ROLAP Based Data Warehouse Schema to XML Schema Conversion

Soumya Sen
A. K. Choudhury School of Information Technology
University of Calcutta
iamsoumyasen@gmail.com

Agostino Cortesi
Computer Science Department
Ca Foscari University
cortesi@unive.it

Nabendu Chaki
Department of Computer Science & Engineering
University of Calcutta
nabendu@ieee.org

***Abstract*—Data Warehouse is one of the powerful tools for analytical processing. XML on the other hand is widely used to handle data in web environment. XML to data warehouse integration is a subject of interest for the business organization to use the semi-structured XML for analytical processing. However, in this research work we approached the problem in reverse direction. Here we generate equivalent XML schema from the existing data warehouse schema for an organization which does not has the XML platform to manage the web data. The proposed reverse engineering framework uses one of the existing methodologies of converting the XML schema to data warehouse schema. However, we have applied it in a reverse approach. Moreover we have established a formalism to prove the soundness and correctness of both the conversion mechanisms.**

Keywords—Data Warehouse Schema; ROLAP; XML Schema; Schema Graph; Formalism; Reverse Engineering

I. INTRODUCTION

A data warehouse is a subject-oriented, integrated, non-volatile, time-variant representation to organize data for analytical processing. The integrated property of data warehouse emphasize its capability to work with heterogeneous data sources. The most widely used data warehouse is called ROLAP (Relational Online Analytical Processing), where analytical data is represented in relational format. Therefore the different types of source data of OLTP (Online Transactional Processing) are converted to ROLAP to perform the analytical processing on heterogeneous data in a unified approach. There are numbers of way to represent OLTP data. In this research work we focused on XML (Extended Mark-up Language) as in web environment XML is the most important language to represent transactional data. Moreover XML is semi-structured in nature where as relational model is structured. Hence this conversion has additional challenge to work with two different types of data model.

XML data is represented in terms of XML DTD or XML schema. However XML DTD has several limitations. XML DTD did not fully support user requirements. Moreover neither the DTD has built-in data types nor supports user-derived data types. Along with this DTD allows only limited control over cardinality. XML Schema has been designed to provide a robust mechanism to define XML document structure and limitations. XML Schemas are capable to represent XML documents. They reference the XML schema namespace and even have their own DTD process. As XML schema has several advantages over XML DTD the research interest is more on XML schema.

XML schema design generally follows 2 types namely Russian Doll Design [1] and Salami Slice Design [1]. The Russian Doll design has a single global element that nests local elements. The Salami design corresponds to having all of the elements defined within the global namespace and then referencing the elements. Russian Doll design has more use in the Industry and therefore in the research work also this schema structure gets more priority.

In the next section we would discuss on different approaches of XML to data warehouse conversion method. However in this research work we view the problem from reverse side, where data warehouse schema is converted to XML schema. This seems to us as a potential problem because numbers of organization are deploying their existing business in web environment. In some cases they already have data warehouse for analytical processing which is build based on other OLTP languages except XML. Thus the reverse engineering solution that we propose and validate in this work bears high relevance in the state of the art context. Majority of the organizations use Relational OLAP (ROLAP) to implement their data warehouses. Therefore to incorporate web based environment in the existing implementation it could be an intelligent decision to use database structure in web which is equivalent to the ongoing system.

II. RELATED WORK

In this section at first we discuss about different existing approaches of converting data warehouse schemas from the XML sources. Among this we would choose one of the efficient methods and then approach with that method in reverse way. We would also incorporate formalism on these methods to proof the correctness and soundness and to establish the reverse engineering.

Data warehouse schema conversion is performed on both XML DTD and XML schema. At first we discuss few methods based on XML DTD. In [2], algorithms were proposed to automatically construct UML diagrams from XML data and the application of the diagrams on the conceptual design of (virtual) data warehouses were based on web data. The UML diagram has been chosen here because UML is a standardized conceptual data modeling language

