MULTI-DIMENSIONAL MODELING
Formal Specification and Verification of the Hierarchy Concept

Ali Salem
Faculty of Sciences of Sfax,University of Sfax, Sfax, Tunisia

Faiza Ghozzi
Institute Superior of Informatics and Multimedia of Gabes,University of Gabes, Gabes, Tunisia

Hanene Ben-Abdallah
Faculty of Economics and Management of Sfax, University of Sfax, Sfax, Tunisia

Keywords: Multi-dimensional model, OLAP, Constraints, Meta-Model, formal Specification, Hierarchy.

Abstract: The quality of a data mart (DM) tightly depends on the quality of its multidimensional model. This quality dependence motivated several research efforts to define a set of constraints on the DM model/schema. Currently proposed constraints are either incomplete, or informally presented, which may lead to ambiguous interpretations. The work presented in this paper is a first step towards the definition of a formal framework for the specification and the verification of the quality of DM schemas. In this framework, the quality is expressed in terms of both the syntactic well-formedness of the DM schema as well as its semantic soundness with respect to the DM instances. More precisely, this paper first formalizes in Z the constraints pertinent to the hierarchy concept; the formalization is treated at the meta-model level. Secondly, the paper illustrates how the formalization can be instantiated and the constraints are verified for a particular sample model through the theorem prover Z\eves.

1 INTRODUCTION

Face to the international, unrestrained economic competition, an increasing interest in decision support systems (DSS) has emerged over the last decade. These latter assist a decision maker in extracting data pertinent to their analysis interests from their transactional systems (called On-Line Transactional Processing-OLTP systems). Unlike transactional systems, the majority of DSS relies on OLAP (On-Line Analytical Processing) systems where data are often stored in multidimensional databases. In such databases, data is organized by center of interests (Facts) and examined according to various axes of analysis (Dimensions) represented through analysis prospects (Hierarchies) (Kimball, 2002). These multidimensional databases are often organized in terms of two type's storage areas: a data warehouse (DW) that regroups all data required for any potential analysis requirements, and/or a set of data marts (DM) each of which regroups data extracts required to one particular analysis requirement.

In order to assist in the development of DM/DW, several multidimensional models have been proposed to specify DM/DW schemas, e.g., the basic star model, its generalization the constellation model, the snow flake model, etc. (Hurtado, 2002] (Lechtenbörger, 2003). On the other hand, evidently, the quality of a DM/DW depends tightly on the quality of its schema. In this context, we consider that the quality of a schema those properties that can be expressed in terms of the schema's syntactic (or structural) specification as well as its semantic correctness (or soundness). The syntactic quality of a schema ensures that elements of the multidimensional model are correctly used together; for example, the acyclicity constraint disallows the existence of a level cycle in the hierarchy. The soundness constraint defines a hierarchical link of dependence between the dimension instances...

Salem A., Ghozzi F. and Ben-Abdallah H. (2008).
MULTI-DIMENSIONAL MODELING - Formal Specification and Verification of the Hierarchy Concept.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - DISI*, pages 317-322

The quality of a schema has been addressed by the definition of a set of constraints at the model level. That is, several researchers (cf., (Hurtado, 2002),(Lechtenbörger, 2003), (Ghozzi, 2003) have defined a set of rules that a schema must respect in order to produce either a syntactically correct schema, or a sound schema with respect to data instances. The rules are defined on the structures and structural elements of a schema. Nevertheless, these works have not proposed a mechanism to validate and verify these rules.

In this paper, we present the first steps towards the development of a formal framework for DM/DW modeling and verification. On one hand, this framework relies on the precise definition of the constraints ensuring the syntactic and semantic correctness of a DM/DW schema. On the other hand, its exploits the formal definition in order to provide for a means to verify both types of correctness. More specifically, in this paper, we first present the formal definition of the Hierarchy concept at the meta-model level in the Z language (Spivey, 1992); secondly, we illustrate how the constraints can be instantiated for a particular model and verified using the Z/eves theorem prover (Saaltink, 1999).

