Toward Intelligent Data Warehouse Mining: An Ontology-Integrated Approach for Multi-Dimensional Association Mining

Chin-Ang Wu[1], Wen-Yang Lin[2,*], Chang-Long Jiang[3], Chuan-Chun Wu[4]

[1]Dept. of Information Engineering, I-Shou University, Kaohsiung County, Taiwan 840, R.O.C., Email: cwu@csu.edu.tw

[2]Dept. of Computer Science & Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan 811, R.O.C., Email: wylin@nuk.edu.tw

[3]Dept. of Electrical Engineering, National University of Kaohsiung, Kaohsiung, Taiwan 811, R.O.C., Email: m0955107@mail.nuk.edu.tw

[4]Dept. of Information Management, I-Shou University, Kaohsiung County, Taiwan 840, R.O.C., Email: miswucc@isu.edu.tw

Abstract

A data warehouse is an important decision support system with cleaned and integrated data for knowledge discovery and data mining systems. In reality, the data warehouse mining system has provided many applicable solutions in industries, yet there are still many problems causing users extra problems in discovering knowledge or even failing to obtain the real and useful knowledge they need. To improve the overall data warehouse mining process, we present an intelligent data warehouse mining approach incorporated with schema ontology, schema constraint ontology, domain ontology and user preference ontology. The structures of these ontologies are illustrated and how they benefit the mining process is also demonstrated by examples utilizing rule mining. Finally, we present a prototype multidimensional association mining system, which with intelligent assistance through the support of the ontologies, can help users build useful data mining models, prevent ineffective

[1] Chin-Ang Wu is also a lecturer in Cheng Shiu University, Niaosong Township, Kaohsiung County 833.
[*] Corresponding author. Tel: +886-7-5919517, Fax: +886-7-5919514

pattern generation, discover concept extended rules, and provide an active knowledge re-discovering mechanism.

Keywords. Data mining, data warehousing, intelligent assistance, multidimensional association rule, ontology

1. Introduction

With the fast and massive accumulation of data in the current information era, especially under the pervasion of the Internet, the utilization of knowledge discovery and data mining technology plays a key role in promoting business competition and improving scientific discoveries. To prevent ineffectual efforts of garbage in, garbage out computation, contemporary knowledge discovery platforms usually accommodate the data warehouse (Inmon, 1995) as a consistent and integrated data repository. As with the typical data warehouse mining process shown in Figure 1, the data in a data warehouse is extracted from multiple and heterogeneous data sources, undergoes a complex cleaning and integration process, and finally loads into the data warehouse as a consistent and integrated data repository. In this way, users do not need to worry about the heterogeneity and the consistency of the data.

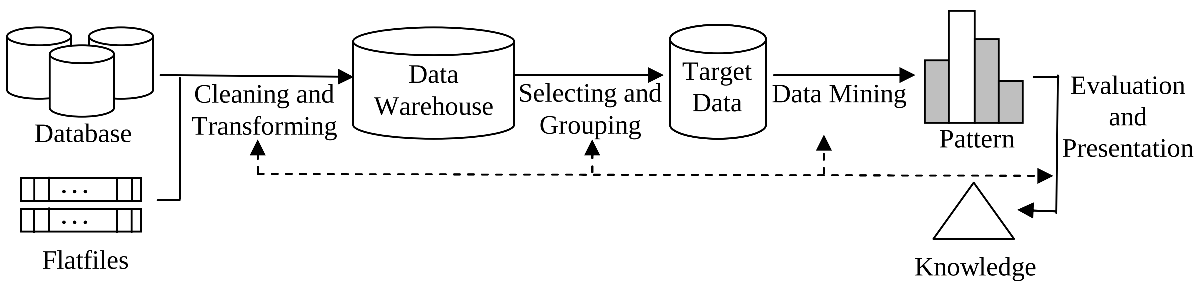

Figure 1. The process of data warehouse mining

In the past decade, much effort has been devoted to data warehouse mining and many applicable solutions have been provided. J. Han's research group conducted data mining from data cubes and multi-dimensional databases (Han, 1998). They developed DBMiner (Han et al., 1997), a system combining OLAP (Chaudhuri & Dayal, 1997) and data mining to provide association mining, classification, prediction and clustering. Many commercialized database systems, such as Oracle and Microsoft's SQL Server, also provide data mining from data warehouses or data cubes. These data mining systems provide an integrity data mining environment allowing users, based on their subjective need, to formulate data mining models (queries) interactively. These systems check the legality of mining model structures, yet not the rationalities [rationales?] of them; the users have to tune the mining models repeatedly until they obtain satisfactory results. This problem is exacerbated when mining multidimensional association rules from data warehouses.

For example, consider the star schema (Kimball et al., 1996) in Figure 2. The following mining model, whose structure will be defined later, represents a user's query for looking at associations among customer's daily purchase of products in Japan.

Transaction ID: *CustID*, *TimeID*

Interested mining attribute: *ProdName*

Condition before grouping: *Country*= "*Japan*".

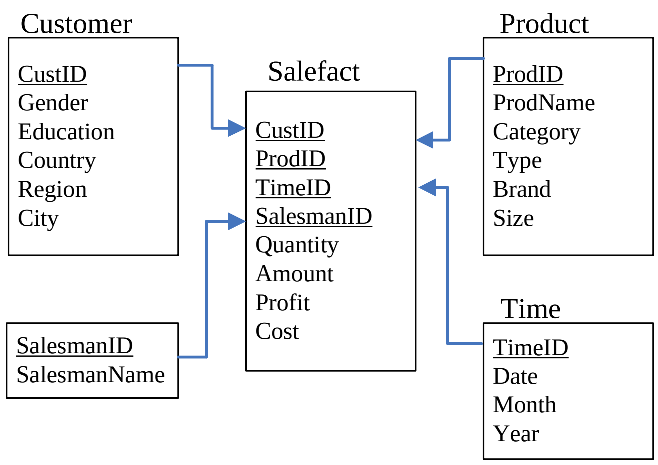

Figure 2. An example of sales star schema

The specification of this example mining model includes deciding on the data granularity (Transaction ID), selecting the pertinent mining attributes, and if necessary, setting the filtering condition before and/or after the grouping process and providing essential information to the system to prepare the target data as depicted in Figure 3. Unfortunately, as will be defined later, the mining model specification is a rather complicated task for novice mining users; many unreasonable mining model settings would be specified. Moreover, some users may even not know how to initiate a mining model. Some of the settings are syntactically legal but semantically unreasonable. Below are some examples of problematic mining models for further illustration.

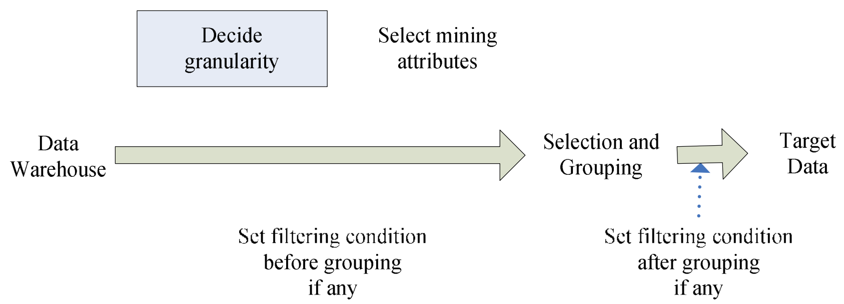

Figure 3. Data selection and grouping

Example 1:

Transaction ID: *Category*

Interested mining attribute: *CustID*, *Gender*, *Education*

This is not a sensible mining model due to the improper inclusion of customer ID as a mining attribute. It will create numerous tedious itemsets such as,

CustID = “*C001*” $\Rightarrow$ *Education* = “*High School*”.

Example 2:

Transaction ID: *Gender*

Interested mining attribute: *ProdName*

This mining model will yield only two transactions, not statistically sufficient to generate convincing rules.

Unfortunately, current data warehouse mining systems do not usually provide any assistance in the formation and semantic checking of the users’ mining model settings. We argue a good mining system should provide intelligent assistance in crystallizing a user’s mining intention into a good mining model that is syntactically correct and semantically reasonable. Although data warehouses have solved the data preprocessing problems effectively, there is at least the following issues hindering the realization of such an intelligent assistance by the data warehouse mining system:

(1) Lack of semantic portrayal of data

Contemporary data warehouse systems are mostly based on the relational data model organized in the form of star schema, which cannot present full relationships between data. For example, the star schema in Figure 2 simply shows the structural relationships between the fact table and the dimension tables. It overlooks the concept hierarchical relationships between dimensional attributes, as shown in Figure 4, which provides essential information for performing typical OLAP operations, such as roll-up, drill-down, and slice-and-dice. If a data warehouse mining system can provide the users with the concept hierarchical relationships between attributes, the users can perform better data mining. As well as the concept hierarchical relationships, some data constraints beyond the current data warehouse mining systems exist. For example, *Gender* is not suitable as the only mining attribute. Such constraints can be helpful in the user's mining model setting.

