Assimilation of Heterogeneous Resources by Utilizing a Unified Format of XML for Constructing OLAP Cubes

Sohail Asghar
Faculty of Engineering and Applied Sciences
Mohammad Ali Jinnah University, Islamabad, Pakistan
e-mail: sohail.asghar@jinnah.edu.pk

Rukhsana and Saqib Subhan
Shaheed Zulfikar Ali Bhutto Institute of Science and Technology
Islamabad, Pakistan
e-mail: rukhsanahunzai@hotmail.com

Simon Fong
Faculty of Science and Technology
University of Macau
Macau SAR
e-mail: ccfong@umac.mo

***Abstract*— Data explosion in quantity and diversified sources of data in variety pose certain problems in data storage on heterogeneous repositories for online analytical processing (OLAP). Over the years, Extensible Markup Language (XML) is emerged as a standard language for integrating and exchanging structural and semi-structured data from heterogeneous and miscellaneous data sources. In this paper we presented a multi-layered model for resolving the complexity of heterogeneous resources as well as multi formatted data. Specifically, the model amalgamates data from an assortment of resources by using XML and transforms them in to data cubes namely X-cubes. Likewise, XML is incorporated into OLAP for generating multidimensional cube called XOLAP. To support this new data cubes, a query translator as a user interface is implemented. A significant contribution of this model is that it facilitates gathering data from heterogeneous data sources and provides a unified format for handling OLAP queries in order to construct multidimensional cubes.**

Keywords- OLAP, XML, Data Integration, Data Warehouse and Query translator

I. INTRODUCTION

With the growth of, widespread and increasing popularity of the Internet the number of data sources available for public access is rapidly increasing both in number and size, while at the same time users and application programs increasingly need to combine data from these different autonomous and heterogeneous data sources [1]. In accumulation to this heterogeneous data are quite complex since they concern different types of information, coming from different sources, and presented on different supports [2]. Furthermore, most modern applications require the simultaneous management of diverse types of data [3]. In order to integrate data from these resources there is a requirement for a language which provides a unified platform and a universally accepted format such as XML. It is known that [4] almost all external data from various sources can be effectively wrapped in XML format.

In the literature there are a number of previous works that contribute to the research of integrating XML and OLAP, one way or another. All of them however have limitation in some aspects. Some selected ones are discussed as follow.

II. RELATED WORKS

With the growth of, Theodorators et al. [5] utilized an OLAP cube for handling multidimensional data with the help of multidimensional query language. In combination with this, a model of Metadata (MD) has been proposed. Moreover, they have defined an aggregation and utilized selection operators for performing drill-down, roll-up, slicing, dicing and pivoting. At the end an instance algebraic expression had been used for MD query finding for cube computation. Potency of their occupation is proposed model reduce query response time to improve availability of data. Limitation of their work is that an optimizer that selects an optimal plan for computing MD queries on cubes using other cubes need to be designed and tested, and embodied in the query optimizer of the underlying query processing layer.

Golfarelli et al. [6] designed a model from XML data source in order to construct a data mart by using Semi-structural procedure. They have utilized both schemas as well as data type definition of XML. In accumulation with this they have presented a conceptual level with the help of logical schema by which data marts can be extracted. An algorithm is also presented in order to construct facts, measures and hierarchies. Strength of their work is that multidimensional design of data warehouse is directly carried out from XML source. Main limitation of their work is that requirements of users cannot be fully supported by existing data because of complexity arise in schema source.

Pedersen et al. [7] suggested a multidimensional SQLm OLAP model. Some languages used for constructing this model also allowed integration of OLAP with XML. In addition with this some operators have been offered like selection and projection in order to support dimensional hierarchies, automatic aggregation and correct aggregation of data. In addition to that UML schema has utilized for integration and OLPA cube construction. Meanwhile, they developed OLAP model, algebra and query languages for getting both powerful and SQL compatibility. Strength of their work is that use of multidimensional SQLm data model, formal algebra and high-level query language is powerful in getting accurate results. Limitation of their work is that pre-aggregation for irregular hierarchies.

Pan et al. [8] developed architecture for a fast and powerful extraction and combination of information from various heterogeneous resources. Additionally, it also

combines data from structured and semi-structured sources to create a unified global schema for multi information from diversified sources. Furthermore, Denodo platform mediator system architecture has been represented through graphics for showing mediators. A query language has used to generate a global schema to generator uniform system for heterogeneous data sources. Strength of their work is that the Denodo Platform, constructed for mediator's architecture, allows for a fast yet powerful extraction and combination of information from various heterogeneous, structured or semi-structured sources, to create a unified global schema for such information. Limitation of their work is that there is a need for finding more automated ways for mapping format and semantic heterogeneities between sources.