The remainder of this paper is organized as follows. In Section 2, we first overview current proposals of constraints for DM/DW schemas; secondly, we present our approach of constraint definition and verification. In Section 3, we present the set of constraints pertinent to the hierarchy concept and their formalization in Z. In Section 4, we show how to instantiate the constraint for a particular model and how to verify the correctness of the constrained model through Z/eves. Finally, Section 5 summarizes our contributions and outlines ongoing work.

2 RELATED WORKS

During our survey of the previous works in this domain, we field studied the hierarchy concept and the constraints related to this concept. Defining the hierarchies classification of certain dimension attributes is crucial because these classification hierarchies provide the basis for the subsequent data analysis. Since a dimension attribute can also be rolling up to more than one other attribute, multiple classification hierarchies and alternative path hierarchies are also relevant (Trujillo, 2001). According to (Lehner, 1998), in the context of statistical databases and on-line analytical processing as well, classification hierarchies provide a basis for defining aggregate data. (Part, 2006) confirms that hierarchies are crucial to multidimensional modeling since they are used in conjunction with aggregation functions to aggregate (“rollup”) or detail (“drill-down”) measures. These quotations prove the importance related to the hierarchy concept. In (Malinowski, 2004), the authors present a conceptual classification of hierarchies and propose graphical notations for them based on the ER model. With respect to dimensions, every hierarchy classification level is specified by a class. An association of classes specifies the relationships between two levels of a hierarchy classification. The only prerequisite is that these classes must define a Directed Acyclic Graph (DAG) rooted in the dimension class (constraint {dag} placed next to every dimension class). The DAG structure can represent both alternative path and multiple hierarchies classification (Lujàn, 2002).

In the GMD model (Franconi, 2004), the authors describe the hierarchy by an order function between the different dimension attributes. (Abello, 2006) presents a multi-dimensional model object oriented, and defines a hierarchy as aggregation relation between the different dimension attributes.

Otherwise, few works formally define hierarchies but they mainly discuss the summarizability conditions and offer some solutions to correct measure aggregations in presence of the so-called heterogeneous hierarchies (Hurtado, 2001).

In (Hurtado, 2002), the authors propose a set of constraints to solve the aggregation problem. These constraints are related to the hierarchical structuring of the dimension attributes and the dimension instances. We note an explicit and complete definition of the hierarchy concept in the works of (Ghozzi, 2003). These works will make the basis of our formal specification.

The constraints expressed in these works differ from a model to another. This difference resides, on the one hand, in the level of expression of the constraint (Meta-model, model) and on the other hand, in the level of checking or safeguarding of the constraint. Moreover, there is no consensus on the whole constraints to take into account. This dissension on the level of the constraints expression in these various works poses a true problem touching with the coherence of the data to incorporate. In other words, it can lead to incoherent results of analyses.

The goal of our work is to lead to a consistent formal specification of a multidimensional Meta-model in constellation. Thus, we offer the designers a means to check their models.

In this paper, we present a formal specification of the hierarchy concept and a domain check validation of this specification with Z/Eves proover. We prove, as well, the consistence of this specification with an initial state theorem (Spivey, 1992).

3 CONSTRAINTS RELATED TO THE HIERARCHY CONCEPT

The constraints expressed at the Meta-model level include the essential constraints to maintain the coherence model. They are related to the basic concepts and independent of any application. This type of constraints can be classified according to basic concepts' of the multidimensional model; fact, dimension, Hierarchy and constellation.

Among these constraints we differentiate between the constraints related to the hierarchy concept. These constraints can be classified in two categories according to their checking level: Structure constraints and instances constraints.

3.1 Structure Constraints

These constraints describe the rules to scheduling hierarchy attributes:

- Unicity of identifier and "All" attributes (Ghozzi, 2003). For example, in supplier dimension (Fig1) we can find several Supplier whish have the same identifier. This can generate ambiguities when we query multidimensional data because facts will be related to more than one Supplier. In addition, All attribute is defined to enclose a hierarchy (Fig 1)
- The identifier is the attribute of the finest granularity and "All" is the attribute of highest granularity (Fig 1) (Ghozzi, 2003). In a hierarchy, attributes are classified according to a partial order (roll up). For example, in supplier hierarchy (Fig 1), ID determines City determines Country etc. Only ID can determine all the information related to a Supplier. The attributes are classified from the finest granularity to the highest granularity. "All" attribute is used to enclose a hierarchy.

