Implementation of XML Schema to Star Schema

Gunjan Chandwani
Department of Computer Science & Engineering
Manav Rachna College of Engineering, Faridabad, Haryana, India

Manvi Breja
Department of Computer Science & Engineering
Manav Rachna College of Engineering, Faridabad, Haryana, India

Abstract- Data Warehouse is most widely used by the organization for strategic analysis and decision making. The data for the Data Warehouse comes from various online transactional systems in various formats. The data warehouse is represented in the form of a star schema. The star schema is the simplest form of a dimensional model, in which data is organized into *facts* and *dimensions*. A fact is an event that is counted or measured, such as a sale or login. A dimension contains reference information about the fact, such as date, product, or customer. A star schema is diagramed by surrounding each fact with its associated dimensions. 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 implementation of XML Schema into the Star schema.

Keywords – Data Warehouse, Schema, Schema graph, Elements, XML Schema, Fact table and dimension table.

I. INTRODUCTION

A data warehouse is a relational database that is designed for query and analysis rather than for transaction processing. They store current and historical data and are used for creating trending reports for senior management reporting such as annual and quarterly comparisons.

W.H Inmonn defines Data Warehouse as a subject-oriented, integrated, time variant and non volatile collection of data in support of management's decision making.

- **Subject-Oriented**: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.
- **Integrated**: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.
- **Time-Variant**: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a ransactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.
- **Non-volatile:** Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

The multidimensional data model is an integral part of On-Line Analytical Processing, or OLAP.[18] It is designed to solve complex queries in real time. It is most popularly used to represent the data in Data warehouse and Data marts. Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse is represented commonly by using the Star schema, Snowflake schema and fact constellation schema.

- **Star Schema:**

The star schema is the simplest form of dimensional model, with data organized into facts and dimension. It is called a star schema because the diagram resembles a star, with points radiating from a center.

- **Snowflake Schema:**

A snowflake schema is a logical arrangement of tables in a multidimensional database such that the entity relationship diagram resembles a snowflake shape. The snowflake schema is represented by centralized fact tables which are connected to multiple dimensions

- **Fact Constellation Schema:**

Fact constellation schema is represented by splitting the original star schema into more star schemes each of them describes facts on another level of dimension hierarchies. The main shortcoming of the fact constellation schema is a more complicated design because many variants for particular kinds of aggregation must be considered and selected.

The rest of the paper is organized as follows. Star Schemas are explained in section II. Introduction to the XML is presented in section III. The basic concepts of Proposed System are explained in in section IV.Preliminaries are given in Section V.The Proposed Algorithm is explained in Section VI. Conclusion is presented in Section VII.

II. STAR SCHEMA

A. Introduction to Star Schema–

A star schema resembles a star, with points radiating from a center. The center of the star consists of fact table and the points of the star are the dimension tables. Usually the fact tables in a star schema are in third normal form(3NF) whereas dimensional tables are de-normalized.

- ***Fact Table :*** Fact table typically has two types of columns: foreign keys to dimension tables and measures those that contain numeric facts. Typical fact tables store data about sales while dimension tables data about geographic region(markets, cities) , clients, products, times, channels.
- ***Dimension Table:*** A dimension is a structure usually composed of one or more hierarchies that categorizes data. If a dimension hasn't got a hierarchies and levels it is called flat dimension or list. The primary keys of each of the dimension tables are part of the composite primary key of the fact table. Dimensional attributes help to describe the dimensional value. They are normally descriptive, textual values. Dimension tables are generally small in size then fact table.

The diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.

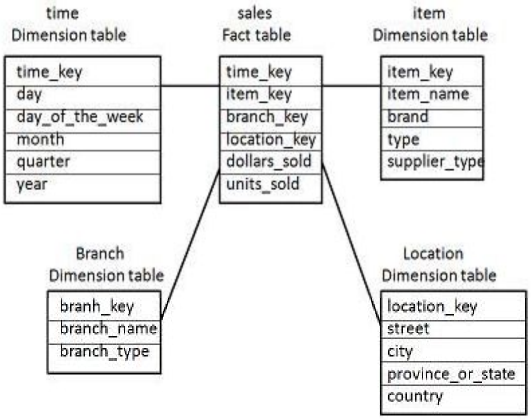


Figure 1: Sample star Schema

The Figure 1 shows a sample star schema,which contains a fact table at the center. It contains the keys to each of four dimensions.The fact table also contains the attributes, namely dollars sold and units sold.Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state, country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.

