# A Model-based Software Architecture for XML Data and Metadata Integration in Data Warehouse Systems

Wan Mohd Haffiz Mohd Nasir, Shamsul Sahibuddin
*Faculty of Computer Science and Information System, Technology University of Malaysia*
*E-mail: wmh2020@hotmail.com, shamsul@fsksm.utm.my*

**ABSTRACT**

*The demand for data integration is rapidly becoming larger as more and more information sources appear in modern enterprises. Extensible Markup Language is fast becoming the new standard for data representation and exchange on the World Wide Web, e.g., in B2B e-commerce, making it necessary for data analysis tools to handle XML data as well as traditional data formats. This paper presents architecture for XML-based data and metadata integration in Data Warehouse System for constructing an OLAP cubes. We are using a Common Warehouse Metamodel (CWM) for metadata interchange that incorporates a common shared metamodel to agree on metadata syntax and semantics.*

***Keywords****: XML, Data Warehouse, CWM, OLAP.*

## 1.0 INTRODUCTION

On-Line Analytical Processing (OLAP) is a category of business software tools that enables decision support based on multidimensional analysis of data. OLAP data, typically drawn from a physical integration of transactional databases, is organized in *multidimensional data models*, categorizing data as either measurable *facts* (measures) or hierarchically organized *dimensions* characterizing the facts. Features like automatic aggregation (Rafanelli, 1990) and visual querying (Thomsen, 1997) supported by OLAP tools ease the process of decision support compared to traditional DBMSs (Lenz, 1997).

Integration of distributed data sources is becoming increasingly important as more business relevant data appear on the web, e.g., on B2B marketplaces, and enterprises cooperate more tightly with their partners, creating a need for integrating the information from several enterprises. The *data warehousing* approach dictates a physical integration of data, mapping data from different information sources into a common multidimensional database schema. This enables fast evaluation of complex queries, but demands great effort in keeping the data warehouse up to date, e.g. when data passes from the sources of the application-oriented operational environment to the data warehouse, inconsistencies and redundancies must be resolved, so the data warehouse provides an integrated and reconciled view of the data of the organization (Jensen et al., 2001a).

XML is a meta language used to describe the structure and content of documents. XML, although originally a document markup language, is increasingly used for data exchange on the Web. The application of XML as a standard exchange format for data available on the Web makes it attractive to use in conjunction with OLAP tools. Previous approaches for integrating web-based data, particularly in XML format, have focused almost exclusively on data integration at the *logical* level of the data model, creating a need for techniques that are usable at the conceptual level which is more suitable for use by system designers and end users. The most wide-spread conceptual model is the Unified Modeling Language (UML) (OMG, 2001a).

Since the data warehouse is storing historical data and new data sources are added from time to time, the amount of data in the warehouse is growing permanently. This can lead to difficulties for the user with making use of the data if no appropriate instrument for supporting the navigation through the warehouse is made available (Auth and Eitil, 2002).

In current data warehouse environments there is either no or only insufficient support for a consistent and comprehensive metadata management. Typically, a multitude of largely autonomous and heterogeneously organized repositories coexist. Do and Rahm (2000) categorize the major metadata types and their interdependencies within a three-dimensional classification approach and then investigated how interoperability and integration of metadata can be achieved based on a federated metadata architecture and standardization efforts such as OIM and CWM. They also examined synchronization alternatives to keep replicated metadata consistent and gave an overview of currently available commercial repositories and discussed interoperability issues to couple data warehouses with information portals.

The need for data movement and data integration solutions is driven by the fact that data is everywhere underneath business applications. The same applies for metadata: metadata is also everywhere underneath the data and object modeling tools, as well as within the repositories of the ETL, Data Warehouse, Enterprise Application Integration, and Business Intelligence development tools. An adequate metadata management should be focused on storing all the metadata in one central repository to avoid redundancy and keep the metadata consistent. In this paper, we propose architecture for XML-based data

and metadata integration in data warehousing system with Common Warehouse Metamodel (CWM) as a standard for modeling and exchanging metadata.

## 2.0 METADATA

Metadata is becoming more and more important in areas like data warehousing, knowledge management, enterprise application integration, and e-business (Agosta, 2001). Although the concept of metadata is not new at all there is still no single, accepted definition but the little useful 'metadata is data about data'. In order to have a sound foundation for further argumentation, we will provide a definition of metadata in the context of data warehousing.

