# CIS: CHANGE IDENTIFICATION SYSTEM

Parimala N. and Vinay Gautam
*School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi, India*

Keywords: Change Identification, Data Warehouse, Multidimensional Schema, Goal-Information, Ontology.

Abstract: Change identification is one of the main challenges for Data Warehouse Schema evolution. Changes to the schema are required, among other situations, when the data warehouse fails to provide information to the decision maker. In this paper we address the issue of identification of changes when such a situation occurs. Towards this, the decision maker is asked to specify the information he/she needs, in business terms, to meet a goal. With the help of ontology and a set of rules we identify whether the information is present in the warehouse or not. The absence of data could be because it is not directly stored or because it is actually absent. In both these cases the changes needed to the data warehouse schema are suggested by the system, called the Change Identification System (CIS).



## 1 INTRODUCTION

In this paper we are concerned with the identification of *potential* changes to a data warehouse system. Zohra Bellahsene (2005) has addressed modification of the data warehouse schema which is not necessarily dictated by the modifications to the underlying data sources. The fact that such a modification is required implies that there is a gap between the information content of the warehouse and the information that is needed by the decision maker. This gap defines the change to be incorporated in the warehouse schema. We are concerned with identifying these gaps.

As an example, consider an Insurance Schema represented as a star schema as given in Figure 1. Let's say that the goal of the decision maker is to "Increase the revenue". In order to achieve this goal, the decision maker may have to analyze "the revenue within individual regions" and "the claim of policies within individual cities".

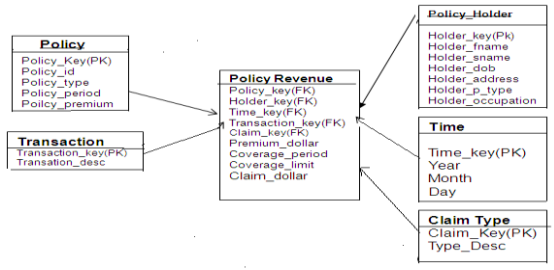


Figure 1: Multidimensional Schema for Insurance.

Consider the case where it is needed to get "the revenue for each city". If we look at the schema there is clearly no dimensional attribute city and revenue is defined as premium_dollar.
Two things can be observed. One, the warehouse is not providing adequate information to the decision maker. Second, the decision maker is expressing the information using business terms which are not the same as schema terms. In order to address the latter, we maintain an ontology of business as well as schema terms and a mapping between these. In order to address the former, we identify the missing information. Information could be missing because it is altogether absent or that it is not directly represented. In the above example, the attribute city is not directly represented but is part of the attribute address. Different cases arise when information is not directly represented. These cases give rise to different kinds of modifications. These have to be identified. To accomplish this we have built the Change Identification System (CIS) shown in Figure 2.

### 1.1 Related Work

Conventional approaches for the management of changes to a multidimensional schema and the contents (which is the data warehouse) can be broadly classified into two categories namely, Schema evolution and Version extension. The former is addressed by Body et al. (2003) and

N. P. and Gautam V..
CIS: CHANGE IDENTIFICATION SYSTEM.
DOI: 10.5220/0003069003470350
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2010), pages 347-350
ISBN: 978-989-8425-29-4

Bartosz Bebel et al. (2006) where the changes are made to the multidimensional structure without retaining the existing definition.

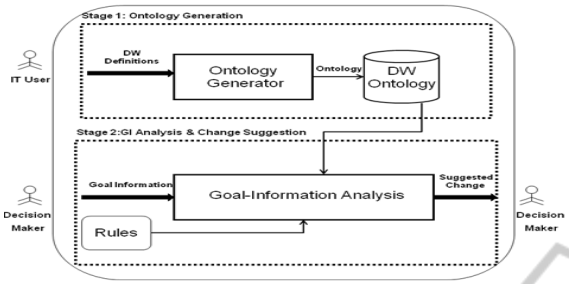


Figure 2: CIS Architecture.

In the version extension approach as given by Shazad, M.K, et al. (2005) and Mathurine Body et al. (2002) a new version is derived from the previously existing one and all versions are maintained. In all these techniques the changes in underlying data sources have driven the identification of data warehouse change. On the other hand, we are looking at the definition of the warehouse per se and analyzing its adequacies to meet the needs.

Ontology has been widely addressed in literature. Ontology specification is formally described by W.L.Lacy (2005). Web Ontology Language (OWL 2008) which has been standardized by W3C has been adopted by many. In particular, Ahlem nabli et al. (2009) uses ontology to resolve semantic and structural ambiguities across heterogeneous sources. Guotong Xie (2008) has used the ontology to build automatically a customized data mart. Ontology has been used to update hierarchies by Fadila Bentayeb et al. (2008). In our approach, the changes to be made are determined by the system using the ontology.