III. INTRODUCTION TO XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures such as those used in web services.

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. Large amounts of data, both financial and business data, and even data obtained from satellites, thatis, most of the data in today’s web driven world are being continuously converted to XML.

Numbers of researches have been made over the time to map different data models to relational model. XML is also no exception. Works in [1, 2, 3, 4] show different ways of transformation from XML to relational model schema.

A: XML Document Type Definition:-

XML data is associated with DTD [5] or XML schema [5]. A Document Type Definition (DTD) defines the legal building blocks of an XML document. It defines the document structure with a list of legal elements and attributes. A DTD can be declared inline inside an XML document, or as an external reference. 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.

B: XML Schema:-

An XML Schema is a language for expressing constraints about XML documents. A Schema can be used:

- to provide a list of elements and attributes in a vocabulary;
- to associate types, such as integer, string, etc
- to constrain where elements and attributes can appear, and what can appear inside those elements, such as saying that a chapter title occurs inside a chapter, and that a chapter must consist of a chapter title followed by one or more paragraphs of text;
- to provide documentation that is both human-readable and machine-processable;
- to give a formal description of one or more documents.

IV.PROPOSED SYSTEM

This paper focuses on the conversion of the XML Schema into the Star schema. OLAP Cube is a shortcut for multidimensional dataset, given that data can have an arbitrary number of dimensions. OLAP data is typically stored in a star schema or snowflake schema in a relational data warehouse or in a special-purpose data management system. The paper [12] proposes the conversion of XML schema to OLAP cube by identifying fact and dimension tables. The paper considers that there is only one root element and since it do not take any connection among different fact tables, therefore it excludes the formation of Fact constellation schema.

This work is an extension of paper[1] in which a method is proposed 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 table and the dimension 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 enhance 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.

V. PRELIMINARIES

A: 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.
c. schema graph specifies what edges are permitted in a data graph

B: Holder Element:

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

C: Contained Element:

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

D: 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.

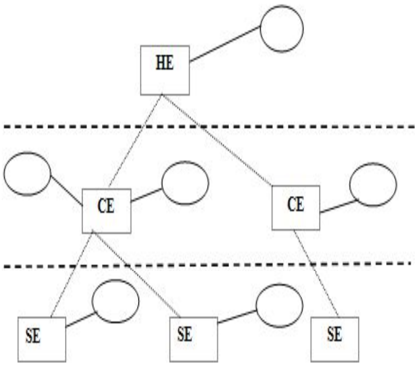


Figure-2: Sample Schema Graph

VI. ALGORITHM OF THE PROPOSED SYSTEM

There are two basic steps to be followed in order to model the Star Schema from the related XML Schema.

Step 1: In the first step, a Schema graph is formed from the related XML Schema.
Step 2: The fact table and the dimension tables and their relationships are identified.
Step 3: 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.