Kit et al. [9] proposed an XML-OLAP system. This provided main algorithms for XML data analysis. In addition to that they have developed some algorithms for Topological Rollup. Like this other main algorithms developed are Speed up the computation, TDA (Top-Down Algorithm), BUA (Bottom-Up Algorithm) and some enhanced versions of BUA for mixed structure. After this they have applied a very large XML data aggregation for achieving information on XML data. Main strength of their work is that due to utilization of topological algorithm efficiency of a system will increase. Limitation of their work is that topological operator's enhancement is required for logical integration of the proposed system.

Pedersen et al. [4] adopted another approach for federation of XML and OLAP data which represented in terms of a formal data model and algebraic query language. In addition to this a federated query language were introduced, SQL-XM and X- path for multidimensional querying. Multidimensional cube has used to represented multidimensional data and some operation on cube have introduced like selection, aggregation and its grouping. Finally a prototype implementation and effective optimization algorithms and techniques have been introduced to reach to optimal solution. Strength of their work is that logical federation handle issue regarding dimensions, hierarchies and correct aggregation of data. Limitation of proposed model is that most important bottleneck in federation is moving of data from XML to OLAP components. So to overcome these efficient optimal techniques are required.

Frank et al. [10] applied prototyping techniques in order to integrate heterogeneous data using XML languages and techniques. These techniques basically transformed all cube data into XML formats in order to integrate data into global cube by utilizing Xquery. Furthermore, general architecture has been proposed in order to construct virtual cube and store the relational informational and construct multidimensional file structure. Additionally they have also presented MDX and Xquery for querying cube. This model makes systematic integration methodology for integrating heterogeneous data warehouse through internet. Main strengths of their work are as follow. 1) Transformation of all cube in to XML formats, will resolves the semantic discrepancies. 2) The integration of data into global cube through Xquery will use to utilize the integration power of internet. Main limitation of their work is that data cube size is very large which may be time consuming. They applied prototyping techniques to integrate heterogeneous data using XML languages and techniques but an implementation of OLAP multidimensional cubes of dispersed domains seem to be lacking. Also it may need a query translator as a part of the prototype for a user friendly environment.

Boussaid et al. [2] proposed a multidimensional conceptual model which enables utilization of conceptual model of XML as a logical modeling formulization of data warehouse. Furthermore, they have been focused on warehousing of complex data namely as X-warehousing. Additionally logical model of XML warehouse schema defined referenced data cube model. An attribute tree has been constructed which enable representation and transformation of XML schema and functioning of pruning and grafting. They have presented XML documents at real OLAP facts. Moreover, the whole warehouse documents modeled at physical model and represented in form of OLAP cubes. Main strength of their work is modeling of complex heterogeneous data in order to build X-warehouse and XML cubes through uniform XML format. Issues of their work include problem exists in updating of XML cube; and some optimization techniques are needed on the model loader module architecture.

We have briefly shown as above the assessment and disparity among proposed models and efforts of different researchers. In incorporation to that we only provide an overview to those writers who really have similar aspects but have carried out by unrelated or diversified approaches. So by carrying out the comparison to our approach with that of Theodorators [5] and Niemi et al., [11] applied similar concepts of OLAP MD cubes. But in contrast to our approach they have not taken XML as a unified format and have not developed a prototype for generating XOLAP cubes of dissimilar domain on user request. Similarly Pokorny [12] has developed a model of XML star schema matching concept to proposed model but slightly different technique in [12] which is a tree structural algorithm concept.

In a similar manner Pedersen et al. [7] has integrated XML with OLAP for cube construction like that of our model; but in distinction to our approach they have utilized UML star schema for constructing multidimensional cube and there lacks of a concept of query translator for handling user need and requirements. Pan et al. [8] developed architecture for a fast and powerful extraction and combination of information from various heterogeneous resources. It reflects the similar concepts to proposed model. Another similar concept is utilization of global and local schema [8]. It has been discussed that utilization of multivalent resources is common to all writers as a supporting concept. Pokorny et al. [13] have taken heterogeneous resources of data and utilized XML star schema for mapping it. But the application of semi-joining concept is a divergent idea. Also, Chen et al. [14] suggested an algorithm for inclusion of mapping of schema is needed to search data gathered from heterogeneous data sources although utilization of tree representation is required.