978-1-4673-8075-1/16/$31.00 ©2016 IEEE

and is powerful enough to express a document described by a DTD. A semi-automatic approach for conceptual designing of a data mart from XML DTD was described in [3]. It [3] explained how the semi-structured nature of the source increases the level of uncertainty on the structure of data, thus requiring access to the source documents and need to ask the designer to find out one-to-one or many-to-one relationships. However the sources were constrained by a DTD using sub-elements. In the previous section we have discussed about other limitations of XML DTD. XML schema overcomes the shortcomings of XML DTD. Now we discuss some approaches based on XML schema. XML schema conversion to OLAP cube by identifying fact table and dimension tables has been showed in [4]. OLAP cube formation using XML source is an important area of research. Conceptual designing based on dispersed XML documents has been done to form both XML warehouse and XML marts [5]. Another research work on this multi-dimensional model based on XML database has been carried out specifically for multimedia data [6]. As the size of multimedia database is usually huge the work [6] is significant for handling high volume data. A generic work on XML schema shows how to convert the contents of the XML schema to multiple schemas of the multi dimensional model [7]. Further the work has been extended to design multiple cubes [8] of multidimensional model from XML schema. A semi-automatic approach [9] was proposed for XML data warehouse design starting from XML schemas as data sources. It generates numbers of UML class diagram from XML schema and then the numbers of classes are reduced using a set of rules. Finally a multi-dimensional (MD) element extraction algorithm [9] is used to automatically identify facts, measures and their corresponding dimensions. An automatic approach for designing the logical schema for a data mart starting from the XML schema describing XML sources using UML and QVT transformation language was described in [10]. It [10] showed a simplification process and a set of rules that applies successive transformations to create the star schema. All of these generated schemas are converted to star schema only. In order to address the other schemas, a formalization method to model star and snowflake schema within XML schema based on attribute tree was proposed and termed as X-Warehousing [11]. It merges users analysis objectives represented through XML schema with XML data sources. A secure data warehouse [12] was proposed on XML schemas by focusing on the security issues relevant to XML schemas. In another research work [13] XML schema to data warehouse schema has been done at first by converting the XML schema to ER-diagram. In the next phase ER-diagram has been converted to ROLAP based data warehouse schema. As ER diagram is generated we could easily convert this to relational model also. The main significance of the work [13] is it supports both OLTP (through ER- diagram and relational model) and OLAP (through data warehouse schema). However in this research work only star schema and snowflake schema are identified. This limitation has been sorted out in [14], where the fact constellation is also identified. The proposed methodology in this paper is capable of accepting multiple related XML schemas. The XML schemas of [14] follow Russian doll design. The given XML schema(s) is converted to a data structure named as Schema Graph. In the next phase Schema Graph is converted to data warehouse schema.

We choose the method of [14] as it can work with multiple XML schemas and supports star schema, snowflake schema and fact constellation. In the next section we briefly describe the proposed framework of [14] and then we propose the reverse methodology based on [14] to generate XML schema from the existing data warehouse schema.

III. A Brief Overview of the Framework in [14]

The proposed framework of [14] accepts more than one related XML schemas. The proposed algorithm [14] has two phases. At the first phase XML schemas are converted to a new data structure named as Schema Graph. Once the Schema Graph is constructed, then in the next phase data warehouse schemas are generated and the type of the schema is identified.

Schema Graph is a level wise separable graph. Every entity of XML schema acts as a vertex in Schema Graph and the name of the vertex is same as the entity name in the XML schema. The entities that appear in the Schema Graph are classified into three types.

A. Holder Element (HE): These elements that have no predecessor in the Schema Graph. Holder Elements are placed at the Level-1 of the graph.

B. Contained Element (CE): These elements are directly connected to the HEs and are called Contained Elements. Contained Elements are placed at the Level-2 of the graph.

C. **Secondary Elements (SE)**: The elements that are directly connected to the CEs are called Secondary Elements. They are placed at the Level-3 of the graph. If elements in the graph appear as connected to SE, they would be placed in level-4. The new vertices that would be connected to the vertices of level-4 would be placed in level-5 and so on. Subsequently new level could be created if the new entities appear in the graph connected to the previous level. All the elements beyond level-3 are termed as Secondary Elements.

