Extending the UML for Designing Association Rule Mining Models for Data Warehouses

José Jacobo Zubcoff[1] and Juan Trujillo[2]

[1] Departamento de Ciencias del Mar y Biología Aplicada. Universidad de Alicante (Spain)
Jose.Zubcoff@ua.es
[2] Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante (Spain)
jtrujillo@dlsi.ua.es

Abstract. Association rules (AR) are one of the most popular data mining techniques in searching databases for frequently occurring patterns. In this paper, we present a novel approach to accomplish the conceptual design of data warehouses together with data mining association rules, allowing us to implement the association rules defined in the conceptual modeling phase. The great advantage of our approach is that the association rules are specified from the early stages of a data warehouse project and based on the main final user requirements and data warehouse goals, instead of specifying them on the final database implementation structures such as tables, rows or columns. Finally, to show the benefit of our approach we implement the specified association rules on a commercial data warehouse management server.

Keywords: Data Warehouses, UML extension, conceptual modeling, multidimensional modeling, Data Mining, KDD, Association rules.

1 Introduction

The discovery of interesting association relationships among huge amount of data is called Association Rule (AR). The aim of the AR is to provide an observation a posteriori of the most common links between data. As an example, a customer selects items from those offered by a retailer (market basket). The retailer keeps a registry of every customer transaction and AR lets us know the relationships between the items that customers are purchasing. This is also referred as Market Basket Analysis.

Data warehouses (DW) store historical, cleansed, subject oriented and integrated data extracted from various sources (Fig. 1). Analysts have to collect all business concerned data to implement a data mining (DM) algorithm. Thus, DW is a perfect framework to apply DM. Typically, DW analysts use multidimensional (MD) models to represent the information they manage. In MD models, data is organized into facts (the aim of analysis) and dimensions representing the context where we wish to analyze facts. Therefore, data mining could be modeled in an MD environment.

In order to take advantage of the effort in the modeling process of DW and to enable the potential of algorithms, rules and modeling techniques, to develop in an extended and well-known standard modeling language, we propose to integrate the AR Mining process into DW modeling, extending the UML with a profile.

A Min Tjoa and J. Trujillo (Eds.): DaWaK 2005, LNCS 3589, pp. 11 – 21, 2005.

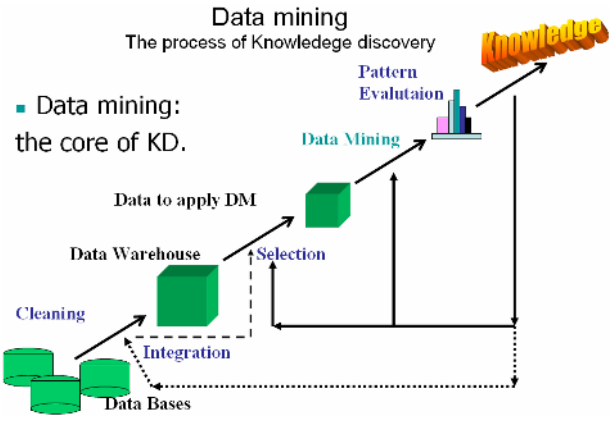

Fig. 1. The process of knowledge discovery

Our proposal is an extension of UML for designing AR Mining models for DW, based on modeling process presented in "Extending the UML for Multidimensional Modeling" [1] which is a UML profile[1] for MD Modeling, that provides the specific mechanisms for conceptual model DW, because it allows us to consider main MD modeling properties and avoids having to learn a new specific notation or language.

The remainder of this paper is structured as follows: Section 2 explains other works that have dealt with association rules, DW and modeling. Section 3 summarizes the conceptual MD modeling on which we are based. Section 4 proposes the new UML extension for designing AR models for MD modeling. Section 5 presents a case study and applies our UML extension for designing AR models for MD modeling, section 6 sketches some further implementation issues. Finally, section 7 presents the main conclusions and introduces immediate and future work.