III. PROPOSED MODEL

Our proposed model is basically extended from the existing work of Pederson [7] by enhancing some issues in his system. In [7], a federation manager is utilized as a query translator. We proposed to add on by interleaving a new component on a third layer (query processing layer) called query translator or query evaluator. Also in our model multi-resources are utilized instead of only XML sources. Our approach will need to assemble several sources for data assortment in a varied and challenging environment under a uniform XML format and engender OLAP cubes namely XOLAP cubes here. These cubes basically signify fluctuated data by utilizing different type of schemas, tools and techniques of OLAP.

Our model is called Multilayered Architecture of Uniform Distributed XOLAP Cubes as shown in Figure 1. It is composed of four main layers: data layer, storage layer, query processing layer and client layer which are described as follow.

A. *Data Layer*

Data Layer acquires data from heterogeneous data sources like warehouses, XML, flat files and relational databases, etc. Inside our system, XML is used as a unified representation format to which gathered data are transformed. DTD and that of its schema implementation are applied. DTD is basically implemented to gathered plane text in form of documents while XML schema implementation is used for gathering information and constructing X cube in order to extract knowledge from data. Major techniques used in XML are for query processing such as XQuery, Xpath and XSLT.

Second repository is a data warehouse which is a massive data storage typically housed over a cluster of servers. The data warehouse is as a centralized repository for all data generated by departments and units of a large organization. Advanced data mining software is required to extract meaningful information from a data warehouse. Third repository is a pool of flat files which is also one of the data sources. Furthermore, after compilation of data from both flat files and data warehouse it will gather them into analysis server that hosts the OLAP data.

B. *Storage Layer*

After data are obtained from different sources they are converted to XML cubes called XOLAP with uniform distributed XML cubes. For data that are already in XML format source will directly be converted into cubes. Data from data warehouse and flat files are converted into relational cubes with the help of query languages like MDX and SQL. Finally the data will convert into Xcubes by applying xquery, xpath and other operation of XML. Consequently these cubes can store multidimensional data as well as providing flexibility for user to extract data by his demands. Using XML can effectively solve the problems of data exchange and sharing between different systems and different data sources. Through analyzing those heterogeneous data integration system used XML technology, all of these systems can effectively solve syntax heterogeneous except semantic heterogeneous between data sources. Metacube in this layer gathered data about all cubes and data about query translator.

C. *Query Processing Layer*

The core of this layer is a query translator and evaluator which gather data from multidimensional XML cubes. User will be able to get data on his own demand by querying and translator will translate the data.

D. *Client Layer*

In this layer there is a user interface manager which handles the query processing for user. This is how it works. User first submits requests to user interface manager for extraction of information, and then it will pass the query to translator by evaluating it. User will be able to access to the required information at this layer.

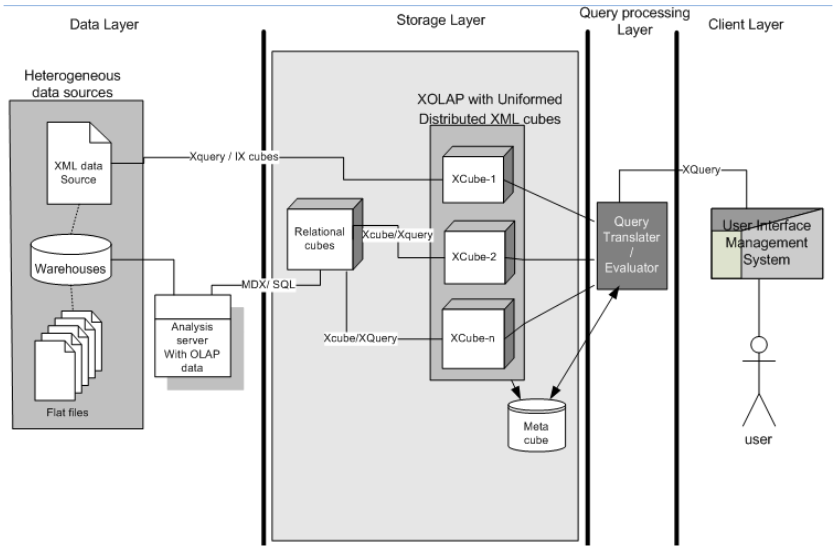


Figure 1. Multilayered architecture of uniformed distributed XOLAP cubes.