A generic structure of Schema Graph is shown in *Fig. 1.*

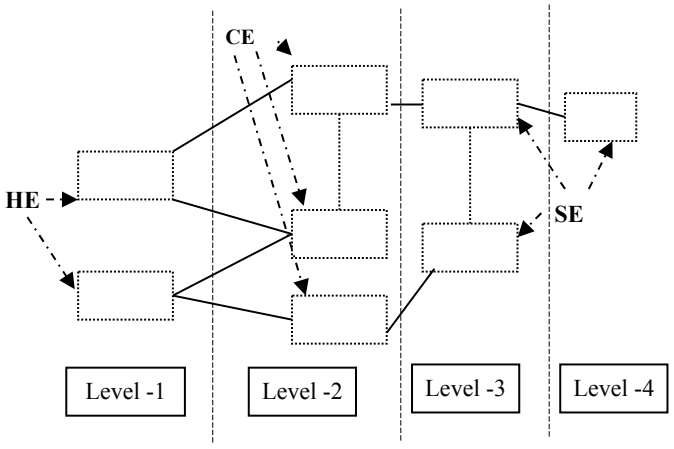

Fig. 1: Schema Graph along with HE, CE and SE

Once the Schema Graph is constructed fact table and dimension tables are identified. If some of the entities do not

have sufficient attributes to form the primary key a key attribute is added to those entities. This is necessary as the ROLAP implementation which is based on relational model requires primary key for each table.

In the proposed methodology of [14] each HE corresponds to a fact table and makes an entry in the fact table, the key attribute of the CEs that are connected to the HE are placed in the corresponding fact table for that HE. CEs appear as the dimension tables. If SEs are found connected with CE the primary keys of SEs are placed in CE. If SEs are present even after level-3, primary keys of the higher level are placed in the table corresponding to the SE of immediate lower level. After this the type of data warehouse schema is identified. A data warehouse schema is identified as star schema if the schema graph consists of HE and CEs only. Snowflake schema is identified if the schema graph consists of HE, CEs and SEs. If there is more than one data warehouse schema the framework checks whether fact constellation is present or not. If it is found atleast one dimension table is shared by more than one fact tables then the overall data warehouse schema is marked as fact constellation.

IV. PROPOSED METHODOLOGY TO GENERATE XML SCHEMA FROM DATA WAREHOUSE SCHEMA

In this section we introduce the framework to generate XML schema from the given data warehouse schema based on the proposed methodology of [14] but in the reverse way.

Once an organization decides to convert the data warehouse schema to XML schema they need to decide which dimension table to act as the root element of the XML schema. They have to select one of the dimension tables from those which are directly connected to the fact table. Otherwise system would randomly choose one of the dimension tables from those which are directly connected to the fact table. This dimension table is named as First Dimension Table (FDT). FDT would appear as HE in Schema Graph. Rest of the dimension tables those are directly connected with fact table are categorized as Connected Dimension Table (CDT). CDTs would correspond to CE in the Schema Graph. Other dimension tables which are not connected to the fact table are categorized as Secondary Dimension Table (SDT). SDTs would correspond to SE in the Schema Graph.

In this research work we also establish the correctness of the method described in [14]. Hence we proof that once an XML schema has been converted to data warehouse schema using the reverse methodology we would get back the original schema. However in some cases newly generated XML schema may differs from the old one. As primary keys has been added for those entities which did not have sufficient attributes to form the primary key. In these cases we can claim that our proposed methodology helps to re-generate better XML schema which is more structured than the original. In this case we have the knowledge of HE, CE and SE. Thus the dimension tables correspond to HE, CE and SE are categorized as FDT, CDT and SDT respectively.

A. Methodology to Construct Schema Graph from Data Warehouse Schema

Here we form Schema Graph. As Schema Graph is a level wise separable graph HEs are placed at the most left and labelled as level-1. CEs are placed at right to respective HEs and labelled as level-2. CEs are also connected to the respective HE. Now SDTs are placed in Schema Graph level wise from level-3 onwards. The attributes corresponding to each entity of the Schema Graph are connected to the respective entities.

If there is more than one fact table the above process is repeated for each of them.

Algorithm:

Step 1: N = Numbers of fact table in the system
Step 2: FOR J = 1 to N repeat the following steps
Step 3: Find out the First Dimension Table (FDT) for each J.

FDT is either given by the user or already known if the data warehouse schema has been constructed from some XML schema. If the user does not specify it then the system randomly chose any one of the dimension table among those which are directly connected to fact table.

