

Pattern Mining over Star Schemas in the Onto4AR Framework

Cláudia Antunes

Instituto Superior Técnico / Technical University of Lisbon
Av. Rovisco Pais 1,
1049-001 Lisboa, Portugal
claudia.antunes@ist.utl.pt

Abstract

Storing data according to the multidimensional model, in particular following star schemas, has demonstrated to be one of the most adequate forms to ease the exploration of data. However, this exploration has been limited to be query-based, leaving the discovery of hidden information to a second plan. The main reason for this, relates to the inability of traditional mining techniques to deal with several data tables at the same time. In this paper, we propose a new approach to mine patterns among data stored as a star schema, based in a domain driven framework, where available knowledge is represented in a domain ontology. Pattern mining is performed by an apriori-based algorithm – the D2Apriori, but more efficient algorithms are being implemented and tested, in order to solve performance issues related with the large amount of data stored in data warehouses.

1. Introduction

The growing interest in data mining and its maturity have contributed to enlarge its application areas. Indeed, this enlargement got several new challenges into the arena, like dealing with complex data, but also old ones, like the need to incorporate background knowledge into the mining process [8].

The use of semantic contexts for frame the mining process is emerging, and tries to adapt existing mining techniques to implement informed methods. With the use of domain knowledge is then possible to focus the mining process “on regions of the trade-off curves (or space) known (or believed) to be most promising” [3], which result on information that is more likely to fulfill user requirements. This knowledge mostly has been used as constraints, but can be used in a more wide sense, namely in the recognition of occurrences of similar but non-equal items.

Complex data presents different issues to the data mining algorithms; among them is its representation in a non-tabular form. In its traditional specification, mining algorithms deal with a set of records in just one table. When dealing with complex data, like for example time series or molecules (graphs), data does

not fit in the tabular schema and implies adapted methods to mine it.

Star schemas are an optimized model for storing data for analysis purposes. They comprise a set of data tables connected to each other, through another table – the *fact table*. Research and applications on the last two decades prove the importance of such models to enhance the exploration of data. However, that exploration is just query-based, through OLAP queries. To this date, in general, mining methods had to denormalize the different data tables in order to create a large one, which could then be mined, using traditional mining methods.

In this paper, we propose a domain driven method to find frequent patterns stored according to a star schema. The proposed method works in the context of the *Onto4AR* framework [2]. This framework provides an environment for introducing existing domain knowledge to guide the mining process, by using a domain ontology. We show that is then possible to discover existing patterns, at the different levels of abstraction stored in the star model, dealing effectively with multiple inheritance.

The rest of the paper is organized as follows: next (section 2), we overview the area of pattern mining and the use of constraints in this context. In section 3, we describe the *Onto4AR* framework, introducing the basic notions related to ontologies and knowledge bases (section 3.1). In section 4, star schemas are explained and mined in the framework: we show that identified patterns cover the ones found by traditional mining algorithms in a denormalized table, and some others not discovered by those methods. The paper ends with the discussion of the difficulties of applying the method on very large data stores.

2. Pattern Mining and Constraints

The discovery of association rules was first introduced in 1993 [1], and is defined as the discovery of “all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence respectively”. With an association rule corresponding to an implication of the form $A \Rightarrow B$, where A and B are propositions (sets of

pairs attribute/value, most of the times named items), that expresses that when A occurs, B also occurs with a certain probability (the *rule confidence*). The *support* of the rule is given by the relative frequency of transactions that include A and B , simultaneously.

Algorithms for this task run in two steps: first, they identify the set of propositions that occur together – usually called patterns, and then, they generate all possible association rules from the identified patterns. Among the most well known algorithms for transactional pattern mining is the *Apriori* [1]. Along with the *minimum support threshold*, it receives a *dataset* composed by a set of *items* – called *itemsets*, corresponding to the recorded individual transactions, and finds a set of itemsets that occur frequently in the dataset – *set of patterns*.

The problem was formulated as a constrained problem, with support restricting the number of discovered patterns. Probably due to this nature, the study of constraints on the area has begun early with the specification of several different interestingness measures and other constraints to filter the discovered patterns [3]. With them, it is possible to both reduce the number of discovered patterns and improve the performance of the algorithms, by pruning uninteresting ones.

3. The *Onto4AR* framework

By exploring the advances in the area of knowledge representation, ontologies began to be used to support the mining process. In particular, the *Onto4AR* framework [2] provides an environment for mining patterns in the presence of domain knowledge, enabling the user to control the mining process. It defines a formal environment for incorporating

background knowledge into the process of discovering association rules that is independent of the problem domain and the nature of the data.