IV. VALIDATION AND IMPLEMENTATION

This section walks through the implementation of the proposed model that comes in two parts: XOLAP cube generation and the query translator. We took data from data warehouse and flat files, and respectively and converted them into XML data format. Then the system generates the relevant multidimensional cubes. The main algorithm for query translation used in the development is also provided.

A. *XML Cubes Generation*

Figure 2 shows an example of elaborate channel dimension of sales domain in the form of XML schema which is named as channel star-schema of XML data. This represents the source code of XML data from where the channel data is representing. It basically shows the dimensions, attributes, hierarchies and levels of channel as ingredient for constructing an OLAP cube.

```
<Create  Id="Action97">
  <ActiveObject >
    <Dimension  Name="CHANNEL" LongName="Channel" ShortName="Channel" PluralName="Channel" Id="CHANNEL.DIMENSION" Schema=
      <Attribute  Name="LONG_DESCRIPTION" LongName="Long Description" ShortName="Long Description" PluralName="Long Descr
      <Attribute  Name="SHORT_DESCRIPTION" LongName="Short Description" ShortName="Short Description" PluralName="Short D
      <Hierarchy  Name="PRIMARY" LongName="Primary" ShortName="Primary" PluralName="Primary" Id="CHANNEL.PRIMARY.HIERARCH
        <HierarchyLevelAssociation  Name="TOTAL_CHANNEL" LongName="HLA1" ShortName="HLA1" PluralName="HLA1" Id="CHANNEL.P
          <AttributeProjection  Name="LONG_DESCRIPTION" LongName="LONG_DESCRIPTION" ShortName="LONG_DESCRIPTION" PluralNa
            <Attribute XMLIDref="CHANNEL.LONG_DESCRIPTION.ATTRIBUTE" />
            <Level XMLIDref="CHANNEL.TOTAL_CHANNEL.LEVEL" />
            <Hierarchy XMLIDref="CHANNEL.PRIMARY.HIERARCHY" />
          </AttributeProjection>
          <AttributeProjection  Name="SHORT_DESCRIPTION" LongName="SHORT_DESCRIPTION" ShortName="SHORT_DESCRIPTION" Plura
            <Attribute XMLIDref="CHANNEL.SHORT_DESCRIPTION.ATTRIBUTE" />
            <Level XMLIDref="CHANNEL.TOTAL_CHANNEL.LEVEL" />
            <Hierarchy XMLIDref="CHANNEL.PRIMARY.HIERARCHY" />
          </AttributeProjection>
          <SourceMappingGroup >
            <DimensionMapGroup  Name="MAPGROUP1" LongName="MAPGROUP1" ShortName="MAPGROUP1" PluralName="MAPGROUP1" Id="CH
              <DimensionKeySourceExpression  Name="KeyMap" LongName="KeyMap" ShortName="KeyMap" PluralName="KeyMap" Id="C
                <SourceColumn  Name="OBJ389992306" LongName="" ShortName="OBJ389992306" PluralName="OBJ389992306" Id="CHA
                <Level XMLIDref="CHANNEL.TOTAL_CHANNEL.LEVEL" />
                <Hierarchy XMLIDref="CHANNEL.PRIMARY.HIERARCHY" />
              </DimensionKeySourceExpression>
              <AttributeSourceExpression  Name="AttrMap1" LongName="AttrMap1" ShortName="AttrMap1" PluralName="AttrMap1"
                <SourceColumn  Name="OBJ389992308" LongName="" ShortName="OBJ389992308" PluralName="OBJ389992308" Id="CHA
                <TargetObject XMLIDref="CHANNEL.PRIMARY.TOTAL_CHANNEL.LONG_DESCRIPTION.ATTRIBUTEPROJECTION" />
                <Level XMLIDref="CHANNEL.TOTAL_CHANNEL.LEVEL" />
                <Hierarchy XMLIDref="CHANNEL.PRIMARY.HIERARCHY" />
                <BaseAttribute XMLIDref="CHANNEL.LONG_DESCRIPTION.ATTRIBUTE" />
              </AttributeSourceExpression>
              <AttributeSourceExpression  Name="AttrMap2" LongName="AttrMap2" ShortName="AttrMap2" PluralName="AttrMap2"
                <SourceColumn  Name="OBJ389992310" LongName="" ShortName="OBJ389992310" PluralName="OBJ389992310" Id="CHA
                <TargetObject XMLIDref="CHANNEL.PRIMARY.TOTAL_CHANNEL.SHORT_DESCRIPTION.ATTRIBUTEPROJECTION" />
                <Level XMLIDref="CHANNEL.TOTAL_CHANNEL.LEVEL" />
                <Hierarchy XMLIDref="CHANNEL.PRIMARY.HIERARCHY" />
                <BaseAttribute XMLIDref="CHANNEL.SHORT_DESCRIPTION.ATTRIBUTE" />
              </AttributeSourceExpression>
            </DimensionMapGroup>
```

Figure 2. Implementation view of channel-star of XML data.