Step 4: FDT corresponds to the first level elements of the Schema Graph. These are the Header Element (HE) of the Schema Graph and placed at the level-1 of the Schema Graph
Step 5: The attributes corresponding to each FDT are also connected as attributes to the respective HE of the Schema Graph.
Step 6: The dimension tables except the FDT which are connected to the fact table are termed as Connected Dimension Table (CDT). CDTs appear as the level-2 elements of the Schema Graph. These are the Contained Element (CE) of the Schema Graph. CEs are connected with the respective HE in Schema Graph.
Step 7: The attributes corresponding to each CDT are also connected as attributes to the respective CE of the Schema Graph.
Step 8: Other dimension tables which are neither FDT nor CDT are termed as Secondary Dimension Table (SDT).
Step 9: Dimension tables connected with FDT and CDTs appear as the level-3 elements of the Schema Graph. These are the Secondary Element (SE) of the Schema Graph. SEs are connected with the respective CE in the Schema Graph.
Step10: The attributes corresponding to each dimension table at this level are also connected as attributes to the respective SE of the Schema Graph.
Step 11: IF there are further Secondary Dimension Tables (SDT) in the schema THEN

a) I=3

b) Repeat Steps 12 to 14 until all the dimension tables are not included in schema graph

Step 12: IF there are SDTs which are connected with the dimension tables correspond to the I^{th} level of schema graph THEN

Place the SDTs of $(I+1)^{th}$ level in the Schema Graph and connect with the elements at I^{th} level. These new elements are also called Secondary Element (SE) in Schema Graph.

Step 13: The attributes corresponding to each dimension table at this level are also connected as attributes to the respective SE of the schema graph.
ENDIF /*Corresponding to IF of Step 12*/
Step 14: I=I+1
End of Repeat /*Corresponding to Step 11 b) */
ENDIF /*Corresponding to IF of Step 11*/
Step 15: ENDFOR /*Corresponding to Step 2*/

B. XML Schema Generation from Schema Graph

After getting the Schema Graph, we head forward to the last step of generating the XML schema. As we are dealing with Russian Doll types of XML schema we use the concept of the nesting of elements.

We denote the HE as the root of the XML schema. CEs are nested under the root element separately in the XML schema. SEs corresponding to the level-3 of Schema Graph is nested under respective CE. If there are further levels of SEs they are nested under their predecessor level of SE of Schema Graph in the XML schema.

Algorithm:

Step 1: Repeat FOR every element at level-1 or Header element (HE)
Step 2: Each HE corresponds to root element of XML schema
Step 3: All the elements at level-2 or Contained Element (CE) of the schema graph connected with the particular HE is nested under the root element separately.
Step 4: FOR each element of level-2 of the schema graph find the elements connected at level-3 or Secondary Element (SE) and nest them under the element correspond to level-2.
I=3
Step 5: Repeat till all the levels are traversed
a) Select the elements at $(I+1)^{th}$ level in Schema Graph which are connected with the I^{th} level elements in Schema Graph.
b) The selected elements of previous step is nested under I^{th} level elements in XML schema
c) I=I+1
End of Repeat
ENDFOR /*Corresponds to FOR of Step-4*/
Step 6: ENDFOR

The type of each attribute is obtained from data warehouse schema definition.

V. ILLUSTRATION WITH EXAMPLE

In this section, we present an example to describe the execution of our methodology. We are starting with a given data warehouse schema as shown in *Fig. 2*. The given schema consists of single fact table named Flightorder_fact. The measure is given as No. of Tickets.

Here we explain the stepwise execution of the algorithm of sub-section-(IV.A) to construct the Schema Graph.
Step 1: N=1 as the schema has one fact table
Step 2: The following steps are going to be executed only once
Step 3: We take FlightOrder as First Dimension Table (FDT) as this dimension table has the same name as fact table.