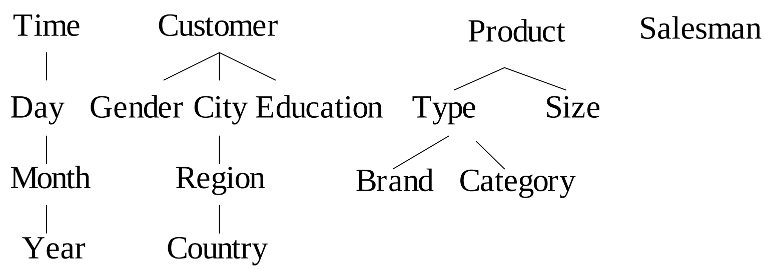

Figure 4. Conceptual hierarchical relationships

(2) Lacking facilities in understanding users' mining intentions

Data mining process is more or less a subjective process that depends on the individual's intention. A user's mining intention is represented by his or her mining model setting to obtain a set of mining results. The satisfactory mining requests conducted previously can be an important asset to provide rich information for sharing with other users. For example, they can be further analyzed to provide recommendations for other users to refer. However, the current data warehouse mining systems provide very few mechanisms in capturing and analyzing the users' previous mining request.

To overcome the above deficiencies of current data warehouse mining systems, we contend integrating background knowledge is necessary. Specifically, the providing of concept hierarchical and constraint relationships between attributes can avoid some settings of unreasonable mining models. This reduces the iterative mining processes and further saves the processing cost. Also, maintaining previously conducted mining processes provides recommendations while the user is setting a mining model. This useful information helps the users clarify their real mining intention. In addition, some domain related knowledge beyond the data warehouse can be helpful to the users for setting a more specific mining scope and also extending the rules derived.

As the concept of ontology has emerged into an effective method for domain knowledge representation and sharing (Uschold & Gruninger, 1996; van Elst & Abecker, 2002), we adopt the ontology technique to construct the above mentioned knowledge into the following four groups: (1) schema ontology: describes the schema structure and relationship between the attributes of the data warehouse; (2) schema constraint ontology: describes the constraints between attributes; (3) domain ontology: collects the related domain and expert knowledge; and (4) user preference ontology: integrates the derived common mining models. With the support of these ontologies, we will show in this paper that a data warehouse mining system can provide intelligent assistance to facilitate more efficient and effective mining. Specifically, we take multidimensional association mining as an example to show how a data warehouse mining system can provide intelligent assistance through the support of the ontologies. We will demonstrate such a system, from the user's perspective, possesses the following advantages:

(1) Helps the users clarify what they need. In other words, it assists the users in building better mining models by providing semantic checking and model recommendations. This avoids the generation of useless patterns, thus the findings of useful patterns can be achieved successfully.

(2) Finds concept extended rules from the existing primitive data warehouse with the help of the proposed domain ontology.

(3) Allows the mining constraints to be set more precisely by including relationship information defined in the domain ontology.

The rest of this paper is organized as follows. In Section 2, we will introduce the multidimensional association rule mining and define its mining models. In Section 3 the framework of intelligent data warehouse mining system incorporating ontologies is presented followed by the introduction of each ontology. How these ontologies can benefit the intelligent mining system is also illustrated with examples of multidimensional association rule mining. In Section 4, we demonstrate the intelligent assistance through examples. Part of the prototype implementation of intelligent system interfaces is presented in Section 5. In Section 6 we discuss related work and finally we conclude and highlight some future research in Section 7.

2. Multidimensional association rule mining model

An association rule has the form, $A \Rightarrow B$, where A and B are sets of items and $A \cap B = \emptyset$. The rule implies transactions in the data warehouse containing A also tend to contain B. A is the *body* or the *antecedent* of the rule and B is the *head* or the *consequent* of the rule. For this rule to be interesting, A and B have to satisfy the user specified *minimum support* (*ms*) and *minimum confidence* (*mc*). The *support* of the rule, $P(A \cup B)$, measures the percentage of the total transactions containing both A and B. The *confidence* of the rule, $P(B \mid A)$, measures the percentage of transactions containing A and also containing B. If the items in a rule involve only one attribute, it is termed a *single-dimensional association rule*. Further, where the items in a rule involving two or more attributes, it is termed a *multidimensional association rule*. An attribute is also called a dimension from the perspective of a multidimensional data model.

Definition 1. (Multi-dimensional association rule) Consider a transaction table composed of k attributes (dimensions). Let x_{im} and y_{jn} be the values of attributes X_i and Y_j, respectively. The form of a multi-dimensional association rule is:

$$X_1\text{= “}x_{1m}\text{”}, X_2 = \text{“}x_{2m}\text{”}, \ldots, X_{\mathrm{i}} = \text{“}x_{im}\text{”} \Rightarrow Y_1 = \text{“}y_{1n}\text{”}, Y_2 = \text{“}y_{2n}\text{”}, \ldots, Y_j = \text{“}y_{jn}\text{”}$$

Following the work in (Han & Kamber, 2001; Zhu, 1998), the multi-dimensional association rules can be categorized into three types as follows.

(1) Intra-dimensional association rule

This type of rule shows association within an attribute. The items in an intra-dimensional association rule come from only a single attribute. For example, the following intra-dimensional association rule,

ProdName=“*LG DVD Burner*”$\Rightarrow$ *ProdName*=“*DVD-R 8X Disk*”,

shows people purchasing an “*LG DVD Burner*” are also likely to purchase a “*DVD-R 8X Disk*”, involving the only attribute *ProdName*.

(2) Inter-dimensional association rule

This type of rule shows association among multiple attributes. The items in an inter-dimensional association rule involve more than one attribute with each attribute appearing only once in the rule. For example, an inter-dimensional association rule,

Gender = “*Female*”, *ProdName* = “*JVC UX-C305 Hi-Fi System*” $\Rightarrow$ *City*=”*Taipei*”,

involves three attributes with no repetition of any attribute in the rule. This rule indicates the females buying “*JVC UX-C305 Hi-Fi*” are likely to live in the city “*Taipei*”.

(3) Hybrid association rule

This type of rule is a combination of inter-dimensional and intra-dimensional associations. The items in such a rule also originate from multiple attributes, but unlike the inter-dimensional association rule, it allows repeated attributes. For example, the following hybrid-dimensional association rule,

Education = “*College*”, *ProdName* = “*Acer PC*” $\Rightarrow$ *ProdName* = “*HP printer*”,

shows people with college education who purchase “*Acer PC*” tend to also purchase “*HP printer*”. Attribute *ProdName* appears twice in the rule.

Data warehouse stores complete and primitive data. The user has to specify what subset of the data in warehouse is of concern and what conditions should be satisfied through the construction of a mining model defined below.

Definition 2. (Mining Model) Suppose a star schema S containing a fact table F and m dimension tables $\{D_1, D_2, \ldots, D_m\}$. Let T be a joined table from S composed of $a_1, a_2, \ldots, a_r$ attributes, so $\forall a_i \in Attr(D_k)$, $1 \le i \le r$, $1 \le k \le m$. Here, $Attr(D_k)$ denotes the attribute set of dimension table D_k. With t_G, $t_M \subseteq \{a_1, a_2, \ldots., a_r\}$ and $t_G \cap t_M = \varnothing$, a mining model of multidimensional association rules from T is defined as

$$MM: <t_G, t_M, [wc], [hc], ms, mc>,$$

where t_G, t_M, wc, hc, ms and mc are the mining model elements, each of which described as follows is involved in acquiring the target data shown in Figure 3.

t_G : the transaction ID (data granularity),

t_M: the pertinent mining attributes,

wc: the optional filtering condition *before* the grouping operation,

hc: the optional filtering condition *after* the grouping operation,

ms: the minimum support, and

mc: the minimum confidence.

Table 1 is an example based on the data warehouse in Figure 2. From all the customers' daily transactions, a user wants to know if there are any associations between customers' education and the purchased products. In such a case, each transaction in the target data is identified by the composite of the customer ID and the transaction date while the mining attributes are education and product. Therefore the mining model will be $t_G = \{CustID, Date\}$ and $t_M = \{Education, ProdName\}$. The corresponding target data is shown in Table 1 with six transactions. However, if instead customer ID is specified as the only transaction ID, i.e., $t_G = \{CustID\}$, the target data will be as shown in Table 2. These two target data represent different views with different granularity. Even if a rule can be generated from both datasets, its meaning will differ.

Table 1. An example of target data grouping by CustID and Date

tid	t_G		t_M	
	CustID	**Date**	**Education**	***ProdName**
1	C001	2008-02-01	College	B,C,E
2	C003	2008-02-03	High School	A,B,E
3	C003	2008-02-10	High School	A,D
4	C004	2008-02-05	Middle High	C,E
5	C005	2008-02-09	College	B,C,D,E
6	C005	2008-02-15	College	A,B,E
*A: IBM60GB B: IBM TP C: RAM 512MB D: Ink Cartridge E: Hard Disk				

Table 2. An example of target data grouping by CustID

tid	t_G	t_M	
	CustID	**Education**	***ProdName**
1	C001	College	B,C,E
2	C003	High School	A,B,D,E
3	C004	Middle High	C,E
4	C005	College	A,B,C,D,E
*A: IBM60GB B: IBM TP C: RAM 512MB D: Ink Cartridge E: Hard Disk			

3. System Framework of Ontology-Integrated Data Warehouse for Multidimensional Association Mining

In this section we will introduce the proposed data warehouse mining system incorporating various ontologies to fulfill the function of intelligent assistance in mining processes. The feasibility of such a system in amending the aforementioned deficiencies of modern data warehouse mining systems will be clarified through suitable examples.

Figure 5 shows the proposed data warehouse mining system framework. Some specific knowledge helpful in supporting the system’s intelligent services is incorporated in this framework, including the characteristics of the data warehouse schema, constraint relationships between attributes, domain specific knowledge and the user preference in the mining model setting. This knowledge is beyond the presentation capability of current data warehouse systems and is structured into four different forms of ontologies: (1) schema ontology, (2) schema constraint ontology, (3) domain ontology, and (4) user preference ontology.

