Designing of the CASE Tool to develop Star Schema using the XML Schema consisting of Clinical Data

Anita Nagar
Manav Rachna College of Engineering
Faridabad

Gunjan Chandwani
Manav Rachna College of Engineering
Faridabad

ABSTRACT

Data Warehouse is most widely used for data analysis which supports the management’s decision making process. The sources for the Data Warehouse are commonly taken from the online transactional systems in various formats. XML is one of the standard format used to represent and transport the data in web based systems. XML allows easy sharing of data between different internet applications which enhances the decision making in organizations. This paper focuses on conversion method of XML Schema into the Star and Snowflake schema.

Keywords


1. INTRODUCTION

Data Warehouse is a central data repository that is maintained separately from an organization’s operational databases. Data warehouse systems allow for integration of a variety of application systems. It stores current as well as historical data and are used for creating reports for senior management reporting such as annual and quarterly comparisons. It is a subject-oriented, integrated, time variant and non volatile collection of data in support of management’s decision making which make it different from other repository system such as relational database system, transaction processing system.

Basically multidimensional model is most popularly used to represent the data in Data Warehouse. The multidimensional model is represented commonly by using the Star schema and the Snowflake schema consist of one large central table (Fact table) and other smaller tables (Dimension tables). The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table. Dimension hierarchies are created by joining the dimensions in such a way that the parent dimension consist of foreign key with the same name as the primary key of child dimension. There is another method also available to represent the model which is a Fact constellation schema.

XML is well established standard for semi-structured data and also poses several benefits in web environment. As data could come from various heterogeneous sources so to make them compatible with the Data Warehouse schema and hence in order to integrate these data, they could be converted to XML. Relational Model on the other hand is the most standard and structured way of representing data model. Numbers of researches have been made over the time to map different data models to relational model. XML is also no exception. Proposed work in [1, 2, 3, 4] show different ways of transformation from XML to relational model schema. In this paper the focus is on converting XML schema to data warehouse schemas based on ROLAP. Thus we are not concentrating on the techniques of converting XML to relational model. Here we refer to some of the existing methods to transfer XML to data warehouse paradigm. XML data is associated with DTD [5] or XML schema [5]. XML provides Document Type Definition (DTD), which explains precisely what elements could appear as document and what the contents of the elements and attributes are. The approaches [7] [8] [9] show how XML data based on DTD have been converted to data warehouse schema. However DTD have some limitations. DTD do not have any built-in data types; also do not support user-derived data types and allow only limited control over cardinality. XML schemas are more powerful to represent XML document structure and overcome the limitations of XML DTD.

Data Warehouse allows integrated data to be processed analytically based on OLAP. So, here we focus on the conversion of XML Schema into the ROLAP due to the limitation of the DTD. [1, 2, 6, 11, 12] have been worked to integrate XML data in Data Warehouse. The paper [6] proposes a method to design multiple cubes of multidimensional model from XML schema. There has been work to convert the contents of XML Schema to the Star schema of Warehouse.

This paper focuses on the conversion of the XML Schema into the Star and Snowflake schema. The paper [12] proposes XML schema conversion to OLAP cube by identifying fact and dimension tables. The paper considers that there is only one root element which excludes the formation of Fact constellation schema as it do not take any connection among different fact tables.

This work is an extension of paper[1] in which a method is implemented in the form of a CASE Tool to convert the XML Schema into the Data Warehouse Schema. At First the Schema graph is being identified from the XML Schema which becomes the first step of the conversion process. In the next step, this Schema graph is used for the identification of the Fact constellation schema as it do not take any connection among different fact tables.

This work is an extension of paper[1] in which a method is implemented in the form of a CASE Tool to convert the XML Schema into the Data Warehouse Schema. At First the Schema graph is being identified from the XML Schema which becomes the first step of the conversion process. In the next step, this Schema graph is used for the identification of the Fact constellation schema as it do not take any connection among different fact tables. Then on the basis of the relationship and connections among the fact table and dimension tables, the kind of Schemas is being identified. This paper aimed to get a CASE Tool which enhances the ETL (Extraction, Transformation and Extraction) phase of Data Warehouse projects. The data could be extracted from the XML Schema according to the proposed methodology and then transformed to make the data compatible and loading to the Data Warehouse.
2. PRELIMINARIES

Schema graph: Schema graph is a way to represent the entities present in XML Schema. It consists of following properties:

a. It consists of different levels.
b. The entities are represented by the vertices.