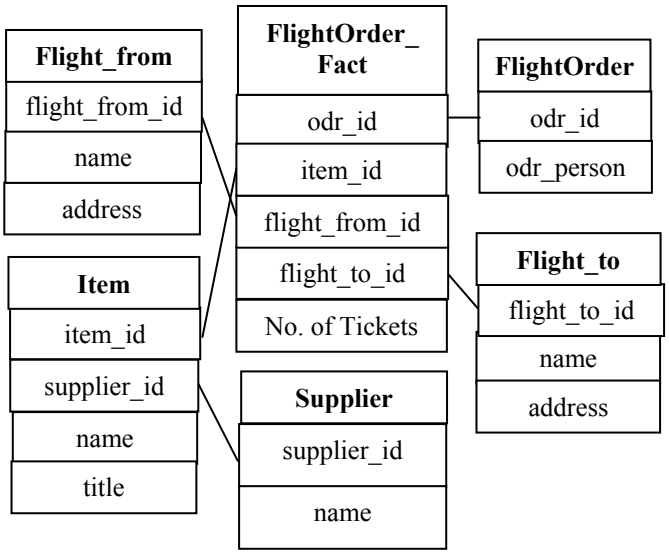

Fig. 2: A Data Warehouse Schema

Step 4: FlightOrder is going to be the HE of the Schema Graph.
Step 5: The attributes of FlightOrder are connected with the HE in Schema Graph.
Step 6: All other dimension tables except FDT that are directly connected with fact table are termed as Connected Dimension Table (CDT). In this example CDTs are Item, Flight_to and Flight_from. All these CDTs are placed in level-2 of Schema Graph and denotes as CE.
Step 7: The attributes of Item, Flight_to and Flight_from are now added with these CEs in the Schema Graph.
Step 8: Other dimension tables are termed as SDTs. Here Secondary Dimension Table (SDT) is Supplier.
Step 9: Supplier is placed at the level-3 of the Schema Graph. Supplier acts as SE in the Schema Graph and also connected with the CE Item in the previous level.
Step 10: The attributes of Item are connected with it in the Schema Graph.
Step 11 to 14: These steps are not executed as there is no further SDT in the Schema Graph.
Step 15: End of Algorithm Execution

The output of the above execution is shown in *Fig. 3*. Element FlightOrder is at level-1, elements Item, Flight_to and Flight_from are at level-2 and finally the element Supplier is at level-2 are shown in Schema Graph.

Finally the XML schema is build by applying the algorithm of sub-section-(IV.B) on the data warehouse schema of *Fig. 3*.

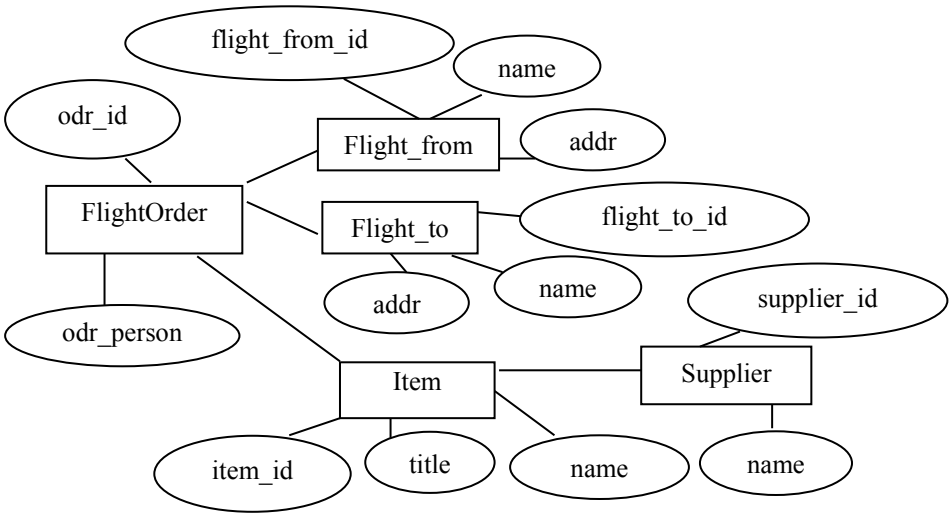

Fig. 3: Schema Graph corresponding to the data warehouse schema of Fig. 2

The XML schema is given below.

```
<xsd:elementname="FlightOrder">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:elementname="odr_id" type="xs:string"
       use=”required”>
      <xsd:elementname="Flight_from"
       type="FlightfromType">
        <xsd:sequence>
          <xsd:elementname="flight_from_id"
           type="xsd:string" use="required"/>
          <xsd:elementname="name" type="xsd:string"
           use="required"/>
          <xsd:elementname="addr" type="xsd:string"
           use="required"/>
        </xsd:sequence>
      <xsd:elementname="Flight_to"
       type="FlighttoType">
        <xsd:sequence>
          <xsd:elementname="flight_to_id"
           type="xsd:string" use="required"/>
          <xsd:elementname="name" type="xsd:string"
           use="required"/>
          <xsd:elementname="addr" type="xsd:string"
           use="required"/>
        </xsd:sequence>
    <xsd:elementname="Item" type="ItemType">
      <xsd:sequence>
       <xsd:elementname="title" type="xsd:string"
        use="required"/>
       <xsd:elementname="name" type="xsd:string"
        use="required"/>
       <xsd:elementname="Supplier"
        type="SupplierType" use="required"/>
      </xsd:sequence>
        <xsd:complexTypename="SupplierType">
          <xsd:sequence>
            <xsd:elementname="name"  type="xsd:string"
use="required"/>
             <xsd:elementname="supplier_id"
type="xsd:string" use="required"/>
          </xsd:sequence>
        <xsd:attributename="odr_person"  type="xsd:string"
use="required"/>
      </xsd:sequence>
    </xsd:complextype>
</xsd:element>
```