In Figure 3, a wizard view of creating star of XML, shows the imported data of channel star schema from the XML schema as shown in Figure 2. This basically creates the dimensions of the channel such as CHANNRL, CUSTOMER, PRODUCT and TIME. The types of the dimensions are defined here by the user.

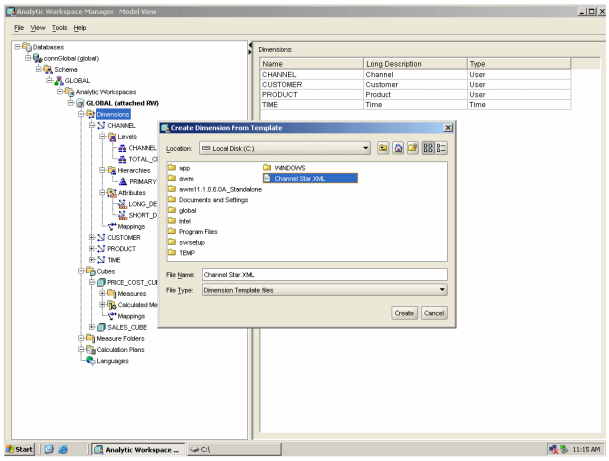


Figure 3. Creating Star of XML wizard view.

Likewise, levels, hierarchies and measures for PRODUCT, TIME, CHANNEL and CUSTOMER are created for generating the sales cubes and unite sales cubes. Figure 4, which is a schema utilization wizard view for OLAP customer cube construction, shows the mapping of the product, customer, time and channel. It reflects the schema allocation and utilization. Generally four types of schemas are supported, depending on the data allocation format, for instance, snow flake schema, star schema and others. Figure 4 shows a screenshot of star-schema mapping of Customer-Dimension. We mapped accordingly to the objects in customer dimension, by clicking on source object, and dragging it to the target object. After mapping all required objects, the user can click on the Apply button to proceed with its implementation or its application. The function of Table mapping view is available from the icon on the toolbar.

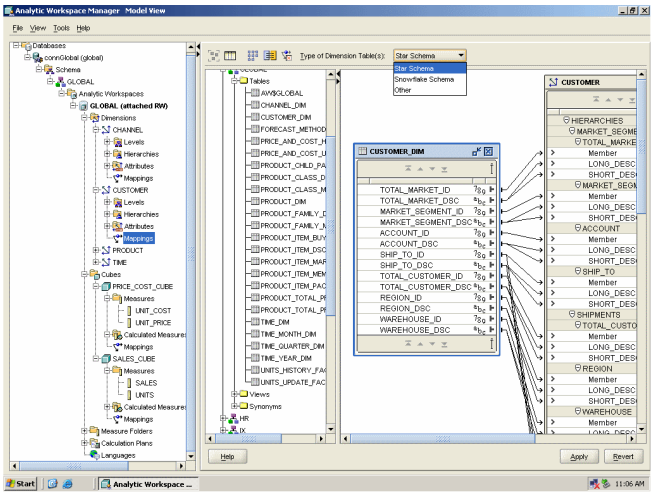


Figure 4. Schema Utilization Wizard View for OLAP Customer Cube Construction

Figure 5, view of Dimensions, measures and cubes of sales domain, shows the allocation of global view of sales domain for constructing cubes. In this window we show the DIMENSION, CUBES and MEASURES of the cubes of sales domain. DIMENSION window shows the dimensions of the sales cube named as product, time, customer and channel. Furthermore, a second bar in that window shows the cube formed named as PRICE-COST-CUBE and its dimensions named as PRICE-COST-CUBE and its dimensions are time and product. While the cubes of sale are SALE-CUBE and it has four main dimensions such as time, channel customer and product. In assimilation to the third bar provides us the information about measures and its cubes. Among them UNIT-COST measure has a cube of PRICE-COST-CUBE, UNIT-PRICE also has PRICE-COST-CUBE, SALES has its SALE-CUBE and at last for UNIT measure the cube is SALES-CUBE. So these are all the representations formed by the above Figures by extracting data from all aspects and by mapping those data through XML schema.