2 Related Work

The mining concept of discovering AR was introduced in [2]. Early proposals used flat files as input data not in a DW environment. Another important contribution was to provide an SQL-like language with a set of primitive operations to support ad-hoc and interactive data mining integrated into databases, specifically extending SQL to support mining operators. DMQL [12], M-SQL [11] and Mine Rule[13] are SQL-like languages. In [3] there is a comparison between the query languages. All efforts to develop a language for DM allow us to integrate the mining process into databases, not in a separated statistic analysis tool, but as a process in a DB framework.

[1] A *profile* is a set of improvements that extend an existing UML type of diagram for a different use. These improvements are specified by means of the extendibility mechanisms provided by UML (stereotypes, properties and restrictions) in order to be able to adapt it to a new method or model.

A Pattern Base Management is proposed in [9], this is a system that stores and manages patterns as data is managed by a database management system. It is a novel issue in knowledge discovery, and it has been further modeled in [10], not in DW context but as an isolated system that manages and stores patterns. On the other hand, several data mining approaches have been proposed to run on top of data warehouses. An algorithm for mining in a star schema or multidimensional model was proposed [3], [17]. In [3], [5], [6] mining was applied in relational databases as a sequence of SQL queries, this represents an important advance of the relationship between AR and databases. Another proposal was [18] which represents techniques that improve performance using SQL. At metadata level, in the Data Mining chapter of the Common Warehouse Model (CWM) [7] there is a specification that describes metadata interchange among DW and business intelligence knowledge management. This approach consists of a number of meta-models that represents common warehouse metadata in the major areas of interest to DW and BI, including the mining process. But, in all these approaches the only benefit is that they work with huge data previously cleaned. These works do not take into consideration main final user goals of the corresponding multidimensional model underneath, so they do not use important terms in DW such as facts, dimensions or classification hierarchies.

Therefore, we argue that there is still a missing work that allows us the specification of data mining techniques together with the multidimensional modeling accomplished to design data warehouses. Furthermore, the sooner this is specified (from the early stages of a DW project such as the conceptual modeling phase), the better the specified data mining techniques will be focused on final user needs.

3 Object-Oriented Multi-dimensional Modeling

In this section, we outline our approach to DW conceptual modeling, based on the UML [1], [8], [14], [19], specified by means of a UML profile that contains the necessary stereotypes in order to carry out conceptual modeling successfully [15]. The main features of MD modeling are: the many-to-many relationships between the

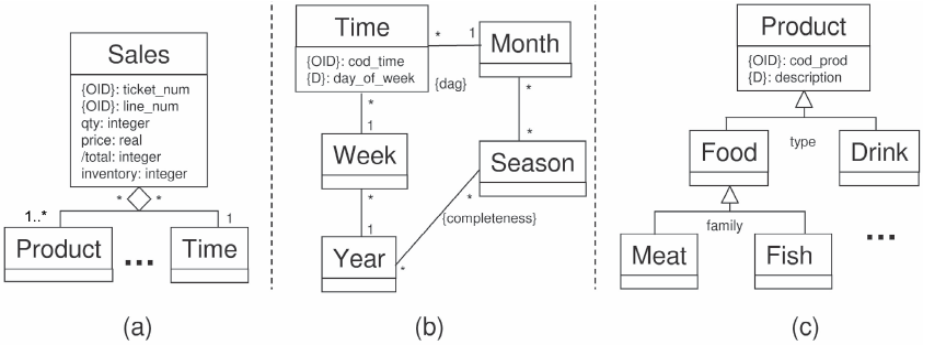

Fig. 2. Multidimensional modeling using the UML

facts and one specific dimension (Fig. 2.a), degenerated dimensions, multiple classification and alternative path hierarchies (Fig. 2.b), the non-strict and complete hierarchies and the categorization of dimensions (Fig. 2.c). In this approach, the structural properties of MD modeling are represented by means of a UML class diagram in which the information is clearly organized into facts and dimensions.

Facts and dimensions are represented by means of fact classes (stereotype Fact) and dimension classes (stereotype Dimension) respectively. Fact classes are defined as composite classes in shared aggregation relationships of dimension classes. The minimum multiplicity in the role of the dimension classes is 1, to indicate that all the facts must always be related to all the dimensions. The relations *many-to-many* between a fact and a specific dimension are specified by means of the multiplicity 1..* in the role of the corresponding dimension class.