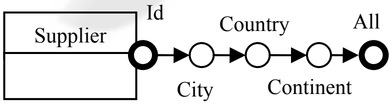


Figure 1: Attributes ID and ALL.

- Nonempty hierarchy (Ghozzi, 2003): Each hierarchy has at least two levels of parameters; ID and the "All" attribute.
- Acyclicity (Franconi, 2004) (Abello, 2006) (Hurtado, 2002) (Carpani, 2001) (Ghozzi, 2003): this constraint disallows the existence of a level cycle in the hierarchy. For example, for the supplier hierarchy (Fig 2), we notice the existence of the cycle (continent, city). This cycle generates a problem of redundancy during the data analysis.

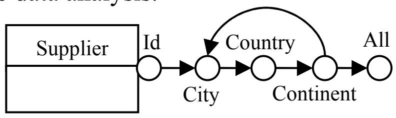


Figure 2: Acyclicity.

- Connection to the top (Hurtado, 2002) (Ghozzi, 2003) this constraint expresses that all parameters, except All, have at least a father (a parameter of less fine granularity) (Fig 3). This constraint guarantees the order structure of parameters in dimension.

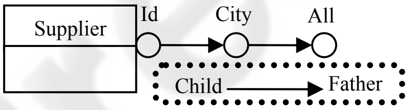


Figure 3: Connection to the top.

3.2 Instances Constraints

The hierarchical structure between the parameters is also applied on their members. Thus, several works speak about the dimension diagram of instances. The constraints on this level describe the relations between the various values of the attributes.

- Partition (Hurtado, 2002) (Ghozzi, 2003). To each parameter member corresponds one and only one member among those of each parameter successor in the hierarchy. In a hierarchy, each member must determine the successor member. If a parameter member corresponds to more than one successor member we can not determine this successor member. For example (fig 4), the city of Paris (member of city parameter) should not belong at the same time to France and Italy (member of country parameter).

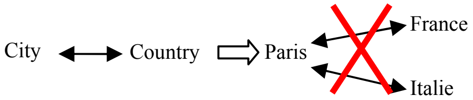


Figure 4: Partition.

- Soundness (Ghozzi, 2003): for each parameters couple in the Hierarchy, there are at least two members belonging to this couple in such a way that these two members are dependent (fig 5). In our schema, this dependence is materialized by the existence of a relation between these members.

City
Country
Paris
France

Figure 5: Soundness.

4 FORMAL SPECIFICATION IN LANGUAGE Z

The formal specification of our model allows the expression of the constraints in an exact and specific way, thus offering the means of validating and checking them.

The selected language of specification is Z language. It is based on the set theory and mathematical logic (Spivey, 1992). The set theory used includes the standard operators of the sets, the Cartesian products and the sets of power. Mathematical logic is a first order predicate calculus. A schema Z is composed of two parts: a part for the declaration and a part for the predicates representing the constraints on the declared variables.

4.1 Specification

We start by defining the two types *NomH* and *Dom*: The first includes Hierarchy names and the second includes the various attribute values in dimensions:

[*NomH*, *Dom*]

Then, the Standard free type, which is used for classification of dimension attributes as weak attributes, parameters, identifier or “All”:

Type ::≪ *weak* ⎵ *Parameter* ⎵ *ID* ⎵ *All*

The *Relval* relation things to see the various relations between the attributes values:

‖*Relval: Dom* ⭘ *Dom*

The *Weight* function assigns to each attribute a weight:

‖ *Weights: AttDim* ≻ *Type*

Each dimension attribute includes a finished set of values (*Dom*). We define it as a compound type.

```
 ⤡AttDim⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡
‖val:⎵Dom
 ⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡
```

The formal definition of a hierarchy in language Z results in the schema named *Hierarchy* where: *N* is the name of the hierarchy; *Att* is a finished whole of dimension attributes *AttDim*. *ParamH* is a sequence describing the attributes hierarchy. A sequence, in language Z, can be considered as a function whose field is an adjoining subset of the natural numbers.