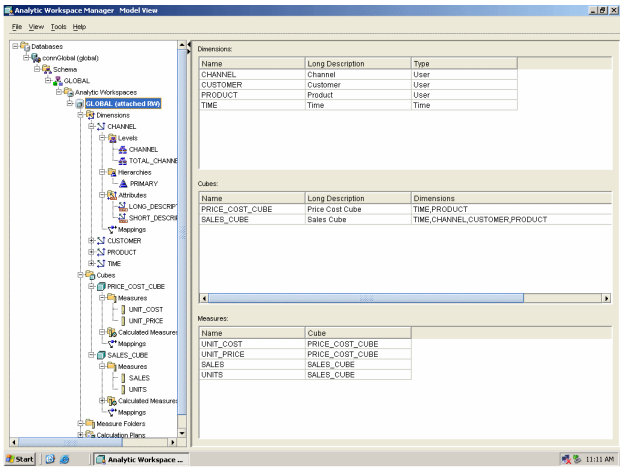


Figure 5. View of Dimensions measures and cubes of sales domain.

Figure 6, sales cube data view and graphical view, shows the mapping of XML logical schema onto graphs. It shows the unit cost and sale cube representation in various time scales. It enables a user to perform basic operation of cubes roll-up, drill-down, slicing, dicing and etc. Similarly, Figure 7, user interface view, shows the quarterly, yearly and monthly sales entries but in different forms of charts.

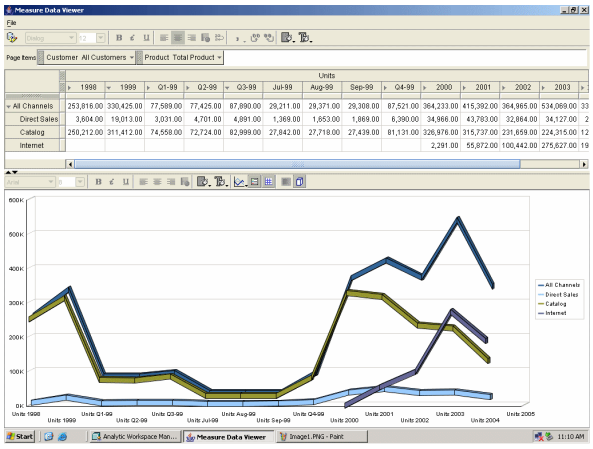


Figure 6. Sales cube data view and graphical representation.

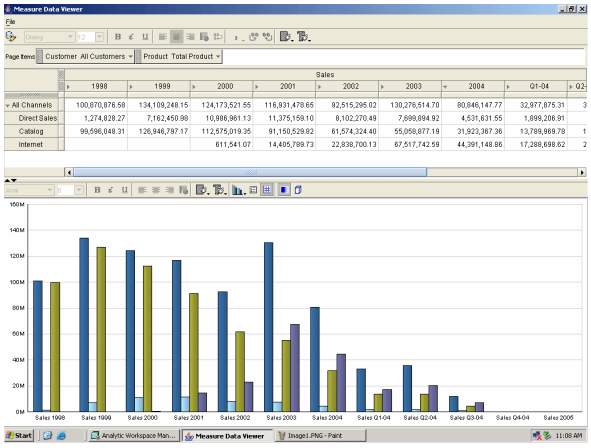


Figure 7. User Interface view and it graphical view.

B. *Implementation of Query Translator*

The module of the query translator is designed to support users with a simple interface and generate a cube of specific domain according to the will of the user. In assimilation to that in our approach we have developed a prototype by programming in C-Sharp framework version 2.0 for retrieving XOLAP cubes and fulfilling user query request. Two screenshots of the prototype system are shown to show the interfaces of the user's works. Figure 8, user interface view of the query translation, shows an interface which has been constructed for cube abstraction of different domains. There are three options over that form. Viewing of all cubes, viewing cube items details or a dynamically extraction of cubes and its dimensions. We have taken data set of sales domain and selected its dimensions. When we give a sales or HR or any other options in the given box, it will automatically allocate the dimensions. Furthermore, a user will also be able to dynamically allocate the dimensions of the specific domain. This interface allows users easy accessing of cubes of different domains by their choices.