A: Example of XML Schema to be Converted into schema Graph

```
<? xml version="1.0"?>
<xsd:schema xmlns:xsd="http??www.w3.org/2001/xmlschema">
<xsd:element name=”Bill” type=”Bill_key”>
<xsd:elementname="Patient_id" type=”Patient_type”>
<xsd:complexType>
<xsd:sequence>
<xsd:elementname="name" type="xs:string" use="required"/>
<xsd:elementname="Age_group" type="xs:integer" use="required"/>
<xsd:elementname="sex" type="xs:string" use="required"/>
<xsd:elementname="Payement_mode_type" type="xs:string" use="required"/>
< xsd:elementname="Date_admitted" type="xs:date" use="required"/>
<xsd:elementname="Date_Discharged" type="xs:date" use="required"/>
<xsd:elementname="address_id" type="address_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="address_type">
<xsd:sequence>
<xsd:elementname="street" type="xsd:string" use="required"/>
<xsd:elementname="city_id" type="city_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="city_type">
<xsd:sequence>
<xsd:elementname="state" type="xsd:string" use="required"/>
<xsd:elementname="country" type="xsd:string" use="required"/>
<xsd:elementname="contact_no." type="contact_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="contact_type">
<xsd:elementname="contact_type" type="xsd:string" use="required"/>
</xsd:sequence>
<xsd:elementname="Diagnosis_id" type=”Diagnosis_type”>
<xsd:complexType>
<xsd:sequence>
<xsd:elementname="Disease_name" type="xs:string" use="required"/>
<xsd:elementname="Disease_type" type="xs:string" use="required"/>
<xsd:elementname="Treatment_id" type="Treatment_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="Treatment_type">
<xsd:sequence>
<xsd:elementname="Treatment_desc" type="xsd:string" use="required"/>
<xsd:elementname="Treatment_cost" type="xsd:decimal" use="required"/>
<xsd:elementname="Equip_code" type="Equip_type" use="required"/>
```

```
</xsd:sequence>
<xsd:complexTypename="Equip_type">
<xsd:sequence>
<xsd:elementname="Equip_desc" type="xs:string" use="required"/>
<xsd:elementname="Equip_cost" type="xs:string" use="required"/>
</xsd:sequence>
<xsd:elementname="Doctor_id" type=”Doctor_type”>
<xsd:complexType>
<xsd:sequence>
<xsd:elementname="Doc_name" type="xs:string" use="required"/>
<xsd:elementname="Doc_spec" type="xs:string" use="required"/>
<xsd:elementname="Doc_dept" type="xs:string" use="required"/>
<xsd:elementname="Doc_type" type="xs:string" use="required"/>
<xsd:elementname="Doc_fee" type="xs:decimal" use="required"/>
<xsd:elementname="Med_code" type="Med_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="Med_type">
<xsd:sequence>
<xsd:elementname="Med_desc" type="xsd:string" use="required"/>
<xsd:elementname="Med_price" type="xsd:decimal" use="required"/>
</xsd:sequence>
<xsd:elementname="Ward_no." type=”Ward_type”>
</xsd:complexType>
<xsd:elementname="Ward_type" type”xs:string” use=”required”/>
<xsd:elementname="Bed_no." type="Bed_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="Bed_type">
<xsd:sequence>
<xsd:elementname="Assigned_date" type="xsd:date" use="required"/>
<xsd:elementname="Discharged_date" type="xsd:date" use="required"/>
<xsd:elementname="Bed per day_charge" type="xsd:decimal" use="required"/>
</xsd:sequence>
<xsd:elementname="Lab_id" type=”Lab_type”>
<xsd:complexType>
<xsd:sequence>
<xsd:elementname="Lab_desc" type="xs:string" use="required"/>
<xsd:elementname="Test_id" type="Test_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="Test_type">
<xsd:sequence>
<xsd:elementname="Test_desc" type="xsd:string" use="required"/>
<xsd:elementname="Test_price" type="xsd:decimal" use="required"/>
<xsd:elementname="Report_id" type="Report_type" use="required"/>
</xsd:sequence>
<xsd:complexTypename="Report_type">
<xsd:sequence>
<xsd:elementname="Report_desc" type="xsd:string" use="required"/>
</xsd:sequence>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xs:schema>
```

B: Method to develop the Schema graph from the XML Schema (Algorithm)

Step 1: 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.

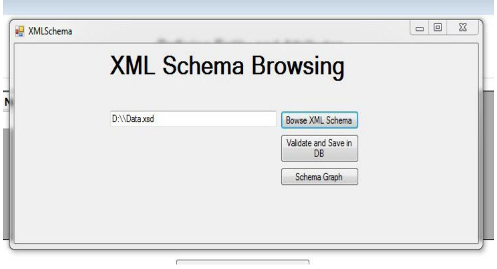


Figure 3: XML Schema as Input

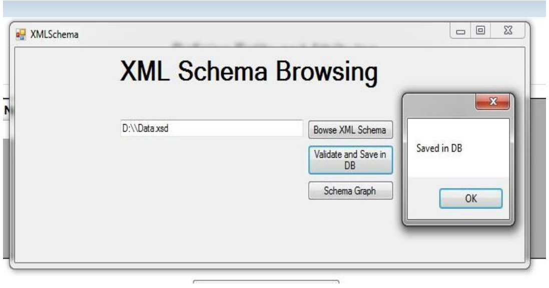


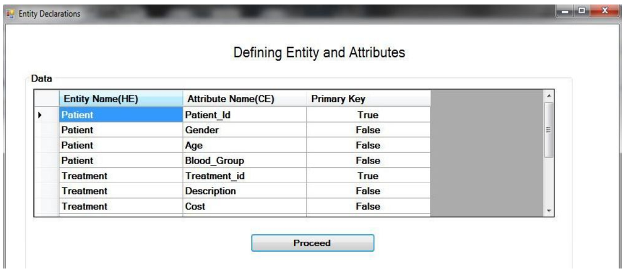
Figure 4: XML Schema Database

Step 2*:* For all **HE** (i=1 to n) perform following : (n is the total number of **HE**)

i) Find the sequence of elements under ith **HE :**

If it is an element then create a vertex for it into the graph and connect it with ith **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**.