Goals are studied as an objective of the system. Paolo Giorgini et al. (2008) suggest a data warehouse schema based on TROPOS methodology and Naveen Prakash & Anjana Gosain. (2008) study the relationship between goal, decision and information. Goals are used by the user to determine the required information.

The layout of the paper is as follows. Section 2 deals with building the ontology. The Goal-Information template and the rules are explained in section 3. Section 4 is the concluding section.

# 2 BUILDING THE ONTOLOGY

As mentioned above we propose a generalized structure of ontology for a data warehouse. The ontology contains schema terms, domain terms which are synonyms of schema terms, related terms, terms which represent time and the relations between these.

The data warehouse concepts which are Fact, Dimension, Attribute, Level, and Hierarchy are defined as classes in the Ontology. Three new classes Time/DateExtension, Term and Synset are introduced. **Time/DateExtension** class is defined immaterial of whether a time or a date dimension is defined in the multidimensional schema. This, we believe will facilitate providing more meaningful information to the user. **Term** Class contains related terms to dimensional attribute. **Synset** class contains the domain terms which have the same sense as the terms in Fact, Dimension and Attribute class. All the classes are derived from the root class Thing.

The relation between the classes in the ontology is represented as OWL Property. Fact_Attribute and Dim_Attribute are represented as object properties which are relations between Fact & Attribute and Dimension & Attribute respectively.

We introduce additional properties between classes.

- Fact_Synset: Fact_Synset is a relation between the classes Synset and Fact..
- Dim_Synset: Dim_Synset is a relation (which expresses synonyms) between the Classes Synset and Dimension.
- Relation between the classes Sysnet and Attributes..
- PartOf: PartOf is a relation between the Attribute Class and Term Class.
- *Derived_Attribute*: Derived attributes are those that can be derived from attributes defined in multidimensional schema. Derived_Attribute is introduced as functional property.

# 3 CHANGE IDENTIFICATION

As explained above, our system helps to identify the changes needed in the schema. The first step is to express the information needed by the decision maker to satisfy a goal as described in section 3.1. Next, the manner in which changes are identified is explained in section 3.2.

### 3.1 Goal-Information Analysis

The information that is required is referred to as the Analysis Component. A goal can have more than one analysis component associated with it. The template consists of the Goal and the Analysis Component (AC). The Analysis Component consists of two parts which are 'What is to be analyzed' and 'How is it to be analyzed''. "What is to be analyzed" describes the data to be analyzed. It refers to the measures which represents the factual data. 'How is it to be analyzed'' describes the business perspective under which data analysis is to be performed. "How or along which dimensional attributes is it to be analyzed". These two represent the context of user information needs. For the sake of brevity, 'What is to be analyzed' will be referred to as 'What' and 'How is it to be analyzed' will be referred to as 'How'.

Consider an example with respect to Figure 1 where the goal is to increase the number of policy holders. Here, Policy is to be analyzed and forms the 'What' component. The axis of analysis could be age, quarter and address. These form the 'How' component.

### 3.2 Identifying Changes

In this step, we search the ontology to know whether the information is present or not. If the information is not present then potential changes are suggested. We have formulated rules to perform these actions. We refer to the terms 'What' and 'How' of the analysis component of section 3.1 as Item.

In the rules given below, IsEqual( ) is a Boolean function which compares two terms and returns true if they refer to the same term else it returns false. The notation classname.Instancename refers to an instance of a class. This could be a dimensional attribute or a fact attribute. getName(Instance) is a method which gives the name of the given 'Instance'. If any rule fires, then the remaining rules are skipped. If no rule fires then rule R0 given below is executed.

```
R0: Message = Item + 'is not
available'+ Change_Suggestion (add
(Item))
```

Rules :

```
R1: If   IsEqual (What, Attribute
[Fact_Attribute].Instance)then Message
= What + 'is available as
Fact_Attibute'.
R2: If  IsEqual (How, Attribute
[Dim_Attribute].Instance)then Message =
How +'is available as Dimensional
Attribute'.
R3: If IsEqual
(What,Dimension.Instance) then  Message
= What + 'is available as Dimension. It
can't be used as a measure of
analysis'.
R4: If IsEqual (What, Fact.Instance)
then Message = What + 'is available as
Fact; Use its attributes for analysis'
R5: If IsEqual(How,Dimension.Instance)
then Message=How+'is available as
Dimension; Use its attributes for
analysis'
R6: If IsEqual(How,Fact.Instance)then
Message=How+'is available as fact;
can't be used  as a dimension of
analysis'.
R7:If IsEqual(Item,Synset.Instance)then
Message= Item+'is available as'+
getName
Synset.Instance[Relation].Class.Instanc
e
R8: If IsEqual (Item,
Derived_Attribute. Instance)then
Message = Item + 'can be derived from
the attribute'+getName(
Derived.Instance.Attribute.Instance).
R9:If IsEqual (How, Term.Instance) then
Message = How + 'is a part of
'+getName(
Term.Instance.PartOf.Attribute.Instance
)
R10:If IsEqual(How,Time/DateExtension.
Instance) then Message = How + 'is not
available but it can be added as a
level of Time dimension'
```