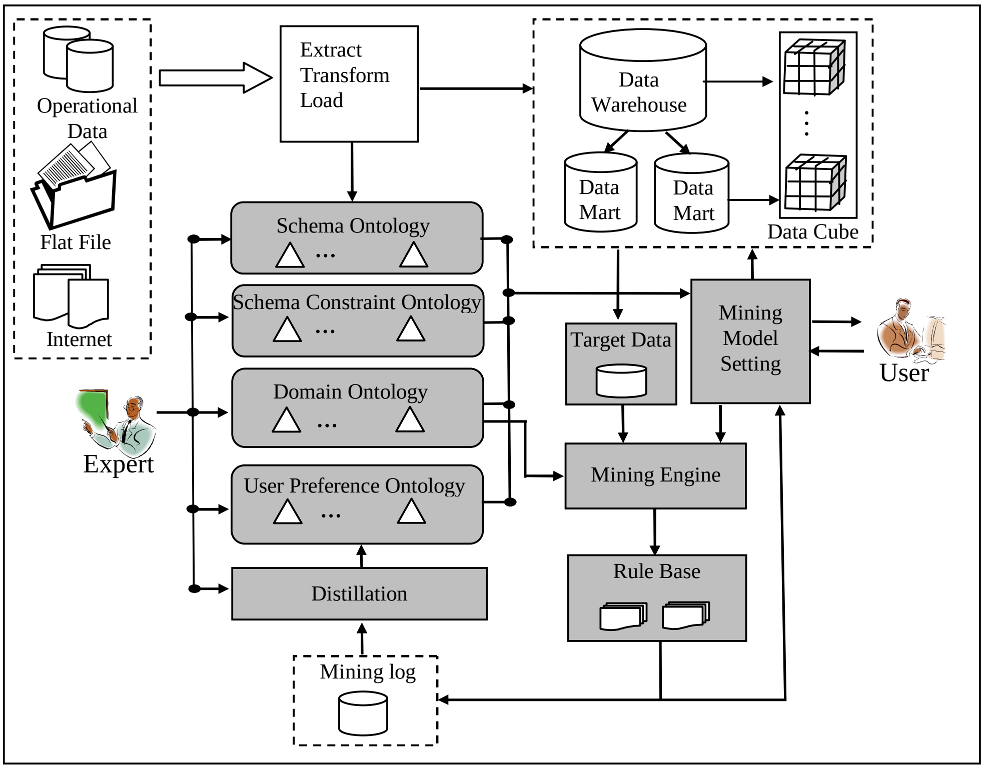

Figure 5. An Ontology-Integrated Data Warehouse for Multidimensional Association Rule Mining

A mining process begins with the setting of a mining model by the user. According to the specified mining model, the target data is prepared and the mining engine is then launched. The user tunes the mining model repetitively until the satisfactory results are found. The model elements of a satisfactory mining are saved in the mining log, providing data for further analysis to closely gather related model patterns. The analyzed results then are utilized to construct the user preference ontology.

The contents of the ontologies except the user preference ontology are preserved by some experts, all of which, as will be clarified later, are applied to assist the users in settings the mining models by performing reasonableness checks and offering element recommendations. The domain ontology also provides the mining engine with further information to find rules with extended concepts. In what follows we will illustrate the details of each ontology.

3.1 Schema ontology

Multidimensional data model shows the inter-relationships of the fact table and dimension tables through the key structures of relational tables. There are other relationships between the dimensions or attributes not shown in the model, yet can benefit the data mining process, including concept hierarchical relationships and different additive characteristics of fact measures in the data warehouse. We use schema ontology to construct such relationships. Figure 6 is an example schema ontology corresponding to the multidimensional data model in Figure 2.

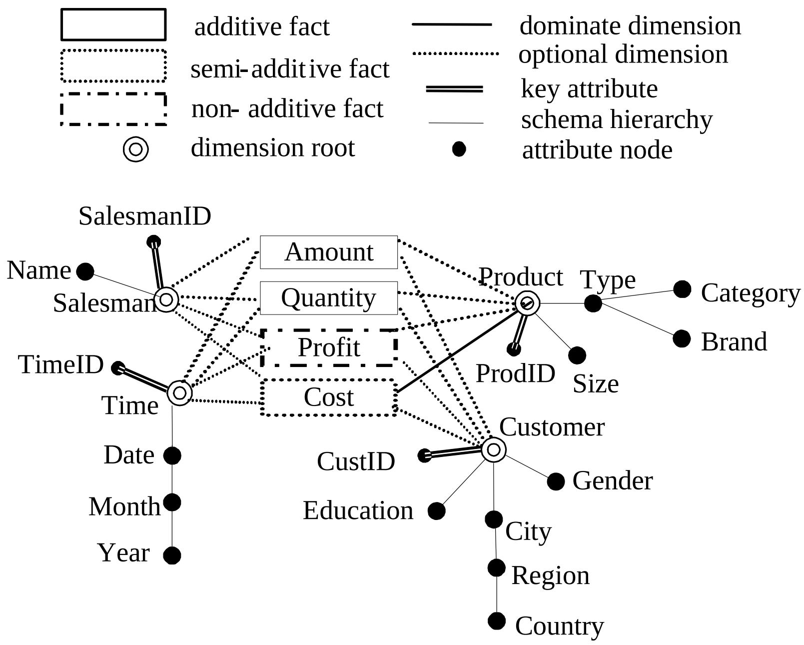

Figure 6. An example of schema ontology

There are three additive types of fact measures for OLAP operation. (1) Additive, which can be summed along all the dimensions. Some typical examples are sale quantity and sale amount which can be summed along all the dimensions. (2) Semi-additive, which can be summed only along certain dimensions. For example, the cost can have sensible aggregation only along product dimension. (3) Non-additive, which will not have any sensible summation along any dimension. For example, profit is calculated by subtracting cost from the sale amount and is a non-additive measure. Another clear example, which is not part of the schema, is the balance amount in a bank statement. It cannot generate any sensible summation for analysis purposes because it is an accumulated amount that is non-additive. The information of the additive types of measures is valuable in that it helps validate user's specification of aggregation. For example, the target data for the mining model with t_G={*CustID*} and t_M ={*Cost*} will generate summation of *Cost* along *CustID*, which is illegal. With the help of information of additive types in schema ontology, the system can detect such ineffective settings for mining.

The hierarchical relationship is important in that it supports the user in better data selection, therefore, it minimizes the infeasible settings of the mining model. For example, according to the schema ontology, the products: "*ASUS EeePC 900*" and "*ASUS EeePC 1000*" have the following concept hierarchy:

$$\begin{tikzcd} & \textit{PC} \text{ (Category)} \\ \textit{EeePC} \text{ (Type)} \arrow[ur] \arrow[dr] & \\ & \textit{ASUS} \text{ (Brand)} \end{tikzcd}$$

If a user knows there are hierarchical relationships between the type of a product and its category, he or she will not try to dig the associations between them. If they do, trivial patterns will be generated because a product type decides its category. For example,

$$Type = \text{“}EeePC\text{”} \Rightarrow Category = \text{“}PC\text{”}$$

is known to people, therefore the mining is redundant.

3.2 Schema constraint ontology

In (Perng et al., 2001; Perng et al., 2002), the authors explored all possible mining spaces of mining attribute combinations under variational transaction IDs and a defined allowed range through data constraints, which can minimize the search space efficiently.

In multidimensional association rule mining, similar constraints imposed on t_G and t_M exist. In addition, we observed some predicates specifically for multidimensional association rule mining. Also, some constraints are domain dependent. For example, t_G ={*Size*} may be valid in the plastic bag business but not in the 3C business. In this paper we present some example constraints for illustration purposes. All these constraint relationships are beyond the representation capability of data warehouses. The schema constraint ontology is used to describe such semantic constraint relationships in general. Figure 7 is an example of a schema constraint ontology derived from the schema in Figure 2. Some attributes have multiple constraints. These constraints are valuable for the system to verify the user specified t_G and t_M of a mining model, and therefore, avoid invalid mining. The following are some examples of constraint relationships related to multidimensional association rule mining.