Entity Name(HE)	Attribute Name(CE)	Primary Key
Patient	Patient_Id	True
Patient	Gender	False
Patient	Age	False
Patient	Blood_Group	False
Treatment	Treatment_id	True
Treatment	Description	False
Treatment	Cost	False

Figure 5: Finding Entities in XML Schema

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

iii) Scan the XML Schema for jth **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**.

iv) 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*/
```

__Explanation of the algorithm:__ 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.

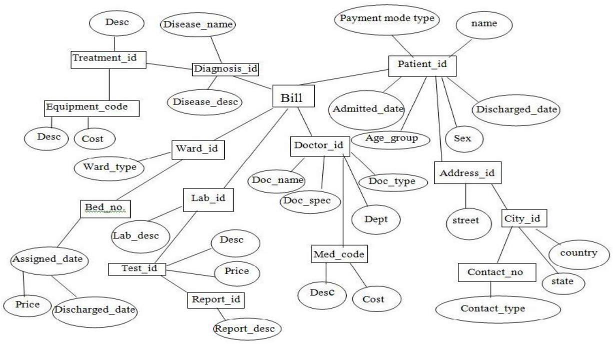


Figure 6: Example of Schema graph

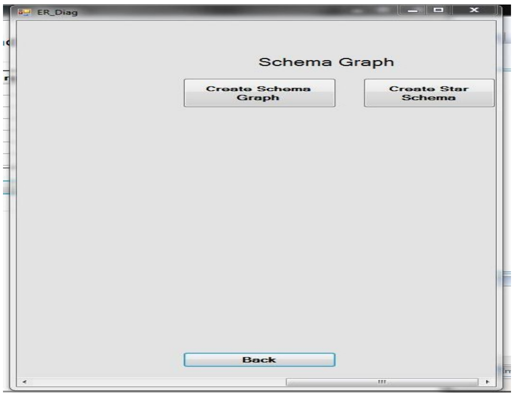


Figure 7: Schema Graph Creation

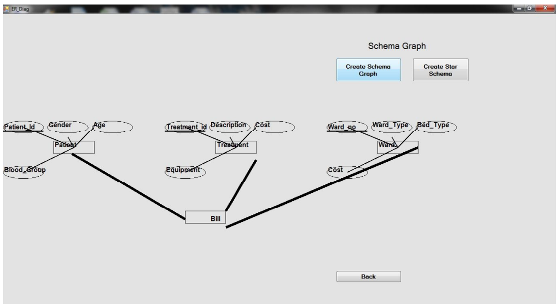


Figure 8: Schema Graph

C. Identification of the Fact table and the Dimension tables

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".

D: 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.

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.
    End For /*HE*/
```

The Star Schema builds from the above example (Figure2)