It may be noted that 'How' has two additional rules which are not present for 'What' rules. Time/Date is typically a dimensional attribute and hence is not considered while designing the rules for measures. Similarly, we believe that a constituent part of a measure is not analyzed but constituents of a composite dimensional attribute may be used as an axis of analysis. Therefore, there is no rule corresponding to PartOf relationship for the item appearing as 'What'.

## 4 CONCLUSIONS

In this paper we have proposed the Change Identification System which helps in finding out whether the information required by the decision maker is available in the data warehouse or not. For doing so, in the first stage we build an Ontology.

In the next stage the decision maker is asked to identify the information he needs from the data warehouse. This is expressed using a goal information template using business terms.

In our system, the rules process the information in the template and identify whether the information is directly present or not. When the data is not directly available in the warehouse, two situations are handled by the system. If the data exists and is not directly derivable then this information is provided. Secondly, if the required information is altogether missing, then changes to the multidimensional schema are suggested.

We have defined a hierarchy for the time dimension by which it is possible to suggest addition of levels in the Time dimension. We propose to extend this to other Dimensions by defining 'belongs to' relationship in the Ontology. We propose to add aggregate functions to the Ontology so that when these are specified by the decision maker, a better analysis of his requirements can be made.

It may be argued that there is no real need to build ontology but business metadata that explains the business context or existing ontology like WordNet can be used. However, ontology not only contains business terms and their taxonomy but also includes relationships between attributes, synonyms, word variants etc. (Guotong Xie, 2008; Matthias Kehlenbeck, 2009). In this sense, an ontology consists of rich domain information. Even though industry standard ontology exists for common data domains, for the not so common, the ontology has to be built anyway.

We are currently building a graphical tool using which the ontology can be defined.

The Change Identification System is implemented using Java, Jena and OWL-API.

## REFERENCES


Lacy W. L.(2005), Representing Information Using the Web Ontology Language. Trafford Publishing.

Zohra Bellahsene (2002), Schema evolution in data warehouses. Knowledge and Information System Journal, Springer-Verlag Newyork, Vol. 4 issue 3, pages 283-304.

Ahlem nabli, Jamel Feki and Farez Gargouri (2009), An ontology based method for normalization terminology. LNCS, Springer, Vol. 4870/2009, pages 235-246.

OWL (2008) http://www.w3.org/TR/2008/WD-owl2-xml-serialization-20080411/

Matthias Kehlenbeck and Michael H. Breitner (2009), Ontology-Based Exchange and Immediate Application of Business Calculation Definitions for Online Analytical Processing. DaWaK , pages 298-311

Naveen Prakash and Anjana Gosain (2008), An approach to engineering requirements of data warehouse, Requirement Engineering Journal, Springer, Vol. 13, No. 1.

Bartosz Bebel, Zbyszko Królikowski, and Robert Wrembel (2006), Managing Evolution of Data Warehouse System by Mean of Nested Transaction, Springer Berlin / Heidelberg, Vol. 4243.

Fadila Bentayeb, Cecile Favre and Omar Boussaid (2008), A User-driven data warehouse Evolution Approach for Concurrent Personalized Analysis Needs, Published in Integrated Computer-Aided Engineering, Vol. 15 No. 1 , pages 21-36.

Paolo Giorgini, Stefano Rizzi and Maddalena Garzetti (2008), GRAnD: A `goal-oriented approach to requirement analysis in data `warehouses, Decision Support Systems, Vol. 45 , Issue 1,Pages 4-21.

Shazad, M. K., J. A. Nasir and M. A. Pasa: Creation and evolution versions in data warehouse, Asian journal of IT, 910-917, 2005, Grace Publication.

Body, M., Miquel, M., Bedard, Y. and Tchounikine, A. (2003), Handling Evolutions in Multidimensional Structures, Proceedings of 19th International Conference on Data Engineering, pages 581- 591

Mathurine Body, Marryvonne Miquel, Yvan Bedard and Anne Tchounikie (2002), A multidimensional and multi-version structure for OLAP Application, 5th ACM international workshop on Data Warehousing and OLAP, Pages: 1–6.

Guotong Xie, Yang Yang, Shengping Liu, Zhaoming Qiu, Yue Pan and Xiongzhi Zhou (2008), EIAW: Towards a Business-Friendly data warehouse using Semantic Web Technology, LNCS, Springer, Vol. 4825.