In order to understand metadata properly one has to understand the process of abstraction, which is executed to create metadata. The prefix 'meta' (Greek for 'beyond') indicates a change from ordinary data to data that resides on a higher level of perception. In order to distinguish data from metadata we introduce the term 'objectdata'. Objectdata represents objects and their relationships in a certain domain. The term is motivated by the object paradigm that aims at "modelling the real world as close to a user's perspective as possible" (Koshafian and Abnous, 1995). Objectdata can describe objects of the real world as well as objects of higher levels of abstraction in an abstraction hierarchy like OMG's Meta Object Facility (MOF). An example for objectdata is the requested items, the amount of each item, or the customer's address of an electronically stored order. The structure and semantics of the order representation is described by metadata e. g. a record or table definition or an XML Document Type Definition (DTD) (Auth and Eitil, 2002).

For the context of data warehousing we define metadata as data that answers questions about all the objectdata in a data warehouse, transformations of objectdata and underlying data flows, and finally the technical and conceptual system architecture. Referring to this definition, with the help of metadata a user should be able to locate the proper objectdata for this task as well as understand and interpret usage, meaning, sources, creation, structure, quality, and topically of the objectdata he is dealing with (Auth and Eitil, 2002).

In a DW system almost every software component produces and consumes metadata. For example the system tables of the central warehouse database store the description of the warehouse data model which is actually metadata. To make further use of this metadata a software structure for this purpose, containing interfaces, and data store and access components must be implemented. Furthermore, metadata is consumed and produces by the whole range of the data warehouse users starting from developers to end users (Auth and Eitil, 2002).

## 3.0 THE COMMON WAREHOUSE METAMODEL FOR METADATA INTERCHANGE

The Common Warehouse Metamodel is a standard for describing technical and business metadata occurring from data warehousing and business intelligence. CWM is hosted by industry consortium Object Management Group (OMG) (OMG, 2001a). Although the main purpose of CWM is designed for metadata interchange between different tools and repositories it can be used also for building active objects models for storing and maintaining metadata (Poole, 2000). CWM is founded on the UML metamodel and extends it with specific meta-classes and meta-relationships for modelling data lineages found in the warehousing domain.

Thus, it provides a complete specification of syntax and semantics necessary for interchanging shared metadata. CWM consists of:

1. A standard language for defining the structure and semantics of metadata in a semi-formal way (Meta-Object Facility (MOF) and Unified Modelling Language (UML)).
2. A standard interchange mechanism for sharing metadata defined in the standard language (eXtensible Markup Language (XML) and XML Metadata Interchange (XMI)).
3. A standard specification (interface) for access to, and discovery of, the metadata defined in the standard language (CORBA Interface Definition Language (IDL)).

CWM claims on comprising the whole DW life cycle including establishment, build, operation, and maintenance phase. Therefore, it is targeted at six categories of users (OMG, 2001b): warehouse platform and tool vendors, professional service providers, warehouse developers, warehouse administrators, end users, and information technology managers. CWM was built on the foundation of UML 1.3 which serves three different purposes (OMG, 2001b):

1. UML is the modelling language for defining CWM. CWM is mostly expressed with help of class diagrams supported by Object Constraint Language (OCL).
2. UML is the underlying metamodel from which CWM packages inherit several classes and relationships.
3. UML Foundation package works as CWM package for object-oriented data sources, which means UML itself is part of CWM.

### 3.1 CWM Structure

In order to reduce complexity and enhance the ease of understandability CWM is structured into 18 packages. Each package is designed to be mostly independent of the others to support reuse characteristics. Furthermore, this allows selecting only those packages for implementation those are appropriate for a specific data warehouse environment. Figure 4 shows the 18 packages which are grouped into four layers. All parts of the CWM related to UML are coloured with dark-grey.

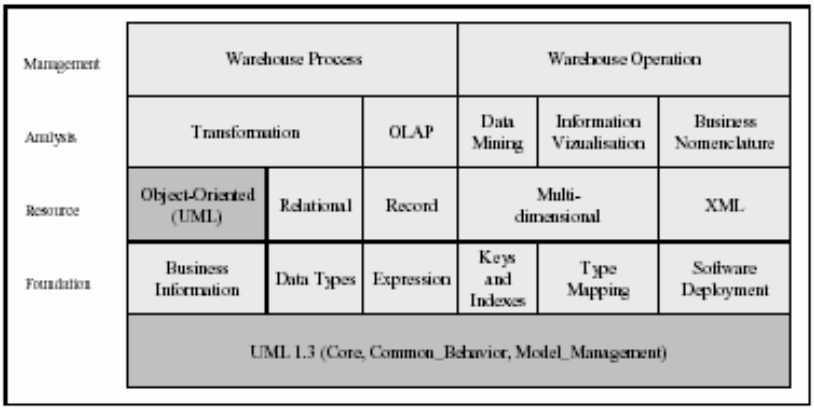


**Figure 1:** Common Warehouse Metamodel (OMG, 2001b)

The four layers of the CWM structure the packages according to specific modeling domains (OMG, 2001b):

**Foundation**

Basic modeling elements and concepts that are shared between several packages throughout CWM can be found in the foundation layer. Generic elements for both technical metadata e. g. Data Types, Expression, and Keys and Indexes packages and business metadata e.g. Business Information package.