Holder Element: These are the elements which have no predecessor. They are placed in Level-1 of the graph.

Contained Element: These are the elements which are directly connected to the Holder element. They are placed in Level-2 of the graph.

Secondary Element: These are the elements which are connected to the contained elements. They are placed in Level-3 of the graph.

If further elements have been encountered in the graph connected to the secondary element, they would be placed in Level-4 of the graph. Subsequently new level could be created whenever any elements would appear in the graph. In the Schema graph, all the entities are represented by the Rectangular shape and the attributes are represented through the oval shaped vertex.

3. METHOD TO CONVERT THE XML SCHEMA INTO THE WAREHOUSE SCHEMA

3.1 Overview of the method

There are two basic steps to be followed in order to model the Star Schema from the related XML Schema. In the first step, a Schema graph is formed from the related XML Schema and then in the next step, the fact table and the dimension tables and their relationships are identified. After the identification of relationship, the type of Warehouse Schema will be identified. The elements present in the Schema graph are HE, CE and SE. If any element found to be without any primary key then a new primary key would be added to it to make it unique. The HE would correspond to the fact table and for the entries of the fact table; the primary attribute of the CE’s would get placed in their corresponding to their HE. And furthermore if there would be any SE then it gets placed in the CE’s table.

3.1.1 Example of Star Schema

```xml
<xsd:element name="Treatment_desc" type="xsd:string" use="required"/>
<xsd:element name="Treatment_cost" type="xsd:decimal" use="required"/>
<xsd:element name="Equip_code" type="Equip_type" use="required"/>
</xsd:sequence>
<xsd:complexType name="Equip_type">
<xsd:sequence>
<xsd:element name="Equip_desc" type="xsd:string" use="required"/>
<xsd:element name="Equip_cost" type="xsd:string" use="required"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="Doctor_id" type="Doctor_type">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Doc_name" type="xsd:string" use="required"/>
<xsd:element name="Doc_spec" type="xsd:string" use="required"/>
<xsd:element name="Doc_dept" type="xsd:string" use="required"/>
<xsd:element name="Doc_fee" type="xsd:decimal" use="required"/>
<xsd:element name="Med_code" type="Med_type" use="required"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="Med_type">
<xsd:sequence>
<xsd:element name="Med_desc" type="xsd:string" use="required"/>
<xsd:element name="Med_price" type="xsd:decimal" use="required"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="Ward_no." type="Ward_type">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Assigned_date" type="xsd:date" use="required"/>
<xsd:element name="Discharged_date" type="xsd:date" use="required"/>
<xsd:element name="Bed per day_charge" type="xsd:decimal" use="required"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="Lab_id" type="Lab_type">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Lab_desc" type="xsd:string" use="required"/>
<xsd:element name="Test_id" type="Test_type" use="required"/>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="Test_type">
<xsd:sequence>
<xsd:element name="Test_desc" type="xsd:string" use="required"/>
</xsd:sequence>
</xsd:complexType>
```

Figure1: Schema Graph
3.2 Method to develop the Schema Graph from the XML Schema (Algorithm)

Find out those entities in XML schema that have no predecessor and denote them as the starting vertices or holders for the entire graph. These entities would be known as HE. They would placed in the Level-1 of the graph.

For all HE (i=1 to n) perform following : (n is the total number of HE)

Find the sequence of elements under i^th HE :

If it is an element then create a vertex for it into the graph and connect it with i^th HE. These elements (vertices) would be denoted as CE. CE would be placed in the level-2 of the graph.

Else if it is an attribute it would be considered as an attribute of the corresponding HE.

For all CE (j=1 to m) perform following: (m is the number of CE in the HE)

Scan the XML Schema for j^th CE:

If it is an element then place it into the graph and connect it with its CE.

These elements would be known as SE. SE would be placed in the level-3 of the graph.

Else if it is an attribute place it would be considered as an attribute of CE.

For every SE (k=1 to p) : (p is the total number of SE at that level)

Repeat the steps to include the entities and attributes as they encountered. Whenever a new entity is added new level is created for it.

End For /*SE */
End For /*CE */
End For /*HE*/

To build the Schema graph, Scan the XML Schema for the elements which are not being nested within any element, are now referred as HE and placed it at level-1 and the elements encountered as a nested element and are directly connected to the HE, are referred as CE’s. If any further element found to be nested within the CE, named them as SE and placed at level-3 or at some lower level as shown in figure below.