Before proceeding with its overview, the basics of ontologies and knowledge bases used in the framework are reviewed.

3.1. Ontologies and knowledge bases

The development of the Semantic Web contributed considerably to advance the area of ontologies, and now they are commonly accepted as a mean to represent and share existing knowledge.

An Ontology is an explicit specification of a conceptualization [4], which means that it is a specification of an abstract, simplified view of a domain. Formally, an ontology is a 4-tuple $O := \{C, \mathcal{R}, \mathcal{H}^C, \mathcal{A}^O\}$. C is a set of concepts, which represent the entities in the domain and \mathcal{R} is a set of attributes for concepts, including relations among concepts. \mathcal{H}^C is a taxonomy or concept-hierarchy, which defines *is-a* relations among concepts: $\mathcal{H}^C(c_1, c_2)$ means that c_1 is a sub-concept of c_2 , or in other words c_2 is a parent of c_1 . Finally, \mathcal{A}^O is a set of axioms that describe constraints on the ontology, making explicit implicit facts [6].

In the counterpart of ontologies are knowledge bases, which specify existing instantiations for a particular ontology. Formally, a knowledge base is a 4-tuple $\mathcal{KB} := \{O, I, inst, instr\}$, where O is an ontology; I a set of instances; *inst* is a function from C to 2^I called *concept instantiation*, and *instr* the *relation instantiation* function defined from \mathcal{R} to $2^{I \times I}$.

Note that while ontologies try “to capture the conceptual structures of a domain of interest” [6], by defining its elements and describing the relations

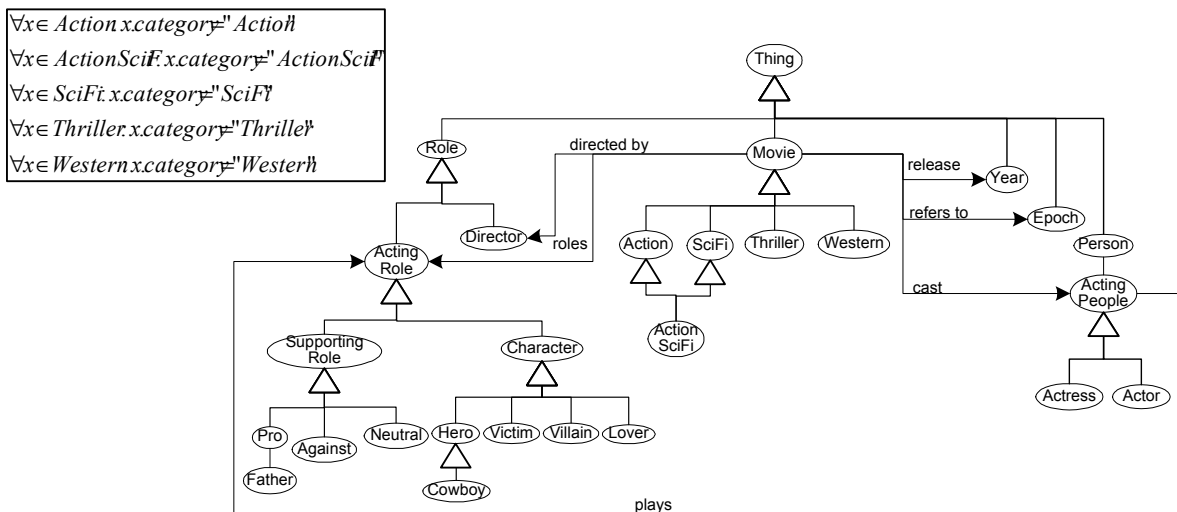


Figure 1 – Ontology in the cinematographic domain

among them, a knowledge base defines a set of elements, which can be understood in that context.

Figure 1 illustrates a simple ontology describing basic knowledge in the cinematographic domain. The schema is represented in UML, where concepts are represented by ellipses (for example *Actor* is a concept, and *Thing* represents the concept, from which all other concepts descend), *is-a* relations as non-named arrows (for example *ActingPeople* is a particular case of *Person*) and other relations by named strong arrows (for example *plays* for example). Instances for those concepts are lines in the dimension tables presented in Figure 6.