**Data Resource**

Packages for defining all kind of data structures, both data sources and targets for warehouse processes. There are packages for object-oriented, relational, record, multidimensional and XML data structures.

**Data Analysis**

The analysis layer provides packages for describing metadata that is used for transforming and analyzing objectdata. The generic term transforming covers all kind of extracting, transforming, and loading processes (ETL processes). Besides information about data sources and targets also data flows and data lineage can be handled. For analyzing data there are packages pertaining OLAP, data mining, and information visualization.

Furthermore, the Business Nomenclature package is located in this layer. Business Nomenclature is the main package for modeling business metadata. It comprises concepts for describing taxonomies and glossaries.

**Warehouse Management**

Consists of Warehouse Process and Warehouse Operation packages merely. Both packages pertain to warehouse operation and maintenance. The Warehouse Process package offers elements for describing transformation processes utilizing events and triggers. The Warehouse Operation packages relate to transformation execution, performance measurement, and system updates.

## 4.0 META INTEGRATION TECHNOLOGY

Meta Integration Technology, Inc. is a Silicon Valley, California based software vendor specialized in tools for the integration and management of metadata across tools from multiple vendors, and multiple purposes including data and object modeling tools, data Extraction, Transformation, and Load (ETL) tools, Business Intelligence (BI) tools, and so on. The need for data movement and data integration solutions is driven by the fact that data is everywhere underneath business applications. The same applies for metadata: metadata is also everywhere underneath the data and object modeling tools, as well as within the repositories of the ETL, Data Warehouse, Enterprise Application Integration, and Business Intelligence development tools (Bremeau, 2001).

### 4.1 Meta Integration Works (MIW)

MIW is a complete metadata management solution with sophisticated functionalities such as the Model Browser, the Model Bridges, the Model Comparator, the Model Integrator, and the Model Mapper all integrated around a powerful metadata version and configuration management as shown in Figure 2.

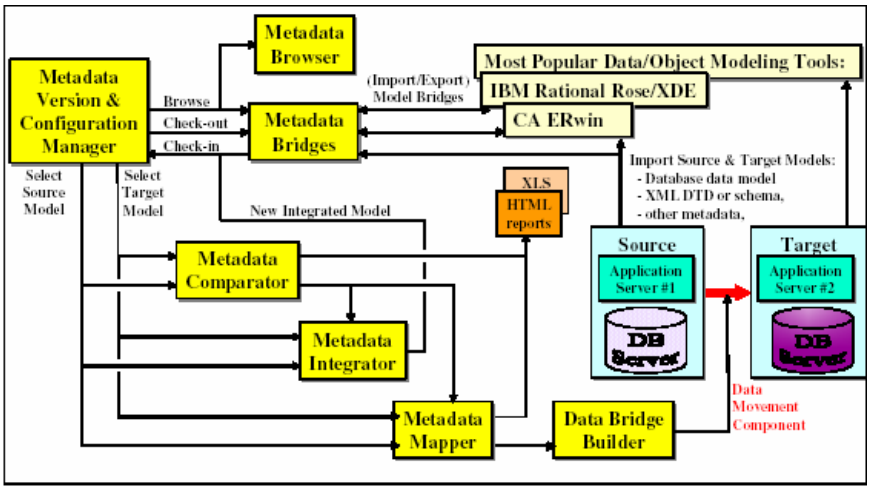


**Figure 2:** Meta Integration functionality (Bremeau, 2001)

MIW is a powerful metadata management solution, and integrates well with today's best practices in software development, as it provides a unique component based approach to the ETL tool market. Indeed, the MIW development environment generates C++ based data movement components that can be easily integrated (plug and play) with any Windows® or UNIX based business applications. Multiple data movement components can be produced for various purposes such as:

- Legacy Data Migration (LDM)
- Enterprise Application Integration (EAI)
- Data Warehousing (DW) and data marts.