Figure 2: Example of Schema Graph

3.3 Identification of the Fact table and the dimension table

When the Schema graph is formed, the entities HE, CE and SE’s are identified. The HE is turned to become the Fact table with the name of HE + “fact” and the CE’s and SE’s would become the dimension tables with their name + “dimension”. In order to maintain the referential integrity, the primary key of the CE dimension would be placed in HE’s Fact table as a foreign key and similarly, the primary element of the SE would get placed in the CE’s Fact table. If in any element table is found to be without primary key, new primary key is then added with the element name + “id”.

3.4 Identification of the type of Warehouse Schema

There is a procedure to find the type of Warehouse Schema in which the checking of the Relationship between the HE as Fact table and the CE’s and SE’s as Dimension tables in the Schema graph. If there is no further SE’s are encountered under the CE’s then the Warehouse Schema is identified as Star Schema otherwise known as Snowflake Schema.

3.4.1 Procedure Star Schema

The Star Schema is identified, if the Warehouse Schema consists of only the HE and CE’s. The HE fact table consists of primary key of the dimension table as their foreign key. The Algorithm [1] is shown below:

Partition the Schema Graph Level wise.

Identify HE

For HE:

a) Form a Fact-Table with the name of HE + “Fact” and Primary key of the HE.
b) Specify the CE connected with this HE and include the primary keys of each CE into the Fact-Table.

3.4.2 Procedure Snowflake Schema

The Snowflake Schema is identified, if the Warehouse Schema consists of HE, CE and SE also. In addition to the Star Schema there is a primary key of SE which is placed in their corresponding CE above connected to it. If there is some further SE’s present in the Schema graph then the primary keys of them get placed in their immediate previous level SE. The Algorithm is shown:
3.4.3 Example
The Warehouse Schema builds from the above example (Figure 2)

Partition the Schema Graph Level wise.
Identify the HE:
For each HE:
Form a Fact-Table with the name of HE + "Fact" and primary key of the HE.
Specify the CE connected with this HE, include the primary keys of each CE into the Fact-Table.
For each CE find SEs, if any. If found Connect it with its CE using the primary key of the SE.
For each SE:
   Check if there is any SE:
      If further level of SE is found the primary key of the immediate higher level is placed in the SE of current level.
End For /*SE*/
End For /*CE*/
End For /*HE*/

3.5 EXPERIMENTAL RESULTS
After implementing the algorithm mentioned in section B, we get the result which is shown in figure 5 and before this we need to design an interfaces for browsing the XML Schema shown in figure 4 and prove the validation of XML Schema to show whether the syntax of XML Schema is appropriate or not and the checking of data to be saved in database is also done during this stage. Then a window pops up to provide the user a choice of selecting the button labeled as Schema graph or Star Schema. On pressing that button labeled as Schema graph, a Schema graph of given XML Schema is shown on the screen. Consequently, on selecting the option of Star Schema, a Star Schema is built against the Schema graph, drawn in figure 5, by using the algorithm mentioned in section D. The Star Schema is shown in figure 6.

4. CONCLUSION
This paper focuses on the CASE Tool, identifies the Star Schema and the Snowflake Schema from the related XML Schema which consists of the structure of the XML document consists of clinical data in above figure. More Often the XML is chosen as the data source to be transported over the internet between various Web applications and here specifically facilitated the transferring of data in XML format from various heterogeneous data sources to the Warehouse Schema. Data present in these heterogeneous data sources in the form of XML Schema and after going through the ETL phase. The Warehouse kept the contents of XML Schema in the form of their Schema. In order to form the Warehouse Schema, the CASE Tool in this paper will be used to convert the XML Schema into the appropriate Warehouse Schema. This approach can be helpful in the business intelligence in organizations to build the Warehouse Schema in short duration as the proposed methodology can be applied on to the changes in the XML sources whenever needed and quickly build the Warehouse Schema corresponding to that change.
5. REFERENCES

[1] Soumya Sen, Ranak Ghosh, Debanjali Paul, Nabendu Chaki; "Integrating Related XML Data into Multiple Data Warehouse Schemas"; Proc. of the First International Conference on Information


[10] Data Mining Concepts and Technique,2nd Edition, Jiawei Han and Micheline Kamber, Morgan Kaufmann Publisher.