4 UML Approach for Association Rule Modeling

Association Rules (AR) look for relationships between items in an entity. This mining process depends on identifying frequent item sets in data, grouped by an entity called Case. Frequent item sets could be used to summarize association between items in the selected attributes. We obtain rules using one or more attributes as input, and one or more attributes to predict. Consequently we have association rules with some input attributes that predicts others (or the same), based on the grouping condition (Case). Consequently, Case from MD point of view could be an –stereotype OID- attribute of a Dimension class, because facts could be aggregated by any dimension or a –stereotype OID- of a Fact class (degenerated Dimension). Input and Predict attributes could be selected from OID attributes (Fact or Dimension) or Fact attributes –stereotype FA- or Dimension attributes –stereotype DA- and rules could have several Input and/or Predict attributes. Case, input and predict are elements of a class called Association Rule Mining Model.

Rules are obtained considering specific settings. Basic settings are minimum support (MinSupp) and minimum confidence (MinConf) that rule must satisfy, max number of records of the item set (MaxItemSetSize) and max number of predicates (MaxNumberOfPredicates) (max number of appearances in the body of the rule). These four settings represent the class Association Rule Mining Settings.

Finally, AR has an antecedent (body) and a consequent (header). The first contains input attribute/s value/s, the last contains predicted attribute/s value/s observed in the grouped item set. AR Mining Results class is built with these four attributes.

Summarizing, AR mining process is represented using three classes: AR Mining Model (ARMM) that is the model, AR Mining Settings (ARMS), and finally AR Mining Results (ARMR) that contains the results of the process, the rules.

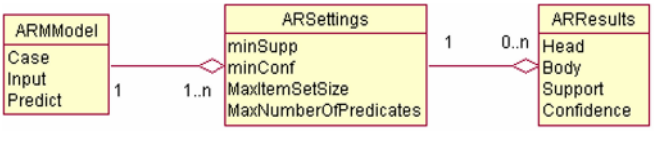

Fig. 3. Association Rule Metamodel

Figure 3 shows the proposed model in which we could see that one AR Mining Model could have several settings, and for each setting we get a set of results.

All conditions take part of Association Rules Settings class (ARSettings) in AR mining model (Fig. 3). Head and Body of the rule are defined as attributes of Association Rule Results class (ARResults), as shown in Fig 3. Each rule obtained has support and confidence, probabilities values that could be represented in percent values, defined as attributes of ARResults class in meta-model.

This proposal allows us to work with attributes from dimensions and facts class of a multidimensional model, to obtain distinct types of association rules: single and multi-dimensional rules, single and multiple predicates rules, hybrid-dimension and inter-dimension association rules, single or multiple-level rules, or any other rules.

There is a "type constraint": association rule mining applied to DW needs the data under study (Input and Predict) to be discrete. Then quantitative continuous data must be categorized. This is a precondition that introduces a constraint in the model.

Domain expert can provide some additional constraints on the rule pattern to be mined, so that generated rules are of interest to the user and more specific and useful. User could restrict to generate rules about a concrete case of an attribute using OCL.

According to [4], an extension to the UML begins with a brief description and then lists and describes all of the stereotypes, tagged values, and constraints of the extension. In addition to these elements, an extension contains a set of well-formedness rules. These rules are used to determine if a model is semantically consistent with itself. According to this quote, we define our UML extension for AR mining conceptual MD modeling, following the schema composed of these elements: *description, prerequisite extensions, stereotypes / tagged values, well-formedness rules, and comments.*

4.1 Description

This UML extension reuses a set of stereotypes previously defined in [1], and defines a set of tagged values, stereotypes, and constraints, which enables us to create AR mining integrated into MD models. The 18 tagged values we have defined are applied to certain components of the MD modeling, allowing us to represent them in the same model and on the same diagrams that describe the rest of the system. These tagged values will represent the participation in the AR structure of the different elements of the MD modeling (fact class, dimension class, base class, attributes, etc.), allowing us to specify how rules will be obtained depending on this mining model structure information and on the value of attributes of the model. A set of constraints are specified in order to define well-formedness rules. The correct use of our extension is assured by the definition of constraints in both natural language and OCL [16].