Finally, the five axioms in the ontology specify the conditions for a movie belongs to each category (*Action*, *SciFi*, *Thriller*, *Western* and *ActionSciFi*). *Equal* axioms are not needed, since two instances are equal if and only if all of their features are equal.

3.2. Problem Formulation in the Onto4AR

The *Onto4AR* framework is centered on the use of an ontology and assumes a new formulation of the problem, where the meaning of an item is clearly defined in the context of the ontology.

As for transactional pattern mining, algorithms receive a dataset composed by a set of items (now called *D2Itemsets*), corresponding to the individual transactions recorded. However, these transactions are known to be framed in some context, given by a *knowledge base*. As stated above, this knowledge base is a tuple with an *ontology*, a set of *instances* and the

correspondence between instances and existing concepts in the ontology. In this context, an itemset corresponds to a set of instances, characterized by a set of *features* (pairs attribute/values), which correspond to traditional items (Figure 2 – bottom).

The result produced by these new algorithms is a set of *patterns*, which correspond to sets of *abstract instances*. These instances are just instances that do not correspond to fulfilled instances, and can exist in theory, but may not exist in the given dataset or other similar one (Figure 2).

Note that this formulation differs from the usual one only in two aspects. First, patterns relate more than simple objects (instances), but specify abstract relations, similar to generalized patterns proposed in [7]. Second, there is no imposition on the number of times that *A* and *B* occur together. All depends on the constraints imposed.

Indeed, the first aspect has been neglected, and the user has to represent the set of transactions at the right abstraction level. Even in the basket analysis problem, items do not correspond to real instances but to some abstraction (when a customer buys a particular beer and consumes it, other customers cannot buy it). In the *Onto4AR* framework, this issue is overcome, since the equality of items is defined outside the algorithm's logic and scope, by axioms in the ontology.

3.3. Mining algorithms in the Onto4AR context

Mining in this new context can be performed by any

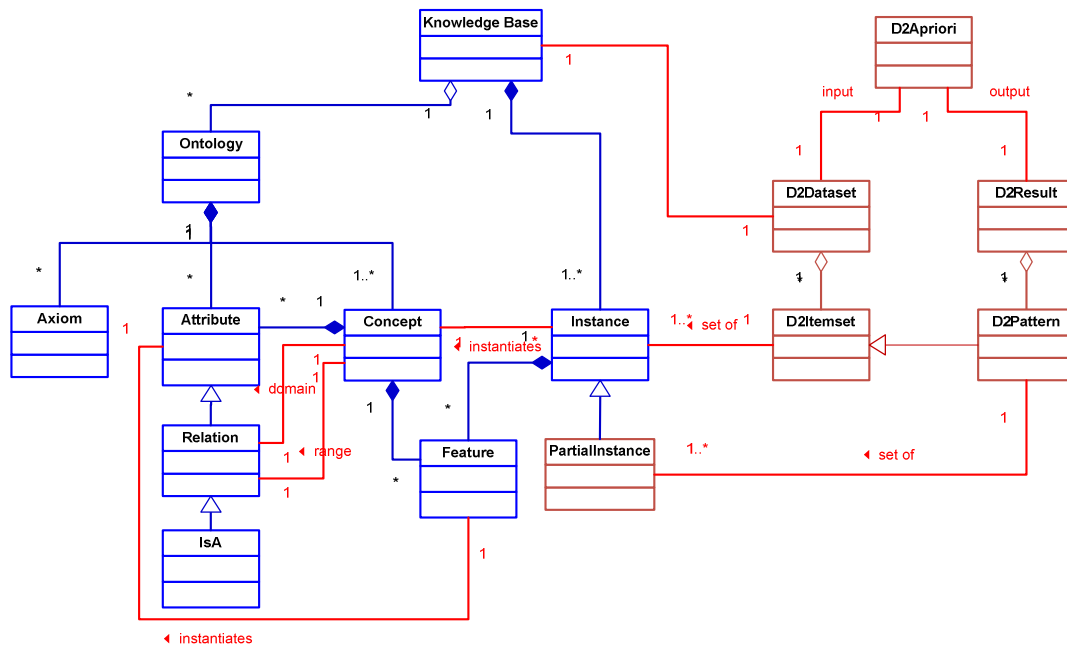


Figure 2 – Onto4AR framework

```

1 Procedure D2Apriori(File data, File KBfile,
2                     Constraint C)
3   KB ← acquireBK(KBfile)
4   D ← readDatafile(data, C)
5   L1 ← {C accepted items in D}
6   k ← 2
7   while (Lk ≠ ∅) do
8     Ck ← candidateGeneration(C, Lk-1)
9     Ck ← antiMonotonicPruning(Ck, Lk-1)
10    Lk ← supportPruning(Ck, D)
11    Ak ← constraintPruning(Lk, C)
12    Lk ← Ak
13    k ← k+1
14  return ∪k Ak

