dimension, *Database Programming & Design*, 9, 6, 33-37.
- 2. Bulos, D., Forsman, S. (2006) Getting Started with ADAPT, *Whitepaper*, http://symcorp.com/downloads/ ADAPT\_white\_paper.pdf, [01-22-2009].
- 3. Cook, S., Jones, G., Kent, S., Wills, A.C. (2007) Domain-Specific Development with Visual Studio DSL Tools, Addison-Wesley, Amsterdam.
- 4. Eclipse (2009) Eclipse Graphical Modeling Framework (GMF), http://www.eclipse.org/modeling/gmf/, [01-23-2009].
- 5. Ferguson, R.I., Parrington, N.F., Dunne, P., Hardy, C., Archibald, J.M., Thompson, J.B. (2000) MetaMOOSE an object-oriented framework for the construction of CASE tools, *Information and Software Technology*, 42, 2, 115-128.
- 6. Genero, M., Poels, G., Piattini, M. (2008) Defining and validating metrics for assessing the understandability of entityrelationship diagrams, *Data & Knowledge Engineering*, 64, 3, 534-557.
- Gluchowski, P., Kurze, C., Schieder, C. (2009) A Modeling Tool for Multidimensional Data using the ADAPT Notation, in: Ralph H. Sprague Jr. (Ed.) *Proceedings of the 42nd Hawaii International Conference on System Sciences*, January 5-8, Waikoloa, Hawaii, USA, CD-ROM, IEEE Computer Society.
- 8. Golfarelli, M., Maio, D., Rizzi, S. (1998) The Dimensional Fact Model: A Conceptual Model for Data Warehouses, *International Journal of Cooperative Information Systems*, 7, 2-3, 215-247.
- 9. ISO/IEC 10027:1990 (1990) Information technology Information Resource Dictionary System (IRDS) framework, *International Standard*.
- 10. ISO/IEC 11179-1:2004 (2004) Information technology Metadata registries (MDR) Part 1: Framework, 2nd edn., *International Standard*.
- 11. Jones, M.E., Song, I.-Y. (2008) Dimensional Modeling: Identification, classification, and evaluation of patterns, *Decision Support Systems*, 45, 1, 59-76.
- 12. Kimball, R., Ross, M., Thorntwaite, W., Mundy, J., Recker, B. (2008) The Data Warehouse Lifecycle Toolkit, 2nd edn., Wiley, Indianapolis.
- 13. Krogstie, J., Lindland, O.I., Sindre, G. (1995) Defining quality aspects for conceptual models, in: Eckhard D. Falkenberg, Wolfgang Hesse (Eds.) *Proceedings of the IFIP international working conference on Information system concepts: Towards a consolidation of views*, March 28-30, Marburg, Germany, Chapman & Hall, 216-231.
- Lindland, O.I., Sindre, G., Sølvberg, A. (1994) Understanding quality in conceptual modeling, *IEEE Software*, 11, 2, 42-49.
- 15. Luján-Mora, S., Trujillo, J., Song, I. (2006) A UML profile for multidimensional data modeling in data warehouses, *Data & Knowledge Engineering*, 59, 3, 725-769.
- 16. Maes, A., Poels, G. (2007) Evaluating quality of conceptual modeling scripts based on user perceptions, *Data & Knowledge Engineering*, 63, 3, 701-724.
- 17. Malinowski, E., Zimámyi, E. (2006) Hierarchies in a Multidimensional Model: From Conceptual Modeling to Logical Representation, *Data & Knowledge Engineering*, 59, 2, 348-377.

- OMG (2003) Common Warehouse Metamodel (CWM) Specification Version 1.1, http://www.omg.org/ docs/formal/03-03-02.pdf, [01-23-2009].
- 19. OMG (2005) Request For Proposal: Information Management Metamodel (IMM), http://www.omgwiki.org/imm/lib/exe/fetch.php?id=welcome\_to\_imm&cache=cache&media=05-12-02.pdf, [01-23-2009].
- 20. OMG (2006) Meta Object Facility (MOF) Core Specification Version 2.0, http://www.omg.org/docs/formal/06-01-01.pdf, [01-23-2009].
- 21. Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J. (2006) Research in Data Warehouse Modeling and Design: Dead or Alive?, in: Panos Vassiliadis, Il-Yeol Song (Eds.) *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, November 10, Arlington, Virginia, USA, ACM Press, 3-10.
- 22. Romero, O., Abelló, A. (2009) A Survey of Multidimensional Modeling Methodologies, *International Journal of Data Warehousing & Mining*, 5, 2, 1-23.
- 23. Roseman, M., Green, P. (2002) Developing a meta model for the Bunge-Wand-Weber ontological constructs, *Information Systems*, 27, 2, 75-91.
- 24. Sapia, C., Blaschka, M., Höfling, G., Dinter, B. (1998) Extending the E/R Model for the Multidimensional Paradigm, in: Yahiko Kambayashi, Dik Lun Lee, Ee-Peng Lim, Mukesh Kumar Mhania, Yoshifumi Masunga (Eds.) Advances in Database Technologies, Proceedings of the ER '98 Workshops on Data Warehousing and Data Mining, Mobile Data Access, and Collaborative Work Support and Spatio-Temporal Data Management (Lecture Notes in Computer Science, Vol. 1552), November 19-20, Singapore, Springer, Berlin, Heidelberg et al., 105-116.
- 25. Schuette, R., Rotthowe, T. (1998) The Guidelines of Modeling An Approach to Enhance the Quality in Information Models, in: Tok Wang Ling, Sudha Ram, Mong Li Lee (Eds.) *Proceedings of the 17th International Conference on Conceptual Modeling* (Lecture Notes in Computer Science, Vol. 1507), November 16-19, Singapore, Springer, Berlin, Heidelberg et al., 240-254.
- 26. Seddon, P. (1997) A respecification and extension of the DeLone and McLean model of IS success, *Information Systems Research*, 8, 3, 240-253.
- 27. Steven, K., Kalle, L., Matti, R. (1996) MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment, in: Panos Constantopoulos, John Mylopoulos, Yannis Vassiliou (Eds.) Advanced information systems engineering, Proceedings of the 8th International Conference on Advances Information System Engineering (Lecture Notes In Computer Science, Vol. 1080), May 20-24, Heraklion, Crete, Greece, Springer, Berlin, Heidelberg et al., 1-21.
- 28. Wand, Y., Weber, R. (2002) Research Commentary: Information Systems and Conceptual Modeling A Research Agenda, *Information Systems Research*, 13, 4, 363-376.
- 29. Warmer, J., Kleppe, A. (2003) The object constraint language: getting your models ready for MDA, Addison-Wesley, Boston, Munich et al.
- Zhu, N., Grundy, J., Hosking, J., Liu, N., Shuping, C., Mehra, A. (2007) Pounamu: A meta-tool for exploratory domainspecific visual language tool development, *The Journal of Systems and Software*, 80, 3, 1390-1407.