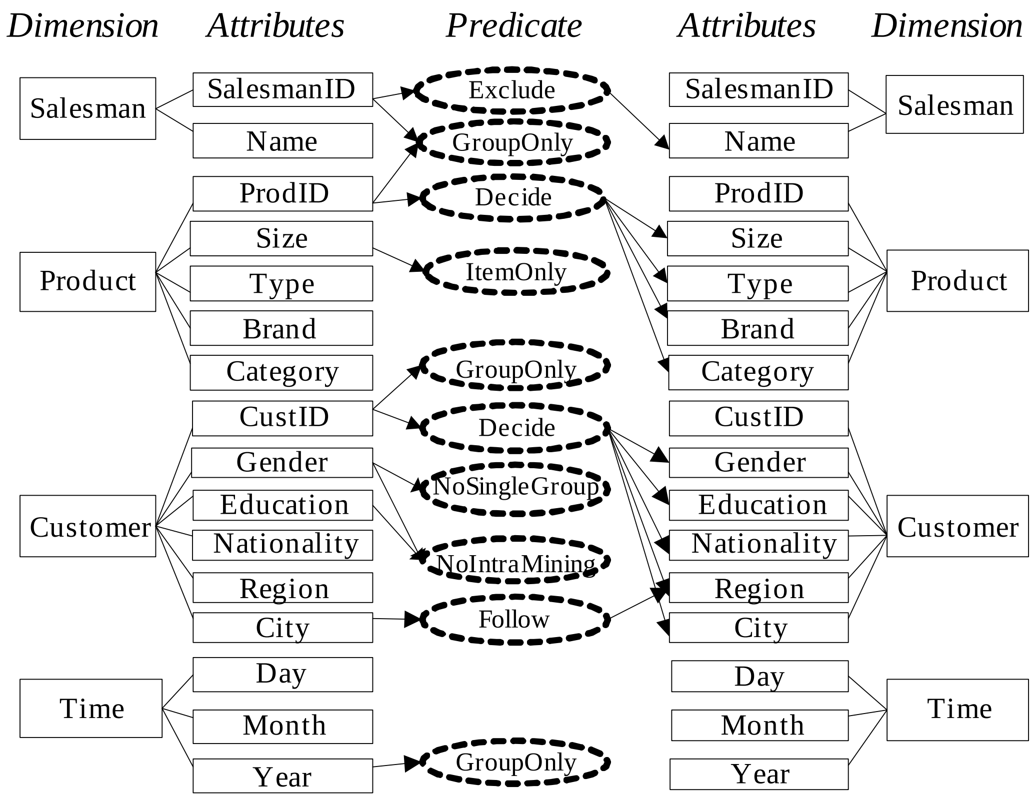

Figure 7. An example of a schema constraint ontology

(1) Decide

This is the relationship analogous to functional dependency. Given two attributes A_1 and A_2, we say A_1 *decides* A_2 if for any two transactions t_1, t_2, if $t_1.A_1 = t_2.A_1$ then $t_1.A_2 = t_2.A_2$. Whenever a functional dependency relationship exists between attributes, redundant mining space or known patterns will be generated and consequently waste the mining efforts. For example, a mining model with t_{G1} = {*CustID*, *Education*} is a redundant form of t_{G2} = {*CustID*} because *Education* is decided by *CustID*. The mining space created for t_{G1} and t_{G2} will be exactly the same. Another example is a mining model with t_M = {*ProdName*, *Brand*} will generate redundant rules because *ProdName* decides *Brand*, therefore associations of *ProdName* and *Brand* are known knowledge. This rule

$$ProdName, \text{“}IBM\ TP\text{”} \Rightarrow Brand, \text{“}IBM\text{”},$$

is an example of a redundant form.

(2) GroupOnly

An attribute with constraint of Is_GroupOnly can be used as a transaction ID only, for example, *CustID*. If *CustID* is chosen as one of the mining attributes, the following tedious rule will be generated:

$$CustID=\text{”}C003\text{”} \Rightarrow Category=\text{”}Printer\text{”}.$$

(3) ItemOnly

This constraint shows an attribute can only be used as an interested mining attribute. For example, *Size* in the 3C Domain is not suitable for deciding the granularity of data.

(4) Follow

Some attributes should be used by following another. For example, a city is located in a region, but different regions could have cities with the same name. Considering the case in the United States, in both states of Tennessee and Ohio, there is a city called Springfield. If the attribute *City* is used alone as a transaction ID, then the results will be misleading. Therefore, ‘*City*’ use is confined by following ‘*Region*’.

(5) NoSingleGroup

An attribute with a constraint of NoSingleGroup cannot be used alone as a transaction ID. For example, *gender* has only two distinct values namely, *male* and *female*. Grouping by *gender* will result in too few transactions to generate any patterns.

(6) NoIntraMining

An attribute with a constraint of NoIntraMining cannot be used alone as an interested mining item. In another word, it cannot be used to mine intra-multidimensional association rule. Examples such as $t_M = \{Gender\}$ *or* $t_M = \{Education\}$, will generate tedious rules like

$$Gender=\text{”}Female\text{”} \Rightarrow Gender=\text{”}Male\text{”}, \text{ or}$$

$$Education=\text{"}College\text{"} \Rightarrow Education=\text{"}High\ School\text{"}.$$

Such types of attributes are adapted to a constraint of NoIntraMining.

(7) Exclude

This constraint prohibits the simultaneous appearance of attributes. For example, assume the calendar date and fiscal date are both in the time dimension. Suppose a fiscal year starts from Sep 1; it is across two calendar years. Then, an example such as using *FiscalYear* and *CalendarMonth* together as the transaction ID would be misleading. The exclude constraint can be used to avoid an invalid combination of transaction ID.

In short, the schema constraint ontology is particularly helpful for checking mining model settings. User specification of t_G and t_M can be verified with the knowledge presented in this ontology. Through the intelligent user interfaces, incorrect or inefficient t_G or t_M can be reminded and adjusted interactively.

3.3 Domain ontology

Domain ontologies are used to construct the domain expert knowledge related to the mining subject of the data warehouse. The contents are specific to certain dimensions, such as product dimension or customer dimension, to enforce the expression of their semantic relationships. Following the W3C recommendation (Heflin, 2004), a domain ontology can be constructed in the triple format of a "Subject-Relationship-Object". For example, the following triplet indicates the hard disk is a component of the PC:

$$PC \xrightarrow{Has\_component} Hard\ disk,$$

where the subject *PC* has a *Composition* relationship with the object *Hard disk*.

Figure 8 is an example domain ontology of 3C products exploring the classification (is-a) and composition (has-a) relationships between products. Other relationships, such as the product features or compatibilities of software to OS or memory to motherboards etc, can be included.

One of the advantages of introducing domain ontology into the mining system is to find concept extended rules from the existing primitive data warehouse. For data with hierarchical relationships, some previous research mentioned extension of association rules (Domingues & Rezende, 2005; Han & Fu, 1995; Lisi & Malerba, 2004; Srikant & Agrawal, 1995) from primitive into a generalized or leveled format. Current data warehouse systems cannot reveal possible extended relationships between data values. For example, the hierarchical relationship in the product dimension of Figure 4 describes only the category and brand of products and their types, yet some possible concept extensions between products such as those presented in Figure 8 are not disclosed. Domain ontology can describe such conceptual relationships relevant to each product.

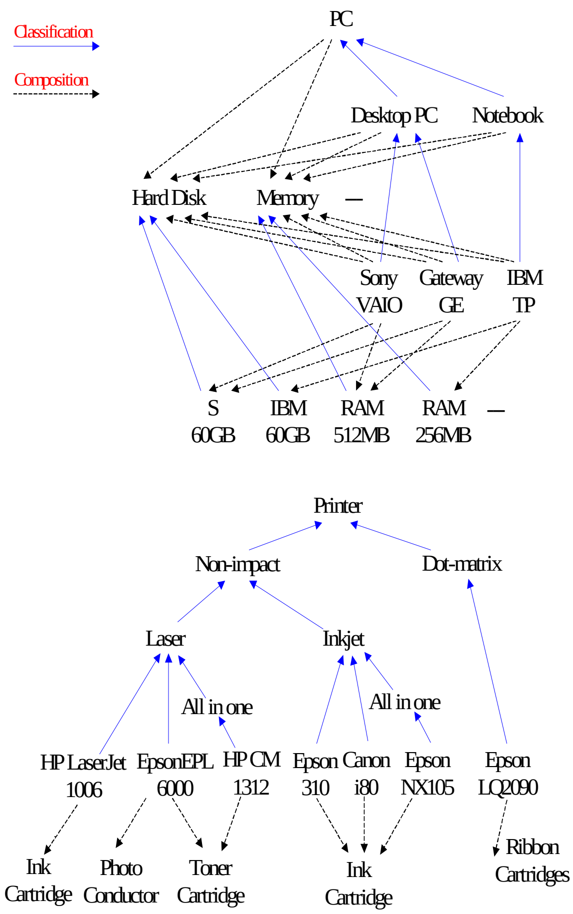

Figure 8. An example of a 3C domain ontology

Besides, different product may have different levels of conceptual hierarchies. For example, in Figure 8 we can see *Epson LQ2090*, *Epson EPL 6000* and *Epson NX105* have one, two and three classification levels respectively. Therefore, it is possible to find extended rules by incorporating the domain ontology with data warehouse mining. For example, in Figure 8, "*PC*" and "*Notebook*" are the generalized classifications of "*IBM TP*" while "*RAM 256MB*" and "*IBM 60G*" are its compositions. The following rule,

ProdName="*HP DeskJet*" $\Rightarrow$ *ProdName*="*IBM TP*",

reveals most customers buying "*HP DeskJet*" tend to also buy "*IBM TP*". According to the composition relationships in the 3C domain ontology, we know *IBM TP* is composed of *IBM 60GB*. Therefore, the following extended rule can be derived:

ProdName="*HP DeskJet"* $\Rightarrow$ *ProdName*="*IBM TP*" *with ProdName*="*IBM 60GB*",

showing customers who buy "*HP DeskJet*" tend to buy "*IBM TP*" with "*IBM 60GB*".

Another motivation of incorporating domain ontology is it helps the user to clarify the scope of the mining target. For example, if a user is only interested in analyzing *PC* components, he can specify a filtering condition excluding all products not satisfying this characteristic by utilizing the composition relationship expressed in the domain ontology. Precisely, the filtering condition of mining model element, *wc*, can be defined to have *ProdName* in the *3C_Domain_Ontology* as follows:

wc = ProdName in 3C_Domain_Ontology ('PC', has_component, var_All)

The predicate *3C_Domain_Ontology* takes the triple of the subject-relationship-object as the parameters, where *var_All* is a variable representing all the products that are components of the *PC*.