```

Figure 3 – Domain Driven Apriori algorithm

adapted transactional pattern mining algorithm that is able to incorporate the available knowledge, represented by the ontology. However, any of these adaptations have to incorporate some basic procedures to do that. Next, we describe each of these procedures and exemplify their application in an apriori-based algorithm – the *D2Apriori* algorithm (Domain Driven Apriori), illustrated in Figure 3.

Naturally, like any other pattern mining algorithm, the method receives the dataset to mine as input (*data* in Figure 3) and in addition, it receives the knowledge base (*KBfile*) and the constraint to apply (*C* – which at least, corresponds to the parameterization of the support constraint).

The data file is then read in the context of the knowledge base, creating the dataset with the incorporation of the semantics for each read transaction (*D*). For reducing the time spent by the discovery process, some measures can be taken, namely when reading the knowledge base: first read the set of concepts and its attributes, and then propagate the concept properties (attributes, relations and default values) through the *taxonomy* (the set of *is-a* relations). In this manner, each instance inherits explicitly its properties (overriding them whenever is needed), which allows for avoiding the navigation over the entire ontology during the mining step.

4. Mining Star Schemas in the *Onto4AR*

The success of the databases technology is deeply related to the existence of data models underlying the data stores. For decades, the relational model dominated, but it became obsolete in the context of decision support. In this context, the multidimensional model, as been demonstrated to be more efficient, mainly through the use of star schemas [5].

The idea behind the multidimensional model is to reproduce the existing relations at the domain level, avoiding the normalization characteristic of relational model. In particular, it is organized around facts,

related to a set of attributes, arranged according to dimensions. In this manner, it comprises two kinds of data tables: the dimensional and the fact tables.

Dimensions are just sets of attributes that describe a same property (or entity), most of the times with attributes following a hierarchy. Facts only record the occurrence of some event, which happens on the combination of different dimensions.

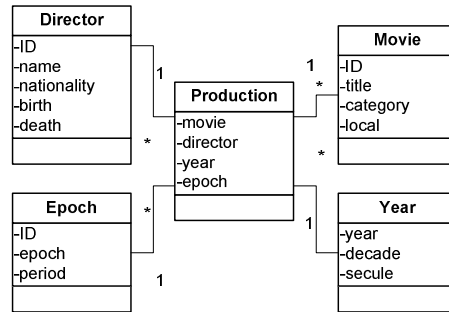


Figure 4 – Star schema for movies production

Figure 4 illustrates a simple star schema for representing data about movies, directors, dates and epochs. The central piece is the fact table (*Production*) which stores the information about each movie (*Movie*), considering its director (*Director*), the release year (*Year*) and the epoch referenced by the movie (*Epoch*). Note, while dimension attributes are mainly descriptive (like *name*, *nationality*, *birth date*), facts are just made of foreign keys to each dimension and usually a set of measures for quantifying the event (not present in our schema).

In the context of *Onto4AR* framework, mining star schemas only requires to have an ontology where each dimension is represented by a concept. Among these concepts, one has to correspond to a *hub* or *key concept*, that is just a concept with at least *n* relations, one for each dimension referenced in the fact table.

In this manner, each itemset in the set of tables that instantiate the star schema (*D2Itemset*) will correspond to the aggregation of facts for the same hub instance.

The ontology illustrated on Figure 1 can be used for mining star schemas in the cinematographic domain since it comprises a concept for each dimension and a hub – the *Movie* concept, since it has relations for all other dimensions, namely *directed by*, *refers to* and *release*. Given this context, consider the dimensions and fact tables created in accordance to the star schema for productions (Figure 4) represented in Figure 6.

The *D2Itemset*

(*Movie* {*Rio Bravo*, *Western*, *USA west*},
Year {1958, 50, 20},
Director {*Howard Hawks*, 1896, 1977, *USA*},
Epoch {*XIX*, *Industrial Revolution*}),

for example, represents the sixth fact in the Production fact table (4506, 1958, 252, 14).