The predicate part of the *Hierarchy* schema gathers constraints connected to:

- The unicity of : identifier [1],
- The unicity of the parameter All [2],
- Nonempty hierarchy [3]: Each hierarchy has at least two levels of parameters in the definition of the hierarchy.
- The identifier is the attribute of the finest granularity [4],
- The All attribute is of the largest granularity [5],
- Acyclicity [6]: The existence of a cycle in the hierarchies is forbidden.
- Soundness [7]: *ParamH* defines a hierarchical link of dependence between the parameter members of a dimension.
- Partition constraint [8] [9]: to each parameter member corresponds one and only one member among those of each successor parameter in the hierarchy.
- Connection to the top: All the parameters, except “All”, have at least a father (a parameter of less fine granularity). This constraint makes it possible to ensure the passage from a level to another. It is checked by definition of *ParamH* as a sequence.

```
 ⤡Hierarchy⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡
‖N: NomH
‖Att: ⎵ AttDim
‖ParamH: seq AttDim
 ⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡
‖[1] ⎵₁x: Att ℕ  Weights x = ID
‖[2] ⎵₁x: Att ℕ  Weights x = All
‖[3] # ParamH ⁄ 2
‖[4] ≙x: Att ℕ x ⊙ ran ParamH     ParamH 1 = x
‖      Weights x = ID
‖[5] ≙x: Att ℕ x ⊙ ran ParamH     ParamH ‿# ParamH⁀ = x
‖       Weights x = All
‖[6] ≙i, j: 1 .. # ParamH ⎵ i Đ j ℕ ParamH i Đ ParamH j
‖[7] ≙i, j: 1 .. # ParamH ⎵ j = i + 1
‖  ℕ ⎵v, w: Dom ⎵ v ⊙ ‿ParamH i⁀ . val     w ⊙ ‿ParamH j⁀ . val
‖      ℕ ‿v    w⁀ ⊙ Relval
‖[8] ≙i, j: 1 .. # ParamH; v: Dom ⎵ j = i + 1      v ⊙ ‿ParamH i⁀ . val
‖  ℕ ⎵w1, w2: Dom
‖      ℕ w1 ⊙ ‿ParamH j⁀ . val
‖           w2 ⊙ ‿ParamH j⁀ . val
‖           ‿v    w1⁀ ⊙ Relval
‖           ‿v    w2⁀ ⊙ Relval
‖           w1 = w2
‖[9] ≙v: Dom; i: 1 .. # ParamH - 1 ⎵ v ⊙ ‿ParamH i⁀ . val
```

‖ ℕ ⎵*w: Dom* ⎵ *w* ⊙ ‿*ParamH* ‿*i* + 1⁀⁀ . *val* ℕ ‿*v* *w*⁀ ⊙ *Relval*

⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡

4.2 Proof Reading

The proof involves a demonstration that the various requirements upon the data type are consistent and not contradictory. To show that the requirements are consistent, we have only to show that the constraint part of the state schema is satisfiable. This is usually achieved by proving an initialisation theorem: we show that an initial state, at least, exists.

In this phase, we will test the consistency of our specification. The step to be followed in this phase is to define a correct hierarchy and to prove that the latter checks the constraints expressed on the level of the Hierarchy schema.

For that, we start with the instantiation of the Clas_Cons Hierarchy of Car dimension (Fig 6):

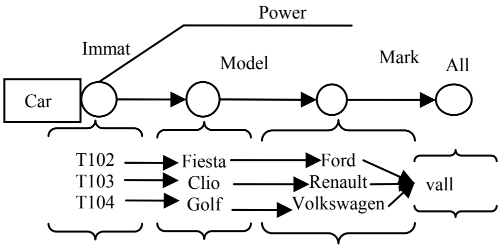


Figure 6: The Clas_Cons Hierarchy of dimension Car.

Among the attributes of Car dimension, we have Immat, Model, Mark, Power and All. The Immat attribute is the identifier. The attributes Model and Mark are classified as parameters. Power is a weak attribute. These attributes are classified according to the hierarchy Clas_Cons (Fig 6).Each attribute contains a set of values related to each other according to the relation Relval (Fig 6).