3.4 User preference ontology

In a data warehouse mining system, a user manifests his or her mining intention by the mining model settings as defined in Definition 2. As we previously pointed out, a user's mining model setting is a highly interactive process between the user and the system. Some users might not know exactly what they want or how they can initiate mining models. Therefore, it is important for a system to provide the users with intelligent assistance in setting the mining models closer to their intentions. Our theme toward realizing this assistance is to utilize the knowledge of experienced mining users in model settings to provide recommendations for questions like how to select the grouping attributes and the interested mining-items, how to judge minimal support and minimal confidence. This is because experienced users have a good sense of the mining processes, so their knowledge is worth sharing with other users.

Specifically, we log the setting history of mining models satisfying the users, and periodically distill and condense them into the structure of the user preference ontology. A detailed description of the distillation process can be found in (Wu et al., 2009). In short, we employ the association rule mining technique over the mining log to find surrogate patterns representative of frequently used queries in the mining history. For example, consider the following rule:

$$t_G = \{CustID\} \Rightarrow t_M = \{Gender, Education\}.$$

It indicates if $\{CustID\}$ is the transaction ID, the pertinent mining attributes always tend to be $\{Gender, Education\}$. Similarly, we can discover association rules revealing close relationships between certain (t_G, t_M) pairs and where conditions *wc* and/or having conditions *hc*, in the form of

$$t_G, t_M \Rightarrow wc\ [, hc]$$

All surrogate rules mined from the system's mining log are then constructed into the user preference ontology shown in Figure 9, where the attribute indexing level provides connections from each attribute to the corresponding t_G and t_M for easy and rapid access. Under a certain t_G, the t_M with close relationships are grouped into the same node, meaning the set of t_M tends to be used together for a mining. In the same way, the close related filtering conditions, *wc* and *hc*, under certain t_M and t_G can also be grouped together, indicating the users usually apply the *wc* and *hc* with such t_G and t_M settings.

The favorable minimum support and minimum confidence associated with each surrogate pattern are calculated by averaging the *ms* and *mc* in the mining models. For example, the following closely related mining models have different minimum supports and minimum confidences:

Model 1: $t_G = \{CustID\}$, $t_M = \{ProdName\}$, $ms = 45\%$, $mc = 68\%$

Model 2: $t_G = \{CustID\}$, $t_M = \{ProdName\}$, $ms = 72\%$, $mc = 76\%$ and

Model 3: $t_G = \{CustID\}$, $t_M = \{ProdName\}$, $ms = 60\%$, $mc = 62\%$.

These mining models will be grouped together in the user preference ontology with the suggested minimum support $ms = 59\%$ and the suggested minimum confidence $mc = 69\%$. Other statistical summary functions such as mean or mode can be used instead. The user preference ontology can be further connected to the rule base, if any, maintaining the mining results of the mining models to facilitate efficient execution of incremental mining (Chueng et al., 1996) or iterative mining (Liu & Yin, 2001) of the associated mining model. Figure 10 is an example of the user preference ontology.

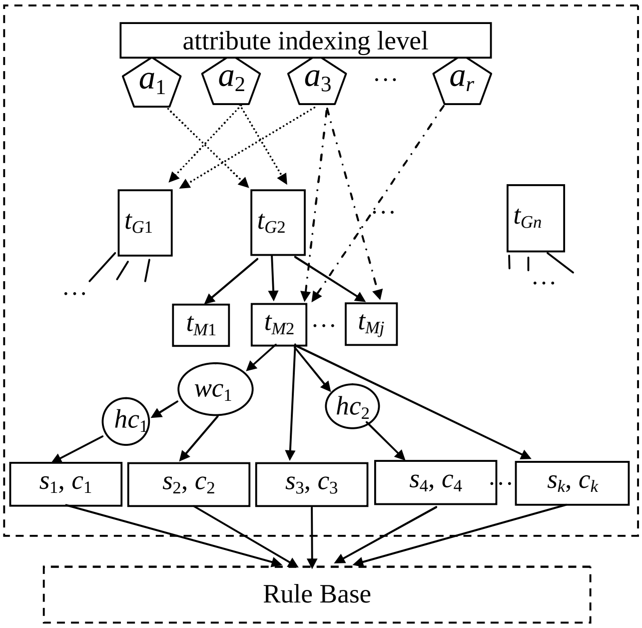

Figure 9. The user preference ontology

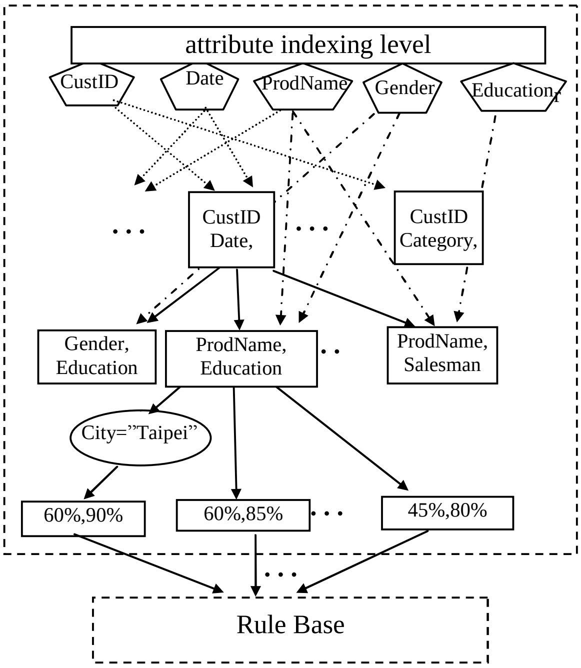

Figure 10. An example of user preference ontology

4. Intelligent Assistance for Mining Model Formulation with Ontologies

Ideally, the user's mining intention can be reflected in the mining model setting. However, without complete comprehension of the schema and domain related knowledge, the end users may develop mining models based on their experiences or intuition. The mining model formulated by the users can possibly be semantically invalid, leading to incorrect or redundant search space or mining results and wasting the mining efforts they have made. For a common mining model formulation interface, a system often provides as many syntactic error checking mechanisms as possible. For example, it provides a popup list or list box for attribute selection to avoid users' typo. However less effort has been made with semantic checking due to the lack of semantic relationship information beyond the data warehouse. Below, we will elaborate on how an intelligent assistance can be built into the proposed data warehouse mining system under the support of ontologies.

Specifically, we will show through the assistance of the ontologies we have introduced, the mining model formulation interface can provide the semantic error detection and mining model element recommendation in an attempt to improve the effectiveness and the efficiency of mining processes. Users can express their mining intention more precisely and even clarify or renew their original mining intentions. The intelligent checking mechanisms proposed in our system framework are shown in Figure 11.

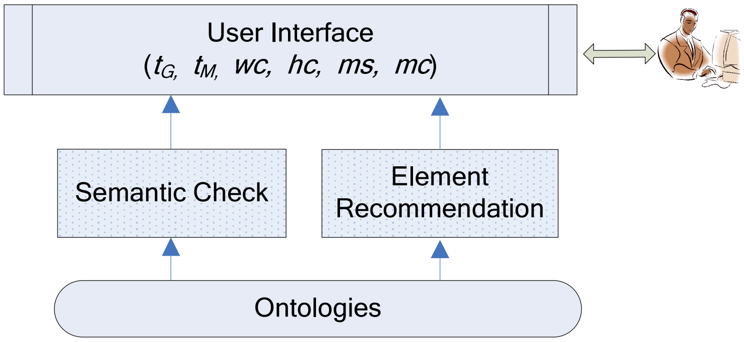

Figure 11. Intelligent assistance in mining model setting

Intelligent assistance in semantic checking

Through the support of schema ontology and schema constraint ontology, the system can provide semantic checking against mining model elements. Figure 12 shows four different results will be displayed to inform the user of the appropriateness of his mining modeling setting, and provide a rationale if errors occur to help the user reformulate the model. Conforming to the definition of the mining model, our system will check the main elements, including data granularity t_G, mining attributes t_M, filtering conditions *wc* and *hc*.

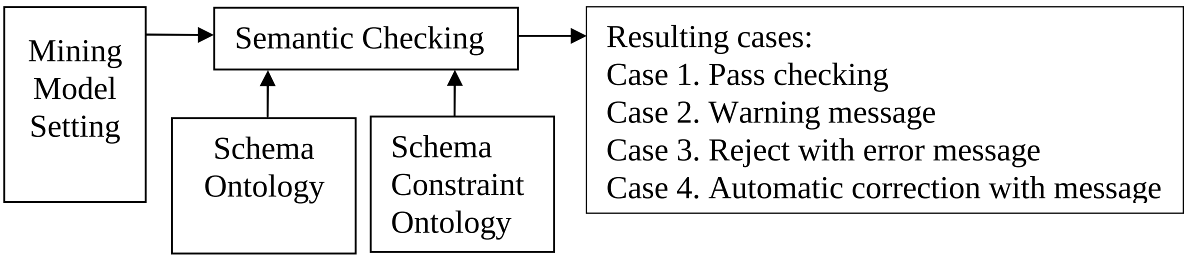

Figure 12. Semantic checking mechanism

(a) Semantic checking of t_G

This checks the semantic legality of the transaction ID for data grouping. The transaction ID set, t_G, represents the data granularity and is the key for the mining transactions. If this question is asked: "What product associations are there between daily customer's purchase?”, then t_G will be {*CustID*, *Date*}; if another question is asked: “From daily product category purchase, are there any associations between customer’s education and gender?”, then t_G will be {*Cateogry*, *Date*}. A user can select t_G based on their needs or interests but may also set it incorrectly because of the lack of semantic understanding. Below are some scenarios of incorrect settings.