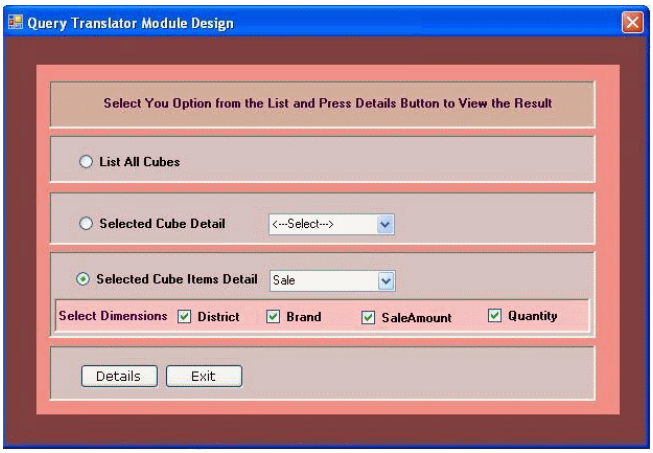
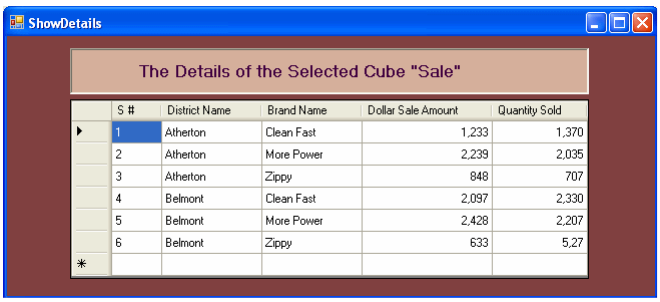


Figure 8. User Interface view for query translation.

Figure 9, real time entries view for the sale cube, shows the interface which represented the important details of selected cube. Given the source code in XML schema, the interface hides the complexities and details from user.

The algorithm of the cube generation is shown below. It reflects the extraction of the multidimensional cubes of different domain. The algorithm generalized the extraction of many diversified cubes and domains of different data set.



S #	District Name	Brand Name	Dollar Sale Amount	Quantity Sold
1	Atherton	Clean Fast	1,233	1,370
2	Atherton	More Power	2,239	2,035
3	Atherton	Zippy	848	707
4	Belmont	Clean Fast	2,097	2,330
5	Belmont	More Power	2,428	2,207
6	Belmont	Zippy	633	5,27

Figure 9. Real Time entries view for Sale Cube.

```
/* AIGORITHM: */
   using System.IO;
   using System.Xml;
   using System.Xml.XSL; // XML style sheet
   using System.Xml.Schema;

      Protected void form_Load()
      {
      // Declaration of Required Variable
      String[] CubeName                    = new    String[]; //Hold Cubes Name
      String[] DimensionsName = new    String[]; // Hold Dimensions Name
      // Show all cubes data
      If (rdoAllCubes.Checked == true)
      {
      showAllCubeDetails();
      }
      // Show single cube data
      Else If (rdoSinglecube.Checked = true)
      {
                CubeName  =  (cboCubesName.SelectedValue).ToString();
                        showSingleCubeData(CubeName);
      }
      // Show Some Dimensions of Cube

      Else If (rdoCubeItem.Selected == true)
      {
      CubeName = (cboCubeItems.SelectedValue).ToString();
      GenerateDimensions(CubeName);
      getDimensionsName();
      showSingleCubeData (CubeName);
      }
      } //Main Ends here
      // For Each Array Items extract the Cube name and pass on To  ExtractCubeData method to display
      Protected void showAllCubes()
      {
      For(int index=0;index<=cubes.Count;index++)
      {
      ExtractCubeData(Cubes[index]);
      DisplayData();
      }
      }
      /* Method to Reterive data for the single cube selected*/
      Protected void showSingleCubeData(:CubeName)
      {
      ExtractCubeData(:CubeName)
      DisplayData();
      }
      //Apply XML classes builtin in C# to Extract XML from the Array of Cubes
      //Apply XSLT to Retrieve XML Data in User Readable format
      Protected void  ExtratCubeData(:CubeName)
      {
      Return (Cube Data after applying XSLT);
      }

      /* Will return an Array containing the Dimensions Name, Being Selected by User */
      Protected void  GenerateDimensions(:CubeName)
      {
      Generate Check Boxes against each :CubeName Dynamically at runtime
      }
      /* Method to reterive the Dimensions of the selected cube*/
      Protected void  getDimensionsName(:CubeName)
      {
      //retrive DimensionName for :CubeName Cubes into Array
      }
```