4.2 Prerequisite Extensions

This UML profile reuses stereotypes that were previously defined in another UML profile in [1]. This profile provided the needed stereotypes, tagged values, constraints to accomplish the MD modeling properly, allowing us to represent main MD properties of DW's at the conceptual level. To facilitate the comprehension of the UML profile we present and use in this paper, we provide a summary of the specification of these stereotypes in Table 1.

Table 1. Stereotype from the UML profile for conceptual MD modeling [19]

Name	Base Class	Description
Fact	Class	Classes of this stereotype represent facts in a MD model
Dimension	Class	Classes of this stereotype represent dimensions in a MD model
Base	Class	Classes of this stereotype represent dimension hierarchy levels in a MD model
OID	Attribute	Represent OID attributes of Facts, Dimension or Base classes in a MD model
Fact-Attributes	Attribute	Attributes of this stereotype represent attributes of Fact classes in a MD model
Descriptor	Attribute	Represent descriptor attributes of Dimension or Base classes in a MD model
Dimension-Attribute	Attribute	Represent attributes of Dimension or Base classes in a MD model
Completeness	Association	Associations of this stereotype represent the completeness of an association between a Dimension class and a Base class or between two Base classes

Table 2. Tagged values defined in the Profile

Tagged Values of the Model			
Name	**Type**	**M[1]**	**Description**
Classes	Set(OclType)	1..*	It specifies all classes of model.
Tagged Values of the Class			
Name	Type	M	Description
Case	{ OID }	1	It specifies the case that an instance of this class uses to group by. The default attribute level Case tagged value is OID.
Input	Set({D,OID,DA, FA})	1..*	It specifies a set of Inputs of the AR. The default attribute level Input tagged value is D case exists, otherwise, OID.
Predict	Set({D,OID,DA, FA})	1..*	It specifies a set of Predicts class used to obtain AR. The default attribute level Predict tagged value is D if exists, otherwise, OID.

Tagged Values of the Attribute			
Name	Type	M	Description[2]
Case	{ OID }	1	
Input	Set({D,OID,DA,FA})	1..*	
Predict	Set({D,OID,DA,FA})	1..*	

Tagged Values of the Instance			
Name	Type	M	Description
Case	{ OID }	1	It specifies the case of an instance
Input	Set({D,OID,DA,FA})	1..*	It specifies a set of Inputs for this instance.
Predict	Set({D,OID,DA,FA})	1..*	It specifies the set of Predicates for an instance

Tagged Values of the Constraint			
Name	Type	M	Description
Involved-Classes	Set(OCLType)	0..1	Classes that are involved in a rule, to be enforced in the constraint
Case	{ OID }	1	It specifies the attribute that is the Case of the itemsets.
Input	Set({D,OID,DA,FA})	1..*	It specifies a set of Inputs of the AR
Predict	Set({D,OID,DA,FA})	1..*	It specifies a set of Predicts of the AR
MinSupp	Double	0..1	It specifies the minimum support of the AR
MinConf	Double	0..1	It specifies the minimum confidence of the AR
MISS	Integer	0..1	It specifies the maximum item set size of the rule
MNOP	Integer	0..1	It specifies the maximum number of predicates of the rule

[1] M stands for Multiplicity

[2] Due to space constraints, we do not include the descriptions of the tagged values of attributes as they are similar to their counterpart tagged values of classes.

4.3 Tagged Values

In this section, we provide the definition of several tagged values for the model, classes, attributes, instances and constraints.

Table 2 shows the tagged values of all elements in this extension. We must set only one Case Tagged Value of the Class for each mining model. The Case tagged value of the class could be defined for a Fact class or for a Dimension class. At attribute level the default Case tagged value must be OID –stereotype OID-. Default tagged values of the attributes of Input and Predict are descriptor attribute -stereotype D- in case they exist, or OID -stereotype OID- otherwise. We must select at least one Input and at least one Predict tagged attribute of the class for each Case tagged value of the class. This means that we could have rules with more than one input and several predict attributes. We could set other than D attribute or OID attribute (from Fact or Dimension class) putting a tagged value of Input or Predict of attribute close to the desired one. If more than one OID exists in a class it is mandatory to set the corresponding tagged value of the attribute. Default value of MinSupp is 0.01 and default value of MinConf is 0.40. Default values of MISS and MNOP are 2000 and 3 respectively. These are attributes of association rule mining settings class.