Example 1: $t_G = \{Size\}$

The system will reject this setting by semantic checking against the constraint *ItemOnly*(*Size*) in the schema ontology.

Example 2: $t_G = \{Gender\}$

The result of the grouping will be only two transactions, which is too few to generate any rules. This will be rejected according to the constraint *NoSingleGroup*(*Gender*).

Example 3: $t_G = \{CustID, Gender\}$

$t_{G1} = \{CustID, Gender\}$ is actually a redundant form of $t_{G2} = \{CustID\}$ according to constraint *Decide*(*CustID*, *Gender*). The mining space for t_{G1} and t_{G2} is exactly the same. The system will automatically correct the setting of t_{G2} with warning messages.

(b) Semantic checking of t_M

This is the semantic legality check of the user’s interested mining items. This checking specifically verifies if t_M violates any of the constraints in the schema constraint ontology.

Example 4: $t_M = \{Year\}$ or $t_M = \{Year, Education\}$

Tedious rules such as

$$Year = \text{“1999”} \Rightarrow Year = \text{“2000” or}$$

$$Year = \text{“1999”}, Education = \text{“}High\ School\text{”} \Rightarrow Year = \text{“2002”}, Education = \text{“}Elementary\text{”}$$

will be generated. According to the constraint *GroupOnly*(*Year*), the model settings will be rejected.

Example 5: $t_M = \{ProdID, Size\}$

According to constraint *Decide*(*ProdID, Size*), the mining item *ProdID* determines the value of *Size*. This setting will generate known rules since it digs the associations between the product name and its size. The following rule is an example:

$$ProdName = \text{“}IBM\ TP\text{”} \Rightarrow Size = \text{“}17\ inch * 15\ inch * 1\ inch\text{”}.$$

Example 6: $t_M = \{Gender\}$

Tedious rules will be generated such as

$$Gender = \text{“}Female\text{”} \Rightarrow Gender = \text{“}Male\text{”}.$$

The system will reject the senseless setting according to the constraint *NoIntraMining*(*Gender*).

(c) Semantic checking of *wc*

In the mining model setting, the *wc* filtering is operated before grouping data into transactions with transaction ID. The checking of *wc* includes type consistency checking and domain checking.

Example 8: *wc* = (*City* = *'Japan'*)

This example has no problem with type consistency but *'Japan'* is actually not a city, therefore the system will respond with a domain checking warning to the user.

Example 9: *wc* = (*ProdName* ***in*** *3C_DomainOntology* (*'All-in-one', Classification, var_All*))

The 3C domain ontology can be used for filtering conditions. If a user is interested in only the "*All-in-one*" related products in market basket analysis, all the objects with '*classification*' relationship to *All-in-one* in 3C domain ontology should be retrieved. The domain ontology retrieved values are then used for selecting related transactions from the data warehouse.

(e) Semantic checking of (t_G, *hc*)

This function checks the semantic legality of aggregation used in the filtering condition *hc*. Note, in the star schema model, there are three different types of measures, additive, semi-additive and non-additive, of which the semi-additive measures are defined along some dimensions. For this reason, the checking of *hc* should be considered in accordance with the grouping ID to avoid invalid aggregation along the wrong dimensions.

Example 10: t_G = {*ProdName*, *Date*}, *hc* = (*sum*(*SaleAmount*) > 1000)

In the schema ontology *'SaleAmount'* is an additive measure, therefore the system will pass the checking.

Example 11: t_G = {*CustID, Date*}, *hc* = (*sum*(*Cost*) < 100)

The system will reject the setting because *'Cost'* is a semi-additive fact and should be, as shown in Figure 6, aggregated along with dimensions including *Product*.

Intelligent assistance in element recommendation

As well as the semantic checking, the system offers recommendations to lead the user, especially the inexperienced one, toward a more efficient mining model formulation process. The functions, taking partial input from the unfinished mining model element created by the user, spontaneously list the recommendations of possible successive mining model constituent drawn from user preference ontology for users to refer to. Based on the example of the user preference ontology in Figure 10, we present some examples as follows:

(a) Recommendation of t_G by giving a partial t_G

A partial t_G is taken as the key to search in the user preference ontology. It can be empty once the user does not know how to start a model setting. The system will respond with the available t_G list.

Example 11: Given partial t_G = {*CustID*}

The system will prompt with the list {*Date*},..., {*Category*} as succeeding t_G candidates for the user to refer to.

(b) Recommendation of t_M by giving t_G and *partial* t_M

With input of t_G and *partial* t_M, the system will search the available sets of mining attributes in the user preference ontology for recommendations.

Example 12: Given t_G = {*CustID, Date*} and partial t_M = {*ProdName*}

The system will prompt the user with the following list: {*Education*},...,{*Salesman*} as referable suggestions.

(c) Recommendation of *ms* and *mc* by giving t_G and t_M

Example 13: t_G = {*CustID, Date*}, t_M = {*ProdName, Education*}

The corresponding *ms* and *mc* of the given t_G and t_M in the user preference ontology will be listed. In this case, *ms* = 60% and *mc* = 85% will be suggested.

5. A system prototype

A prototype implementation of the system is delivered to demonstrate the feasibility of our study. We use Borland C++ Builder to develop this system and the data warehouse is stored in the SQL Server 2005. A data mining job involves the following steps:

(1) The user finishes a mining model setting through the intelligent user interfaces.

(2) Data is selected from data warehouse, data mart or data cube according to the mining model setting and is transformed to target data in the format shown in Table 1.

(3) Target data is fed into mining engine for frequent itemsets and rule generation.

For step 1, we designed a step by step wizard to guide the users toward correct mining model setting, especially for those who are not familiar with data mining. Figure 13 to Figure 17 show some examples of the user interfaces. The dimension tree is a treeview box showing the dimensions and the attributes of the selected data warehouse. The user specifies the transaction ID or pertinent mining attributes by drag-and-drop. If the settings have ever been improper or incorrect, the system will prompt the user with warnings or error messages as shown in Figure 13 to remind the users. The system also provides recommendations of transaction ID or mining attributes, as shown in Figure 14 and Figure 15. After the steps of transaction ID and mining attribute settings, optional filtering conditions before transaction grouping can be set as shown in Figure 18, if necessary. In Figure 16, the system provides drag-and-drop of dimension attributes to the attribute edit box. List boxes for the relational operator and the attribute value settings are also provided for users to perform data selection. This decreases the user's manual typo as much as possible. The system also shows the filtering instances from the user preference ontology, serving to support users who have little sense of setting filtering conditions. The optional filtering condition after transaction grouping is also provided in a similar way. The last step is to set the pertinence threshold, as shown in Figure 17. Users set the minimum support and minimum confidence with the suggestion values provided by the system.

The target data selection and grouping is manipulated by customized functions we have developed. In step 3, for the processing of frequent itemset generation and rule derivation, the domain ontology is utilized to derive extended concepts of rules. We use the AROC algorithm in (Tseng et al., 2007) to generate frequent itemsets and rules. It integrates the classification and composition relationships in the domain ontology to extend the implications of the rules.