The description of this example in Z is written to use the Axiom Box Schema. The form of definition includes a constraint upon the object being introduced. Such definitions are said to be axiomatic, as the constraint is assumed to hold whenever the symbol is used: it is an axiom for the object. In the Z notation, our example description is:

‖*Immat, Model, Mark, all, Power: AttDim*

‖*clas_Cons: NomH*

‖*T102, T103, T104, Fiesta, Clio, Golf, Ford,*

‖ *Renault, Volkswagen, vall: Dom*

⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡

‖*Weights Immat = ID*

‖ *Weights Mark = Parametre*

‖ *Weights Model = Parametre*

‖ *Weights all = All*

‖ *Weights Power = Faible*

‖*Immat . val =* ‿‿*T102 T103 T104*⁀⁀

‖*Model . val =* ‿‿*Fiesta Clio Golf*⁀⁀

‖*all . val =* ‿‿*vall*⁀⁀

‖*Mark . val =* ‿‿*Ford Renault Volswagen*⁀⁀

‖*Relval*

‖ = ‿‿‿*T102 Fiesta*⁀ ‿*Fiesta Ford*⁀ ‿*Ford vall*⁀ ‿*T103 Clio*⁀

‖ ‿*Clio Renault*⁀ ‿*Renault vall*⁀ ‿*T104 Golf*⁀ ‿*Golf Volkswagen*⁀

‖ ‿*Volkswagen vall*⁀⁀⁀

Then, we define the *HierarchyInstance* schema which will play the role of a hierarchy instances. In addition, we must assign the various values necessary to the various sets already declared to the level of the *Hierarchy* schema.

⤡*HierarchyInstance*⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡

‖*Hierarchy*

⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡

‖*N = clas_Cons*

‖*Att =* ‿‿*Immat Model Mark all Power*⁀⁀

‖*ParamH = Immat Model Mark all*ﬂ

⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡⤡

At last, we proved the theorem of following initialization state:

theorem *ConsistanceHierarchie*
⎵*Hierarchy* ℕ *HierarchyInstance*

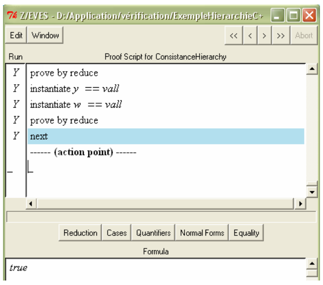
To prove this theorem we use the Z/Eves Prover.

This prover is semi-automatic. His is bases on set theory and first order logic.

In the environment of demonstration, one can use several commands to indicate to Z-Eves tracks of possible demonstrations. For some, it is a question of clicking, for others; we have to introduce other sub-theorems and axioms to assert the prover.

In our case to prove the initial state theorem we proceeded as follows (fig 8):

- *invoke*: this command replaces all the schema used in predicate by their definition.
- *prove*: this command allows various options of demonstration and simplification.
- *use*: is used to explicitly call another theorem or an axiom.
- *prove by reduce*: it has the same role as the prove command, but it simplifies even better.
- *case* and *next*: in our proof, there are many predicates to be shown. The best solution is to use these two commands which make it possible to show them separately.
- *instantiate*: this command is useful to instantiate one or more quantified variables.

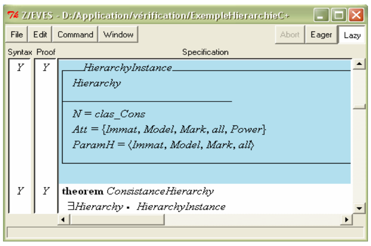

Z/EVES - D:/Application/vérification/ExempleHierarchieC+
Edit
Window
<<
<
>
>>
Abort
Run
Proof Script for ConsistanceHierarchy
Reduction
Cases
Quantifiers
Normal Forms
Equality
Formula

```
Y  prove by reduce
Y  instantiate y  == vall
Y  instantiate w  == vall
Y  prove by reduce
Y  next
   ------ (action point) ------
```

```
true
```

Figure 7: Proof script for theorem *ConsistanceHierarchy*.