On the other hand, the D2Itemset

(*Movie {Indiana Jones, Action, Europe}*,
Actor {Harrison Ford, Harrison Ford, male, USA, 1942},
ActingRole {Hero, Indiana Jones},
Actor {Thomas Sean Connery, Sean Connery, male, UK, 1930},
ActingRole {Father, Henry Jones})

is an example of the cast for the movie *Indiana Jones*, which should correspond to two facts in the Cast star schema (Figure 5). In this second case, the dimension *Movie* is used to aggregate the different facts in the fact table, resulting in a larger itemset (with more instances).

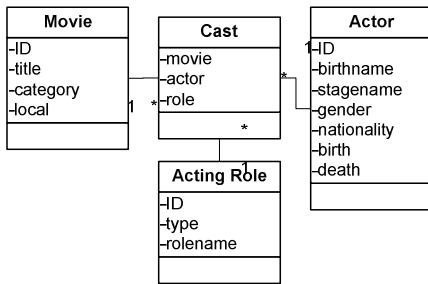


Figure 5 – Cast star schema

Note that it is possible to have more than one hub concept for a pair ontology / star schema. In this case, it is necessary to choose the one for aggregating the facts, and several distinct pattern analyses can be performed.

The cast star schema is such a case, with all involved concepts being hubs, as long as we consider the inverse relations for *roles*, *cast* and *plays*. Using movies as aggregation, patterns would relate mostly movies categories with the presence of some acting roles. For example: “*Action movies usually have an hero and a villain*”. Considering *Acting People* it is possible to find patterns describing the kind of acting roles played by some particular actor. For example: “*John Wayne is usually a cowboy in Western movies*”.

With this new representation for itemsets, it is then possible to address two important issues for pattern mining over star schemas: to consider different length itemsets and to deal with inheritance and multiple inheritance.

The first issue occurs in transactional pattern mining, since the usual solution is to denormalize the star in a single flat table. By using itemsets as

aggregations of facts, it is possible to consider several different instantiation of some concepts together. Secondly, in the new context it is possible to recognize the instances of sub-specialization concepts as particular cases of more generic ones, allowing for a correct support counting.

Consider the dataset represented in Figure 6 again, and a minimum support threshold of 20%. Since there are only fifteen itemsets (15 movies and facts), it corresponds to a minimum frequency of three occurrences. Among the patterns discovered by *D2Apriori*, are intra-dimensional and inter-dimensional patterns like the two following ones, respectively:

(*Movie {Western, USA west}*)

(*Movie {Thriller}*,
Director {Alfred Hitchcock, 1899, 1981, UK}).

All of them are discovered by traditional approaches over denormalized tables. However, the new method allows for the discovery of other patterns that in the previous context were not considered frequent.

(*Movie {SciFi, space}*)

is an intra-dimensional example, discovered by our approach. Indeed, there are three *SciFi* movies: *Star Wars* and *Alien*, but also *Superman* and *Total Recall*, since these last ones are Action *SciFi* movies, a known specialization of *SciFi* pictures in the context of the given ontology (Figure 1).

This is achieved in a simple way, by propagating the default values through the hierarchy of classes. The default value “*SciFi*” for the attribute *category* for *SciFi* instances is inherited by any instance of specialization of *SciFi* concept, namely *ActionSciFi*. By allowing the existence of multiple values for the attribute *category*, is then possible to discover patterns involving *SciFi* and *Action* movies that were not identified with previous existing tools.

In terms of the *D2Apriori* algorithm, this is done during the dataset reading (line in 4 in Figure 3), where each *D2Itemset* is created for each aggregation of facts. In this step, the axioms in the ontology are used to determine which concept is instantiated by each dimensional instance. Whenever some instance instantiates a concept that descends from another one, it inherits the default values of its parents. In this manner, instances may have a variable number of features, assuming the different inherited behaviors.