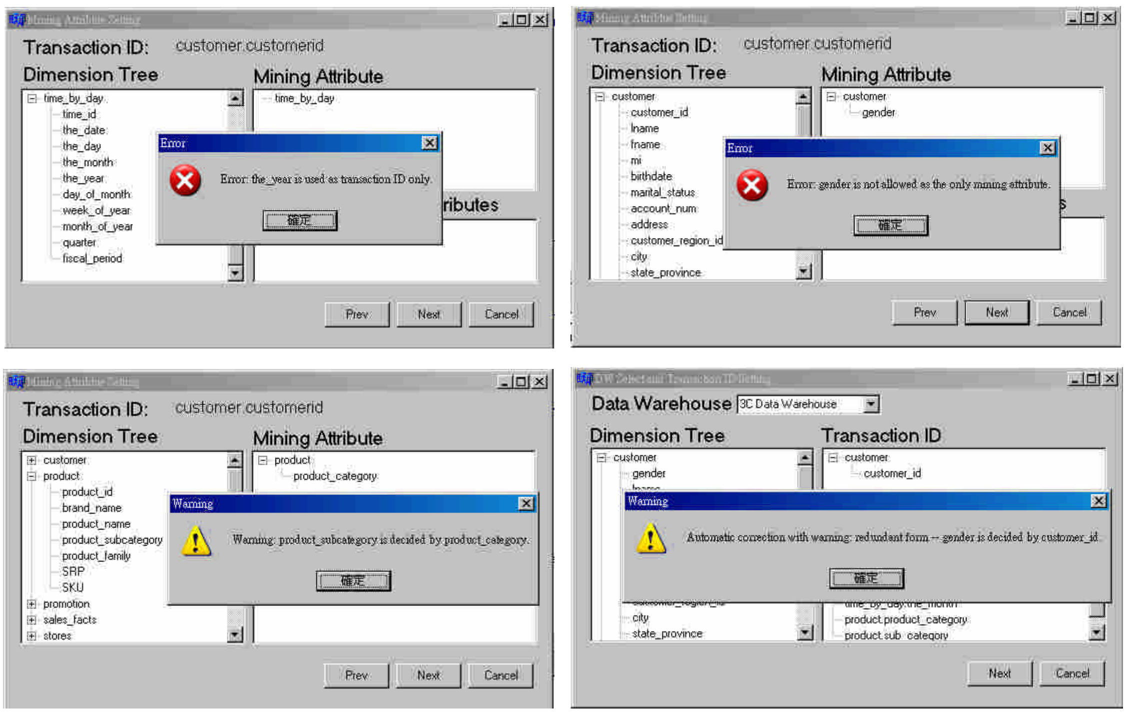

Figure 13. Some examples of mining model settings with improper actions

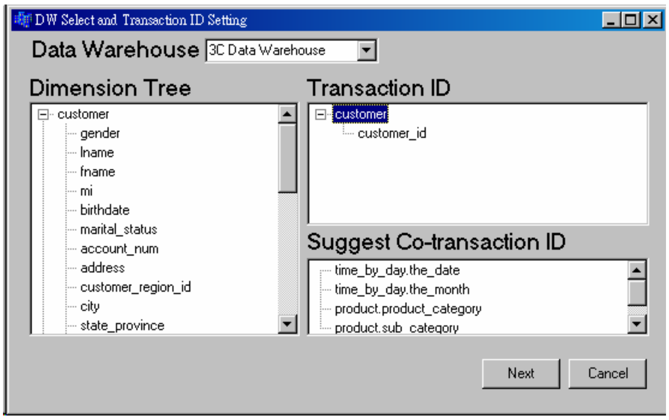

Figure 14. An example of recommendations for transaction ID setting

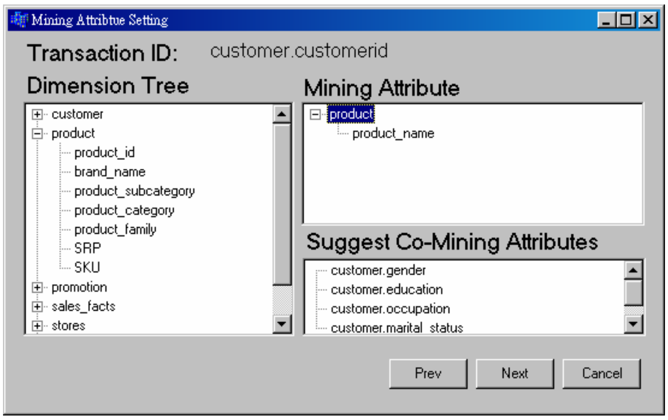

Figure 15. An example of recommendations for mining attribute setting

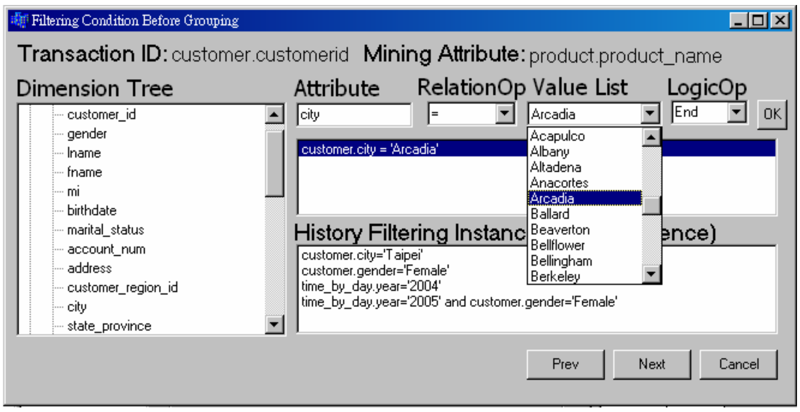

Figure 16. An example of filtering condition settings before transaction grouping

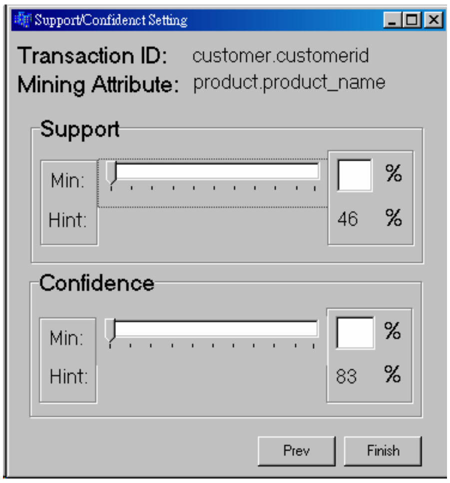

Figure 17. An example of a pertinence threshold setting

6. Related work

This section gives an overview of the literature related to our work in twofold: the use of ontology in data mining and data warehouse mining.

(1) Use of ontology in data mining

If the concept hierarchy or taxonomy can be viewed as an ontology, then the use of ontologies in data mining can be traced back to 1991 when Nunez used information of the classification hierarchy and attribute processing cost to improve the efficiency of the classification process (Nunez, 1991). Later, Han & Fu (1995) and Srikant & Agrawal (1995) also proposed combining classification hierarchies to mine multilevel association rules and generalized association rules, respectively. Their works were later extended by Chien et al. (2007), who not only applied classification but also composition hierarchical knowledge to mining fuzzy association rules. These researches, however, concentrated on the design of the algorithms, yet discussion of ontology structure design and its benefit to data mining were not covered. Until recently, research on applying ontology to data mining was exploited by several studies such as, ontology-based induction of rules (Aronis et al., 1996; Taylor et al., 1997), ontology-based business understanding (Sharma & Osei-Bryson, 2009), ontology-based post-processing and explanation of association rules (Domingues & Rezende, 2005; Liao et al., 2009; Marinica et al., 2008; Svatek et al., 2005), ontology-supported selection of classification algorithms (Bernstein et al., 2005; Lin et al., 2006), ontology-guided new attributes generation from databases (Phillips & Buchanan, 2001), and ontology-based integration and preprocessing of data (Euler & Scholz, 2004; Perez-Rey et al., 2006).

Differing from the above work on dealing with the issue of incorporating ontology in the individual phase of the well known KDD process proposed by Fayyad et al. (1996), there has been work conducted from an integral perspective. For example, Kopanas et al. (2002) pointed out the essence of incorporating ontology (the term *domain knowledge* is used instead) to the KDD process and demonstrated their viewpoints using a telecommunication customer insolvency case study. Cespivova et al. (2004) conducted a systematic study by discussing the roles of medical domain ontology in each aspect of the KDD process. A similar study was also presented in (Gottgtroy et al., 2004; Kuo et al., 2007). A position paper presented by Charlest et al. (2006) discussed the synergy of combining case based reasoning and ontology in the context of data mining assistance framework, though the issue of realization and implementation was left aside. In 2006, Pan & Pan proposed an ontology supporting data mining from databases. They maintained previous mining results in ontology that can further be applied for incremental association rule mining.

(2) Data warehouse mining

Currently, the research on data warehouse mining is mostly concentrated on data mining from data cubes or multi-dimensional databases. J. Han's research group pioneered this research subject (Han, 1998; Han et al., 1999). The study conducted by Ester and his colleagues (Ester et al., 1998; Ester & Wittmann, 1998) instead considered the problem of incrementally updating mined patterns from data warehouses. In 2000, Psaila and Lanzi studied multi-level association mining from a primitive data warehouse and proposed a mining algorithm. Since then, substantial works have been devoted to discovering multidimensional association rules from data warehouses (Ng et al., 2002; Chung & Mangamuri, 2005; Tjioe & Taniar, 2005; Messaoud et al., 2006; Yang et al., 2008).

The research by Priebe & Pernul (2003) first exploited the issues of incorporating ontology into knowledge discovery from data warehouses. In particular, it proposed an intelligent web portal integrating OLAP and information retrieval through ontology, yet it focused on information retrieval issues but not on data mining. Subsequent work on multiple source integration for data warehouse OLAP construction includes Niemi et al. (2007) and Shah et al. (2009). In (Wu et al., 2007), we presented the problems with contemporary association rule mining in data warehousing systems, explained the essence that incorporates ontologies to resolve the problems, anddemonstrated a preliminary framework.

7. Conclusions

The purpose of data mining is for users to find real and useful knowledge they actually want. In this paper we have shown a data warehouse mining system framework with intelligent assistance incorporating schema ontology, schema constraint ontology, domain ontology and user preference ontology. We have demonstrated the intelligent assistance provided by the mining system in guiding users through the mining processes. This improves the mining effectiveness and efficiency in four aspects as follows. First, the processes of the mining model settings are assisted by intelligent functions, minimizing the possibilities of illegal settings of mining models. Also, appropriate recommendations of the mining model elements are provided while the users are setting the mining model. This avoids execution of ineffective or redundant mining processes and also guides the users through the approaching of the mining models that are closer to their mining intention. Second, with the support of domain ontology, mining rules can be extended and generalized. Third, the information in the domain ontology can be included in the filtering condition to obtain a more specific search space. More precise knowledge can be discovered. Fourth, it provides the system with knowledge browsing capability that a mining model can be examined against the user preference ontology for any duplication or similarities. This saves the system's resources. In this paper, we have discussed the intelligent assistance in general. A preliminary implementation of this system framework has also been provided to demonstrate the claimed benefits.

The ontologies we have proposed in this paper are implemented in relational table structures. Nevertheless, these ontologies are local to the specific mining system we have proposed. Making them globally sharable is challenging and is an important future work.

Acknowledgements

This work was supported by the National Science Council of R.O.C. under grant NSC 95-2221-E-390-024.

References

Aronis, J.M., Provost, F.J., & Buchanan, B.G. (1996). Exploiting background knowledge in automated discovery. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 355–358).