V. CONCLUSION AND FUTURE WORK

We proposed a multilayered architecture of unified distributed XOLAP cube. It allows data from multi-sources such as from a data warehouse, flat files and that of relational tables to be converted to multidimensional cubes in a single format of XML. Consequently, due to uniform format the X-cubes can be integrated in to a single global cube without dealing with any issue of heterogeneity. In addition to that a number of query languages like MDX, SQL and X-QUERY can be used for generating different cubes of specific domains. Supporting components are implemented along with the proposed architecture. In this project we implemented a working prototype under this architecture. We converted the testing data to XML format and generated XML and OLAP cubes namely XOLAP cubes. These cubes are generated from the testing data that come from two basic domains: sales domain and disaster management domain. As an extension, we have implemented an X-query evaluator for handling OLAP query.

So far we have developed the prototype pertaining to the two layers: data layer and storage layer. It can generate multidimensional cubes of two testing domains. In the near future we plan to complete the implementation on the remaining two layers, query processing layer and client layers. Moreover, the query evaluator will use the Meta-Cube stored in storage layer containing the descriptions of X-Cubes and then distribute the OLAP query to appropriate cube for response. To implement the query processing layer is our next step. At this stage, the client layer supports OLAP queries only in XQuery format. The OLAP query will be extended to support other formats.

REFERENCES

[1] F. Wasil, W. Gray and N. Fiddian, "Establishing an XML Metadata Knowledge Base to Assist Integration of Structured and Semi-structured Databases," Proc. ACM 17th Australasian Database Conference, UK, Vol.170, pp.69-78, 2006

[2] O. Boussaid, R. B. Messaoud, R. Choquet and S. Anthoard, "X-Warehousing: An XML Based Approach for Warehousing Complex Data", pp.39-56, ADBIS, 2006

[3] J. Lu, C. Huang and T. Chuang,"Querying and Browsing XML and Relational Data Sources", SAC'02, March 11-15, Seoul, Korea, 2007

[4] D. Pedersen, K. Riis and T. B. Pedersen, "XML-Extended OLAP Querying", Proc. 14th Intl. Conf. on Scientific and Statistical Database Management, IEEE Computer Society, Washington, DC, USA, pp.195-206, 2002

[5] D. Theodoratos and T. Sellis, "Answering Multidimensional Querying on Cubes Using Other Cubes," Proc. the 12th IEEE International Conference on Scientific and Statistical Database Management, 26-28 July 2000, pp.109 – 122

[6] M. Golfarelli, S. Rizzi and B. Vrdolijak, "Data warehouse designing from XML sources", Proc. the 4th ACM intl. workshop on Data Warehousing and OLAP, Atlanta, Georgia, USA, pp-40-47, 2001

[7] D. Pedersen, K. Riis and T. B. Pedersen, "A Powerful and SQL – Compatible Data Model and Query Language for OLAP", Australasian Computer Society, Vol.5, 2001

[8] A. Pan, J. Raposo, M. Alvarez, P. Montoto, V. Orjales, J. Hidalgo, L. Ardao, A. Molano and A. Vina,"The Denodo Platform Integration" in proc. 28th IntI conf. on Very Large Data Bases Hong Kong, VLDB Endowment 2002

[9] C. Kit, T. Amagasa and H. Kitagawa, "A Development of an XML-OLAP System", pp.206-212, 2002

[10] T. S. C. Frank and C. Chen, "Integrating heterogeneous data warehouse using XML technologies", Journal of Information Sciences, Vol.31, pp. 209-229, Jan, 2005

[11] T. Niemi, J. Nummenmaa and P. Thanisch, "Constructing OLAP Cubes Based on Queries," ACM, New York, USA, pp.9-15, 2001

[12] J. Pokorny, "Modeling Stars Using Schema", ACM, Atlanta, Georgia, USA, pp.24-31, 2001

[13] J. Pokarny, "XML Data Warehouse: Modeling and Querying", Proc. Databases and Information System II, pp. 67-80, 2002

[14] C. Chen, G. Miliha, S. Padmanabhan and I. Rouvellou, "Query Translation Schema for Hecterogenous XML Data Sources", ACM, Bremen, Germany, November 5, 2005