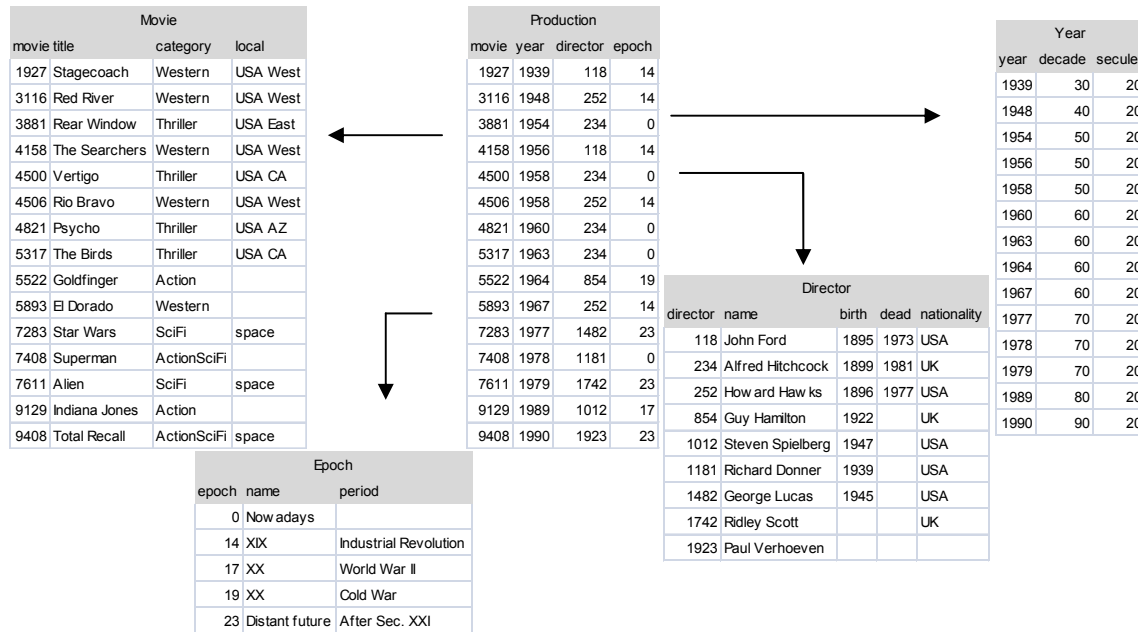


Figure 6 – Dimension and fact tables for star schema Movies production

5. Conclusions

Mining in the presence of the semantic context for data, becomes to be one of the most challengeable issues in the knowledge discovery process, mostly due to the advances in the area of knowledge representation. Indeed, ontologies begun to be used worldwide, and they can be both used by humans and by information systems. In the area of information systems, it is now generally accepted, that these formalisms are a key to share and reuse the existing domain knowledge, and in the last years they become to be considered in the mining process. The *Onto4AR* framework is one of the mining approaches that try to incorporate available domain knowledge into the pattern mining process, through the use of ontologies.

In this paper, we have explored the referred framework to solve another unsolved issue in the area of pattern mining – the discovery of information in data stored following a star schema. This is achieved, by making use of the knowledge about the hierarchy of concepts, overcoming the results achievable through traditional pattern mining approaches.

Despite *D2Apriori* algorithm fulfill our goal, it is clear that it won't be able to deal with real stars. This is true, either in the context of a domain driven approach or in the traditional formulation, essentially due to memory consumption resulting from the explosion of both generated candidates and patterns discovered. Note that the memory consumed by the knowledge base is residual facing the amounts of data stored in a data warehouse. New pattern-growth based algorithms

are being developed and tested for mining stars, achieving very interesting results.

6. References

- [1] Agrawal, R., Imielinsky, T., and Swami, A. Mining Association Rules between Sets of Items in Large Databases. *Proc ACM SIGMOD Conf Management of Data*, 207-216. Washington DC, USA. 1993
- [2] Antunes, C. An Ontology-based Framework for Mining Patterns in the Presence of Background Knowledge". *Int'l Conf Advanced Intelligence (ICAI 08)*, 163-168. Post and Telecom Press. Beijing, China. 2008
- [3] Bayardo, R.J., The Many Roles of Constraints in Data Mining. *SIGKDD Explorations*, (4): 1, i-ii. ACM Press 2002.
- [4] Gruber, T.R., A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisitions*, (5):2, 21-66. Academic Press, 1998
- [5] Kimball, R. and Ross, M., *The Data warehouse Toolkit - the complete guide to dimensional modeling*, Wiley. 2002.
- [6] Maedche, A., *Ontology Learning for the Semantic Web*, Kluwer Academic Publishers, 2002.
- [7] Srikant, R., Vu, Q., and Agrawal, R. Mining Association Rules with Item Constraints. *Proc Int'l Conf Knowledge Discovery and Data mining (KDD'97)*, 67-73. Newport Beach, USA. ACM Press, 1997
- [8] Yang, Q., Wu, X., 10 Challenging Problems in Data Mining Research. *Int'l Journal of Information Technology & Decision Making*, (5): 4, 594-604. World Scientific Publishing Company, 2006