Finally, we succeeded in proving the theorem of the initialization state. We have, so, to prove that there is not a contradiction to the constraints specified in the predicate part of the Hierarchy schema. Consequently, we proved the consistence of our formal specification of the hierarchy concept.



```
Y  Y  HierarchyInstance
        Hierarchy
        ----------------------------
        N = clas_Cons
        Att = {Immat, Model, Mark, all, Power}
        ParamH = ⟨Immat, Model, Mark, all⟩

Y  Y  theorem ConsistanceHierarchy
        ∃Hierarchy • HierarchyInstance
```

Figure 8: Proved theorem *ConsistanceHierarchy*.

5 CONCLUSIONS

In this paper, we gathered all the constraints related to the hierarchy concept judged to be essential to maintain the coherence of the data to incorporate and ensure the integrity of the structures and the multidimensional data compared to these constraints. We defined these constraints at the meta-model level. In addition, we proposed to classify these constraints according to two categories: the constraints related to the structure and the constraints related to the instances. Further more, we precisely formalized in Z the Hierarchy concept. This formalization provides the designers with the means of validating the hierarchical structuring of the dimension attributes of their models.

REFERENCES

Abelló A., Samos J., Saltor F. “YAM2: a multidimensional conceptual model extending UML”. Information Systems. Vol 31, p 541-567, 2006

Carpani F., Ruggia R., “An Integrity Constraints Language for a Conceptual Multidimensional Data Model”. In 13th International Conference on Software Engineering & Knowledge Engineering (SEKE’01), Argentina, 2001.

Codd E. F., Codd S.B., Salley C.T., "Providing OLAP (On Line Analytical Processing) to Users-Analysts: An IT Mondate", Rapport technique, E.F. Codd and Associates, 1993.

Franconi E. and Kamble A., “The GMD Data Model and Algebra for Multidimensional Information” Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Latvia, June 7-11, 2004, Proceedings.

Ghozzi F., Ravat F., Teste O., Zurfluh G., “Modèle Dimensionnel à Contraintes”. In Revue des Sciences et Technologies de l'Information, Série RIA- ECA, Hermes –Lavoisier, Vol. 17, N. 1-2-3, p.43-56, 2003.

Hurtado C., “Mendelzon A., Reasoning about summarizability in heterogeneous multidimensional schemas”, in: Proc. of the 21st ACM Int. Conf. on Management of Data and Symposium on Principle of Databases Systems, p. 169–179, 2001.

Hurtado C., Mendelzon A., "OLAP Dimension Constraints". In 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS’02), Madison, USA, p. 169-179, 2002.

Kimball R., Ross M., "The Data Warehouse Toolkit", Wiley, New York, 2th edition, 2002.

Lechtenbörger J., Vossen G., "Multidimensional normal forms for data warehouse design". In Revue Information Systems, Vol. 28, N. 5, p. 415-434, 2003.

Lujàn S, Trujillo J., Song, "Extending the UML for Multidimensional Modeling" The Unified Modeling Language: 5th International Conference, Dresden, Germany, September 30 - October 4, 2002, Proceedings.

Lehner W., Albrecht J., Wedekind H., “Normal forms for multidimensional databases”, in: Proc. of the 10th Int. Conf. on Scientific and Statistical Database Management, p. 63–72, 1998.

Malinowski E., Zima´nyi E., “OLAP hierarchies: A conceptual perspective”, in: Proc. of the 16th Int. Conf. on Advanced Information Systems Engineering, p. 477–491, 2004.

Prat N., Akoka J., Comyn-Wattiau I., “A UML-based data warehouse design method” Decision Support Systems, Volume 42, Issue 3, p. 1449-1473, 2006

Trujillo J. C., Palomar M., Gómez J., Song:I "Designing Data Warehouses with OO Conceptual Models". In IEEE Computer, Vol. 34, N.12, p. 66- 75, 2001.

Spivey J.M., “The Z Notation : a Reference Manual”. Prentice-Hall, 1992.

Saaltink M.. The Z/EVES 2.0 User’s Guide. ORA Canada, OneNicholas Street, Suite 1208, Ottawa (Ontario), K1N 7B7, 1999.