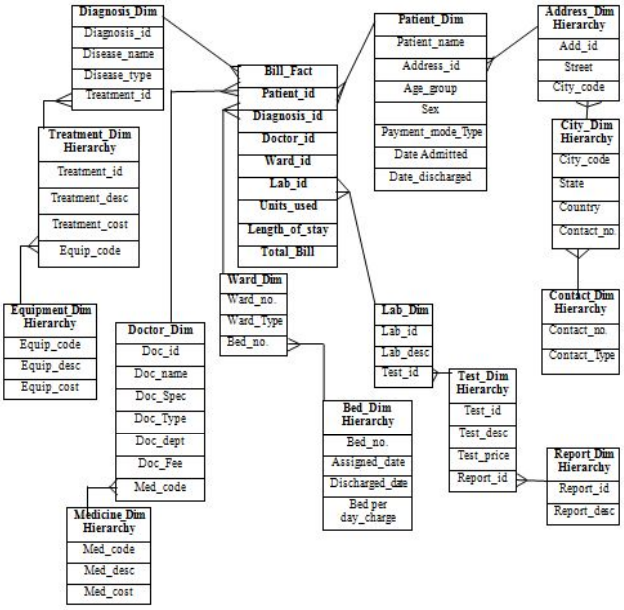


Figure 9: Star Schema generated from XML Schema

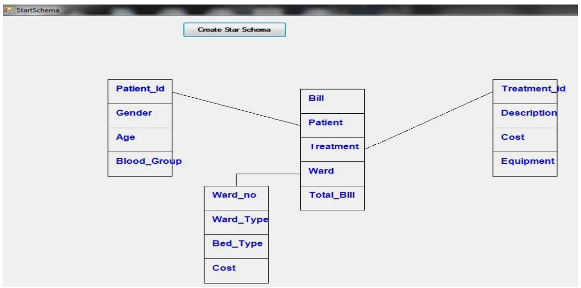


Figure 10: Star Schema From Schema graph

VI. CONCLUSION

This paper focuses on the identification of 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 methodology in this paper will be used to convert the XML Schema into the appropriate Warehouse Schema. As XML Schema offers several benefits over DTD's such as it allows the text that appears in elements to be constrained to specific types. It allows user-defined types to be created. It allows uniqueness and foreign-key constraints. Therefore, 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.

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

[2] Sarbani Dasgupta, Soumya Sen, Nabendu Chaki; "A Framework To Convert XML Schema to ROLAP"; Proc. of 2nd Intl. Conf. on Emerging Applications of Information Technology, 2011.

[3] Yuan Sun; Hexin Chen; Mianshu Chen; Xinying Wang; Aijun Sang; "Multi-dimension Multimedia Retrieval Model Implementation Based on XML Database" International Conference on Signal Processing Syatems, 2009.

[4] Rajugan, R.; Chang, E.; Dillon, T.S.; "Conceptual Design of an XML FACT Repository for Dispersed XML Document Warehouses and XML Marts", 5th International Conference on Computer and Information Technology, 2005.

[5] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau; "Extensible Markup Language (XML) 1.0 (Fifth Edition)"; W3C Recommendation; www.w3.org/TR/RECxml.

[6] Parimala N and Payel pahwa; "From XML schema to cube" International Journal of Computer Theory and Engineering; Vol. 1, No 3 August 2009.

[7] Boris Vrdoljak, Marko Banek, and Stefano Rizzi: Designing Web Warehouses from XMLSchemasY. Kambayashi, M. Mohania, W. Wöß (Eds.): LNCS 2737, pp. 89-98, 2003

[8] Wolfgang Hummer, Andreas Bauer, Gunnar Harde: XCube – XML for Data Warehouses, DOLAP'03, November 7, 2003, USA.

[9] M. Golfarelli, S. Rizzi, and B. Vrdoljak, .Data warehouse design from XML sources., Proc. DOLAP'01, Atlanta, pp. 40-47, 2001.

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

[11] Payel pahwa and Parimala N; "Conceptual design of data warehouses from xml schemas" 2nd International Conference on Intellectual Capital, knowledge management & Organizational Learning 21-22 Nov, 2005 American University of Dubai, United Arab Emirates.

[12] M. Jensen, T. Møller, and T.B. Pedersen, .Specifying OLAP Cubes On XML Data., Journal of Intelligent Information Systems, 2001.

[13] Ramanath, M.; Kumar, K.S.; "A rank-rewrite framework for summarizing XML documents" 24th International Conference on Data Engineering Workshop, ICDEW 2008

[14] Rajugan, R.; Chang, E.; Dillon, T.S.; "Conceptual Design of an XML FACT Repository for Dispersed XML Document Warehouses and XML Marts", 5th International Conference on Computer and Information Technology, 2005.

[15] Belen Vela; Carlos Blanco; Eduardo Fernandez; E.Marcos "Model Driven Development of Secure XML Data Warehouses: A Case Study". EDBT 2010, Lausanne, Switzerland.

[16] Kimball,R, and Ross , The Data Warehousing Tool kit , John Wiley & sons,2002

[17] Dov Dori , Roman Feldman, Arnon Sturm. Transforming an Operational System Model to a Data Warehouse Model: A Survey of Techniques, Proceeding of the International Conference on Software- Science, Technology & Engineering (SwSTE'05)

[18] Khurram Shahzad ,Abid Sohail ," A Systematic Approach for Transformation of ER Schema to Dimensional Schema"