4.4 Well-Formedness Rules

We can identify and specify in both natural language and OCL constraints some well-formedness rules. These rules are grouped in Table 3.

Table 3. Well-Formedness constraints

- Correct type of the tagged values:
The Case tagged value should be defined for an OID attribute of a Fact or Dimension class of the model
context Model **inv** self.classes-> forAll(a | a.attributes ->forÄll(c | c.Case) -> notEmpty() implies self.attribute.oclIsTypeOf(OID))
- Relationship between Input and Predict tagged values of classes and respective tagged values of its attributes:
If is not defined Input and Predict tagged values for an attribute is settled to D defined for its class.
context Model **inv** self.classes->forAll(c| c.Input ->forÄll(a | a.attributes.Input)->isEmpty() implies a.attribute.oclIsTypeOf(D)=Input) forAll(c | c.Predict->forÄll(a | a.attributes.Predict)->isEmpty() implies attribute.oclIsTypeOf(D)=Predict)
- Categorization of continuous values of an Input and Predict tagged value of attribute
Input and Predict Tagged values must be Type of Integer or must be discrete
context Model **inv** self.classes-> forAll(a | a.attributes ->forÄll(p | p.Predict) -> notEmpty() implies self.attribute.oclType(Integer)) or self.attribute.oclType(Set(String))) self.classes-> forAll(a | a.attributes ->forÄll(p | p.Input) -> notEmpty() implies self.attribute.oclType(Integer)) or self.attribute.oclType(Set(String)))

4.5 Comments

In addition to the previous constraints, the designer can specify association rules with OCL. If the Input, or Predict values of a class or an attribute depends on the value of an attribute of an instance, it can be expressed as an OCL expression (see Fig. 5).

Normally, AR constraints defined for stereotypes of classes (fact, dimension and base) will be defined by using a UML note attached to the class instance[2] corresponding to the Case of the rule (only one Case for Note). We do not impose any restriction on the content of these notes in order to allow the designer the greatest flexibility, only those imposed by the tagged values definitions.

5 A Case Study

The goal of our proposal is to model AR mining in dimensional modeling framework. Typical example is Market Basket analysis modeled as star-schema where Sales are the Fact class and Product, Time and Customer are dimension classes.

To discover rules in a MD model we have to select the case we want to analyze relationships of. To obtain associations of products in a market basket, we have to set a basket as Case with ticket number as key to group by, and select Input and Predict attributes, (*SubCategory.Description)* as Input and Predict from Product dimension to