Bernstein, A., Provost, F., & Hill, S. (2005). Toward intelligent assistance for a data mining process: an ontology-based approach for cost-sensitive classification. *IEEE Transactions on Knowledge and Data Engineering* 17(4), 503–518.

Cespivova, H., Rauch, J., Svatek, V., Kejkula, M., & Tomeckova, M. (2004). Roles of medical ontology in association rule mining Crisp-Dm cycle. In *Proceedings of ECML/PKDD Workshop on Knowledge Discovery and Ontologies*.

Charest, M., Delisle, S., Cervantes, O., & Shen, Y. (2006). Intelligent data mining assistance via CBR and ontologies. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications* (pp. 593–597).

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehouse and OLAP technology. *ACM SIGMOD Record* 26(1), 65-74.

Chien, B.C., Zhong, M.H., & Wang, J.J. (2007). Mining fuzzy association rules on has-a and is-a hierarchical structures. *International Journal of Advanced Computational Intelligence and Intelligent Informatics* 11(4), 423–432.

Cheung, D.W., Han, J., Ng, V.T., & Wong, C.Y. (1996). Maintenance of discovered association rules in large databases: An incremental update technique. In *Proceedings of the 12th International Conference on Data Engineering* (pp. 106–114).

Chung, S.M., & Mangamuri, M. (2005). Mining association rules from the star schema on a parallel NCR teradata database system. In *Proceedings of International Conference on Information Technology: Coding and Computing* (pp. 206–212).

Domingues, M.A., & Rezende, S.O. (2005). Using taxonomies to facilitate the analysis of the association rules. In *Proceedings of the 2nd International Workshop on Knowledge Discovery and Ontologies*.

van Elst, L., & Abecker, A. (2002). Ontologies for information management: balancing formality, stability, and sharing scope. *Expert Systems with Applications* 23(4), 357–366.

Ester, M., Kriegel, H.P., Sander, J., Wimmer, M., & Xu, X. (1998). Incremental clustering for mining in a data warehousing environment. In *Proceedings of 24th International Conference on Very Large Data Bases* (pp. 323–333).

Ester, M., & Wittmann R. (1998). Incremental generalization for mining in a data warehousing environment. In *Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology* (pp. 135–149)

Euler, T. & M. Scholz. (2004). Using ontologies in a KDD workbench. *In Proceedings of ECML/PKDD Workshop on Knowledge Discovery and Ontologies.*

Fayyad U., Piatetsky-Shapiro G., and Smyth P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM* 39(11), 27–34.

Gottgtroy, P., Kasabov, N., & MacDonell, S. (2004). An ontology driven approach for knowledge discovery in biomedicine. In *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*.

Han, J. (1998). Toward on-line analytical mining in large databases. *ACM SIGMOD Record* 27(1), 97–107.

Han, J., Chiang, J.Y., Chee, S., et al. (1997). DBMiner: A system for data mining in relational databases and data warehouses. In *Proceedings of the 1997 Conference of the Centre for Advanced Studies on Collaborative Research* (pp. 250–255).

Han, J., & Fu, Y. (1995). Discovery of multiple-level association rules from large databases. In *Proceedings of the 21st Very Large Databases Conference* (pp. 420–431).

Han J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*, Morgan Kaufmann.

Han, J., Lakshmanan, L.V.S., & Ng, R.T. (1999). Constraint-based, multi-dimensional data mining. *IEEE Computer* 32(8), 46–50.

Heflin, J., Editor (2004). *OWL Web Ontology Language Use Cases and Requirements*, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-webont-req-20040210/

Inmon, W.H. (1995). *Building the Data Warehouse*, John Wiley & Sons, Inc., New York, NY.

Kimball, R. (1996). *The Data Warehouse Toolkit Practical For Building Dimensional Data Warehouses*, John Wiley & Sons, Inc.

Kopanas, I., Nikolaos N., Avouris, M., & Daskalaki, S. (2002). The role of domain knowledge in a large scale data mining project. In *Proceedings of the 2nd Hellenic Conference on AI: Methods and Applications of Artificial Intelligence* (pp. 288–299).

Kuo, Y.T., Lonie, A., & Sonenberg, L. (2007). Domain ontology driven data mining: A medical case study. In *Proceedings of ACM SIGKDD Workshop on Domain Driven Data Mining* (pp. 11–17).

Liao, S.H., Ho, H.H., & Yang, F.C. (2009). Ontology-based data mining approach implemented on exploring product and brand spectrum. *Expert Systems with Applications* 36(9), 11730–11744.

Lin, M.S., Zhang, H., & Yu, Z.G. (2006). An ontology for supporting data mining process. In *Proceedings of IMACS Multiconference on Computational Engineering in Systems Applications* (pp. 2074–2077).

Lisi, F.A., & Malerba, D. (2004). Inducing multi-level association rules from multiple relations. *Machine Learning* 55(2), 175–210.

Liu, J., & Yin, J. (2001). Towards efficient data re-mining (DRM). In *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, *Lecture Notes in Computer Science* 2035 (pp. 406–412).

Marinica, C., Guillet, F., & Briand, H. (2008). Post-processing of discovered association rules using ontologies. In *Proceedings of IEEE International Conference on Data Mining Workshops* (pp. 126–133).

Messaoud, R.B., Rabaséda, S.L., Boussaid, O., & Missaoui, R. (2006). Enhanced mining of association rules from data cubes. In *Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP* (pp. 11–18).

Ng, E.K.K., Ng, K., Fu, A.W.C., & Wang, K. (2002). Mining association rules from stars. *In Proceedings of the 2002 IEEE International Conference on Data Mining* (pp. 322–329)

Niemi, T. Toivonen, S., Niinimaki, M., & Nummenmaa, J. (2007). Ontologies with semantic web/grid in data integration for OLAP. *International Journal on Semantic Web and Information Systems* 3(4), 25–49.

Nunez, M. (1991). The use of background knowledge in decision tree induction. *Machine Learning* 6(3), 231–250.

Pan, D., & Pan, Y. (2006). Using ontology repository to support data mining. In *Proceedings of the 6th World Congress on Intelligent Control and Automation* (pp. 5947–5951).

Perez-Rey, D., Anguita, A., & Crespo J. (2006). OntoDataClean: Ontology-based integration and preprocessing of distributed data. *Lecture Notes in Computer Science* 4345, 262–272.

Perng, C.S., Wang, H., Ma, S., & Hellerstein, J.L. (2001). Farm: A framework for exploring mining spaces with multiple attributes. In *Proceedings of the 1st IEEE International Conference on Data Mining* (pp. 449–456).

Perng, C.S., Wang, H., Ma, S., & Hellerstein, J.L. (2002). User-directed exploration of mining space with multiple attributes. In *Proceedings of the 2nd IEEE International Conference on Data Mining* (pp. 394–401).

Phillips, J., & Buchanan, B.G. (2001). Ontology-guided knowledge discovery in databases. In *Proceedings of the 1st International Conference on Knowledge Capture* (pp. 1230–130).

Priebe, T., & Pernul, G. (2003). Ontology-based integration of OLAP and information retrieval. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications* (pp. 610–614).

Psaila, G., & Lanzi, P.L. (2000). Hierarchy-based mining of association rules in data warehouses. In *Proceedings of ACM Symposium on Applied Computing* (pp. 307–312).

Sharma, S., & Osei-Bryson, K.M. (2009). Framework for formal implementation of the business understanding phase of data mining projects. *Expert Systems with Applications* 36(2), 4114–4124.

Shah, N., Tsai, C.F., Marinov, M., Cooper, J., Vitliemov, P., & Chao, K.M. (2009). Ontological on-line analytical processing for integrating energy sensor data, *IETE Technical Review* 26(5), 375–387.

Srikant, R., & Agrawal, R. (1995). Mining generalized association rules. In *Proceedings of the 21st Very Large Data Bases Conference* (pp. 407–419).

Svatek, V., Rauch, J., & Flek, M. (2005). Ontology-based explanation of discovered associations in the domain of social reality. In *Proceedings of the 2nd International Workshop on Knowledge Discovery and Ontologies*, 2005.

Taylor, M., Stoffel, K., & Hendler, J. (1997). Ontology-based induction of high level classification rules. In *Proceedings of SIGMOD Data Mining and Knowledge Discovery Workshop*.

Tjioe, H.C., & Taniar, D. (2005). Mining Association Rules in Data Warehouses. *International Journal of Data Warehousing and Mining* 1(3), 28–62.

Tseng, M.C., Lin, W.Y., & Jeng, R. (2007). Mining association rules with ontological information. In *Proceedings of the 2nd International Conference on Innovative Computing, Information and Control* (pp. 300–303).

Uschold, M., & Gruninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review* 11(2), 93–155.

Wu, C.A., Lin, W.Y., Tseng, M.C., & Wu, C.C. (2007). Ontology-incorporated mining of association rules in data warehouse. *Journal of Internet Technology* 8(4), 477–485.

Wu, C.A., Lin, W.Y., Jiang, C.L., & Wu, C.C. (2009). Favorable support threshold recommendation for multidimensional association mining using user preference ontology. In *Proceedings of 2009 IEEE International Conference on Granular Computing* (pp. 586–591).

Yang, W., Li, Y., Wu, J., & Xu, Y. (2008). Granule mining oriented data warehousing model for representations of multidimensional association rules. *International Journal of Intelligent Information and Database Systems* 2(1), 125–145

Zhu, H. (1998) *On-Line Analytical Mining of Association Rules*. Master's Thesis, Simon Fraser University, U.S.A.