VI. Formalism on XML schema to and from Data Warehouse Schema Conversion

In this section we are going to proof that once an XML schema is converted to the data warehouse schema using the method of [14] and when we get back the XML schema from the converted data warehouse schema applying the methodology of this paper they are equivalent to each other. In fact, the re-generated XML schema is often better than the original XML schema in terms of structure. This is because, the primary keys are added during XML to data warehouse conversion to those XML elements not having sufficient attributes to form the primary key. From this point onwards if we continue the conversion mechanism in both ways the result would be same. It is depicted in *Fig. 4.*

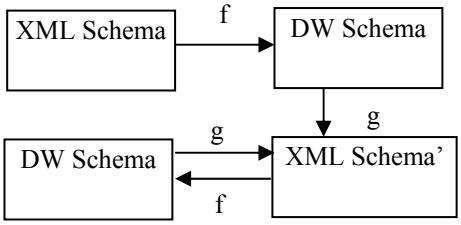

Fig. 4: XML schema to/from data warehouse schema conversion

This could be formalized using closure operator

$\rho: L \rightarrow L$ is a closure if

$\rho(L) = \rho(p(L)$

In our setting, we may say that $\rho = g \circ f$ is a closure operator.

It is to be noted that

$XML \rightarrow DW \rightarrow XML' \geq XML$.

The above formalism ensures the correctness and soundness of both of the conversion mechanism.

VII. CONCLUSION

This research work contributes the new idea of developing XML schema from existing data warehouse schema. Moreover, as the proposed method is conceptualized based on an existing method but applying it in the reverse way - the concept of reverse engineering is also incorporated in this work. The proof of formalism on both of these methods guarantees the correctness as well as soundness. Hence, the reverse engineering is deployed successfully.

This research work opens a new area of research where the data warehouse schema could be converted back to the heterogeneous data sources from which the data warehouse has been integrated. This would help in developing the reverse methods and therefore reverse engineering could be practised with greater intensity shed by verification or testing or formalism.

References

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

[2] M.R.Jensen, T.H.Moller ,T.B. Pedersen; "Converting XML Data To UML Diagrams For Conceptual Data Integration" , 1st International Conference on Data Integration over the Web (DIWeb) at 13th Conference on Advanced Information Systems Engineering (CAISE'01), 2001.

[3] M. Golfarelli, S. Rizzi, and B. Vrdoljak, "Data warehouse design from XML sources" Proc. of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP'01), Atlanta, pp. 40-47, 2001.

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

[5] 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.

[6] 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 Systems, 2009.

[7] 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.

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

[9] Z.Ouaret, O.Boussaid, R.Chalal "A global and comprehensive approach for XML data warehouse design" Proceedings of the 11th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), 2014.

[10] Z. .Ouaret, O.Boussaid, R.Chalal "Towards the automation of XML data warehouse logical design" Proceedings of the 9th International Conference on Digital Information Management (ICDIM), 2014.

[11] O. Boussaid, R.B.Messaoud, R.Choquet, S.Anthoard "X-Warehousing: an XML-Based Approach for Warehousing Complex Data" Proceedings of the 10th East European Conference, (ADBIS) Thessaloniki, Greece, September 2006.

[12] Belén Vela, Carlos Blanco, Eduardo Fernández-Medina, Esperanza Marcos "Model Driven Development of Secure XML Data Warehouses: A Case Study" EDBT 2010, March 22–26, 2010, Lausanne, Switzerland.

[13] Sarbani Dasgupta, Soumya Sen, Nabendu Chaki "A Framework To Convert XML Schema to ROLAP"; IEEE Proc. of the IEEE Second International Conference on Emerging Applications of Information Technology (EAIT 2011), Kolkata, India, 2011.

[14] Soumya Sen, Ranak Ghosh, Debanjali Paul, Nabendu Chaki "Integrating XML Data Into Multiple Rolap Data Warehouse Schemas" AIRCC International Journal of Software Engineering & Applications (IJSEA); Volume 3, Number 1, January 2012.