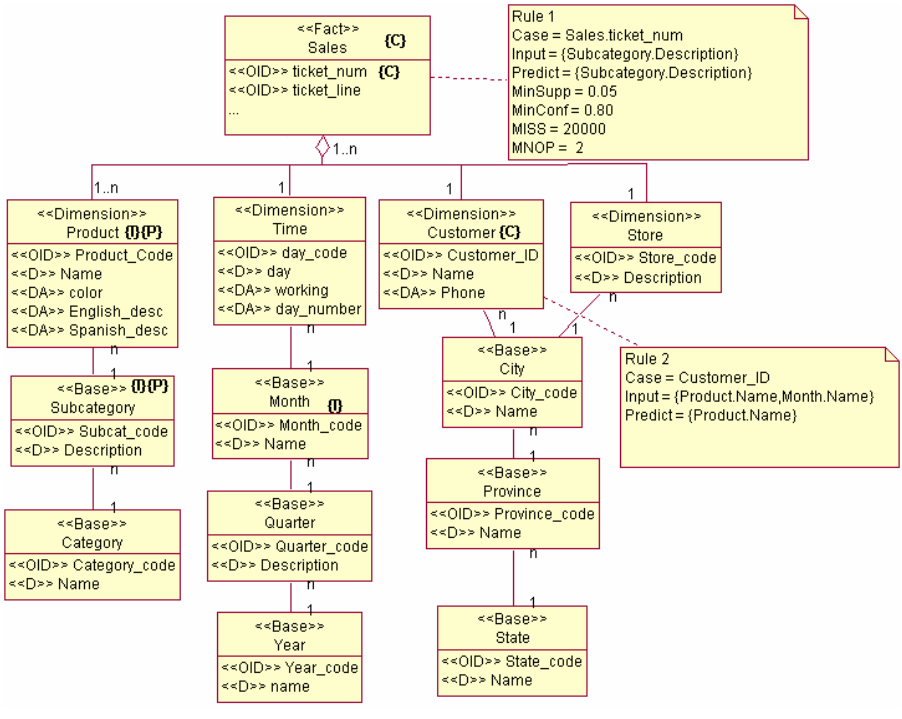

Fig. 4. Example of multidimensional model with AR information and constraints

[2] The connection between a note and the element it applies to is shown by a dashed line without an arrowhead as this is not a dependency [21].

predict which subcategories are related in a market basket. Fig. 4 shows an MD model that includes a fact class (*Sales*), four dimensions (*Product, Time, Customer* and *Store*) and eight base classes (*Subcategory, Category, Month,Quarter, Year, City, Province* and *State*). *Sales* fact class has two OID -stereotype *OID-*: one is ticket_num, which is a degenerated dimension, the other one is *ticket_line*. Remember that Fact attributes could be used as Case if they are previously categorized.

Adding corresponding tagged values we could obtain different association rules from this model. Two AR constraints have been specified as notes in Fig. 4, each note is attached to the class that contains the corresponding Case tagged value:

Rule 1 uses *Sales.Ticket_num* -stereotype *OID-* as Case marked as –tagged value *C-* and *Subcategory.Description* -stereotype *D-* marked as -tagged value *I* and *P-* as Input and Predict respectively, the *MinSupp* is 0.05. The *MinConf* is 0.80, *MISS* as a maximum of 20000 frequent itemsets and *MNOP* of 2, are the association rule mining settings. Rules obtained could be "*If helmet and mountain-bike then road bikes (0.52, 0.3)"* which means if a customer buys helmet and mountain-bike also buy road bikes in the 52% of the cases and that this holds in 30% of the transactions.

Rule 2 uses *Customer.ID* -stereotype *OID-* as Case, *Product.Name* -stereotype *D-* and *Month.Name* -stereotype *D-* as Input -tagged value *I* - and *Product.Name* as Predict -tagged value *I* and *P-*. The *MinSupp* is 0.01, *MinConf* is 0.40, *MISS* as a maximum of 2000 frequent itemsets and *MNOP* of 3, by default. Rules obtained could be "*If helmet and September then Road Tire (0.9, 0.04)"*.

6 Implementation

The model was implemented in SQL Server 2005 (Beta version). Analysis Services is a component of SQL Server 2005 which allows us to implement DW with the concepts of a MD model. Based on the model used as Case study (Fig. 4) we have created the Fact table *Sales*, the dimensions (*Product, Time, Customer* and *Store*) and their hierarchies (*Subcategory, Category, Month, Quarter, Year, City, Province* and *State*). Finally we have defined the association Rules. To define the rule 1 in the previous section, we consider Customer dimension as Case, that means that case key is stereotype OID of customer, and Subcategory of Product as Input selecting *EnglishProductSubcategory* as key of a nested table as Fig. 5 shown. In SQL Server predict means that will be used as Input and Predict, otherwise use *Predict only*.

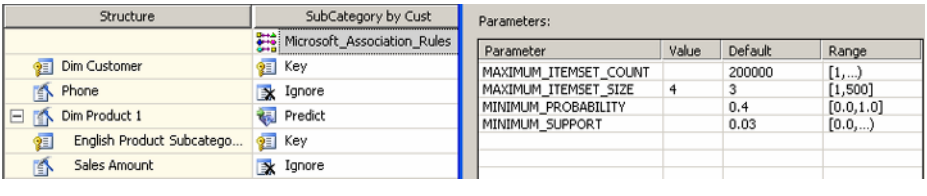

Fig. 5. Model structure example in Analysis Services

7 Conclusions and Future Work

In this paper, we have presented an extension of the UML that allows us to model Association Rules in the conceptual multidimensional modeling of DW. Thus, the defined AR are directly related to the main final user needs and goals of the data warehouse. To achieve this, we have provided the needed stereotypes, tagged values and constraints that allow us to represent AR in multidimensional objects such as facts, dimensions, classification hierarchy levels and so on. To show the benefit of our approach, we have applied our extension to a market basket analysis case study. Finally, we have also shown how the information represented with our approach is implemented on a commercial data base management server with data mining facilities such Microsoft SQL Server 2005 (Beta version). In this way, all AR defined with our approach in the MD modeling at the conceptual level are directly implemented in the DW framework. Our immediate future work is to align our approach with the Model Driven Architecture (MDA) and to extend our profile to represent other data mining techniques rather than just association rules.

References

[1] S. Luján-Mora, J. Trujillo and I. Song. *Extending the UML for Multidimensional Modeling*. In Proc. 5th International Conference on the UML'02, vol 2460 of LNCS, pages 290-304, Dresden, Germany. September 2002. Springer-Verlag.

[2] R. Agrawal, T. Imielinski and A. Swami. *Mining Asociation Rules between Sets of Items in Large Databases*. In Proc. ACM SIGMOD 93. pages 207-216. Washington DC. 1993.

[3] M. Botta, J. Boulicaut, C. Masson and R. Meo. *A Comparison between Query Languages for the Extraction of Association Rules*. DaWaK 2002: 1-10

[4] E. Ng, A. Fu, K. Wang. *Mining Association Rules from Stars*. ICDM (IEEE) Maebashi TERRSA, Maebashi City, Japan December 9 - 12, 2002, pages 322-329.

[5] L. Dehaspe and L. Raedt. *Mining association rules in multiple relations*. In Proc. of the 7th Workshop on ILP, vol. 1297, pages 125–132, Prague, Czech Republic, 1997.

[6] S. Nestorov, N. Jukić .*Ad-Hoc Association-Rule Mining within the Data Warehouse*. Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)

[7] OMG: *CWM Common Warehouse Metamodel Specification*. http://www.omg.org.

[8] OMG: *UML Unified Modeling Language Specification* 1.5. 2004.

[9] S. Rizzi et al. *Toward a logical model for patterns*. In Proc. ER Conference, pages 77-90, Chicago, 2003.

[10] S. Rizzi. *UML-Based Conceptual Modeling of Pattern-Bases*. In Proc. 1st Int. Workshop on "Pattern Representation and Management (PaRMa'04), Crete, Greece, March 2004.

[11] T. Imielinski, A. Virmani. *MSQL: A Query Language for Database Mining*. Data Mining and Knowledge Discovery, 3. 1999: 373-408

[12] J. Han, J. Fu, W. Wang, K. Koperski, O. Zaiane. *DMQL: A Data Mining Query Language for Relational Databases*. In SIGMOD'96 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada, 1996.

[13] R.Meo, G.Psaila, and S.Geri. *A new SQL-like operator for mining association rules*. In Proceedings of the 2nd Int'l Conference on Very Large Databases, India. September 1998

[14] J. Trujillo, et al., *Designing Data Warehouses with OO Conceptual Models.* IEEE Computer, special issue on Data Warehouses, 2001(34): p. 66-75.
[15] M. Gogolla and B. Henderson. *Analysis of UML Stereotypes within the UML Metamodel*. 5th Int Conf. on the UML- The Language and its Applications. 2002. Dresden, Springer,
[16] J. Warmer and A. Kleppe. *The Object Constraint Language Second Edition. Getting Your Models Ready for MDA*. 2003: Addison Wesley.
[17] H. Günzel, J. Albrecht and W. Lehner. *Data Mining in a Multidimensional Environment*. ADBIS 1999: 191-204
[18] H. Cokrowijoyo, D. Taniar . *A framework for mining association rules in Data Warehouses* . IDEAL 2004: 159-165
[19] S. Luján-Mora, J. Trujillo, I. Song: *Multidimensional Modeling with UML Package Diagrams*. ER 2002: 199-213