The code of the produced data movement components can be reviewed through any Quality Assurance (QA) processes, and does not depend on any middle ware (free of any run-time cost at deployment time). The Model Mapper provides the mapping migrations required to support the perpetual changes in the source and destination data stores. Indeed, one of the key features of MIW is the built-in support for change management facilitating the maintenance and/or generation of new versions of the data movement components as needed. Data Connectors are available for most popular databases via ODBC (as DB2), as well as for XML data sources (as HL7 for Health Care) to service the expanding needs in the fields of EDI, e-business, and enterprise information portals. MIW is entirely written in Java, and can be connected to a local or centralized metadata repository.

### 4.2 Meta Integration Repository (MIR)

MIR is based on a modern 3-tier architecture as shown in Figure 2-3 with support for multi-users, security, and concurrency control. The repository metamodel integrates standards like the OMG CWM and UML, and supports XMI compliant metadata interchange. MIR can manage massive amounts of metadata and make it persistent on most popular RDBMS like DB2, Oracle or SQL Server. The underlying repository database is fully open allowing users to build their own metadata Web portals, or use their existing data tools to perform metadata reporting, mining, and even intelligence (Bremeau, 2001).

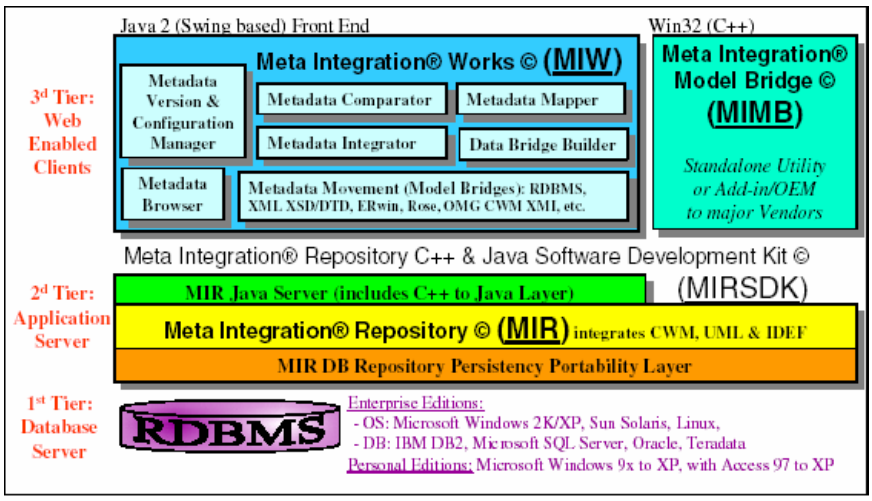


**Figure 3:** Meta Integration architecture (Bremeau, 2001)

### 4.3 Meta Integration Model Bridge (MIMB)

MIMB is a utility for legacy model migration and metadata integration. MIMB also operates as an add-in integrated inside popular modeling, ETL, and BI tools. With over 40 bridges, MIMB is the most complete metadata movement solution on the market. MIMB supports most popular standards and the market leading tool vendors, as illustrated in Figure 4:

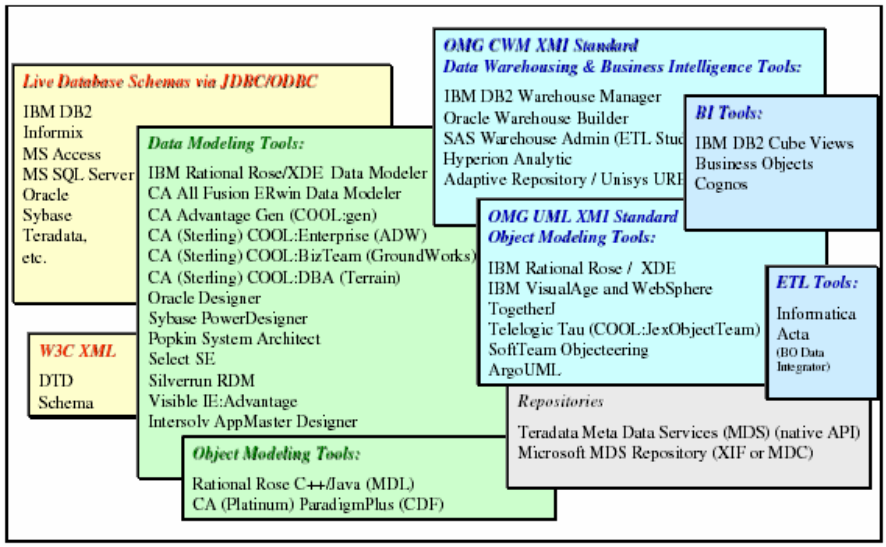


**Figure 4:** Meta Integration supported tools (Bremeau, 2001)

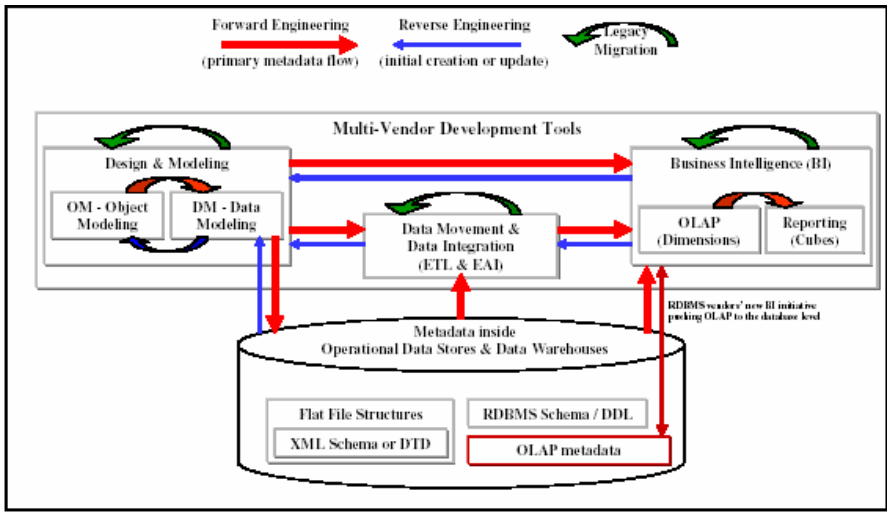


**Figure 5:** Business cases for metadata movement solutions (Bremeau, 2001)

### 4.4 Metadata flow scenarios

The generic metadata flows can be summarized as in Figure 5.

The tools vendors themselves can provide some of these metadata movements; for example, IBM Rational® Rose® provides bi-directional integration between UML object modeling and physical data modeling. Similarly, BI vendors provide the forward engineering from their OLAP dimension design tool to their OLAP based reporting tool. However, large corporations use best-of-breed tools from many vendors. In such case, MIMB can play a key role implementing all the metadata movement required for the integration of their development tools.

## 5.0 GENERAL ARCHITECTURE FOR XML DATA AND METADATA INTEGRATION

There are several types of data sources that need to be integrated such as relational data and XML data. In this paper we will focus on the way XML data can be integrated and propose a solution for XML data movement from XML DTD to the data warehouse and Common Warehouse Metamodel (CWM). Previous research did not concentrate on how to integrate metadata that can not be extracted from

DWH components but resides in various other sources such as from e-business application. This research focused on how to integrate XML documents and XML DTD as a metadata in the data warehouse. Figure 6 illustrates the general architecture for XML data integration and the metadata integration.

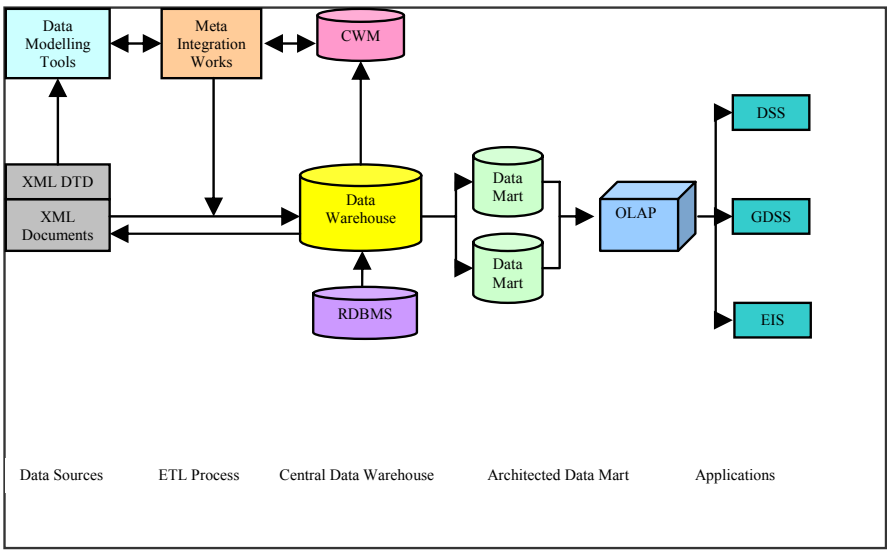


**Figure 6:** Architecture for XML Data and Metadata Integration

The detailed explanation for XML data and Metadata integration are described in section 5.1. Our architecture used a central metadata repository that implements a common metamodel based on OMG's CWM and serves as a hub for metadata interchange. Metadata and metamodels are exchanged between CMW repository and local metadata from data warehouse stores utilizing XMI as a standard interchange. Meta Integration Model Bridge (MIMB) will be use for XML metadata integration from XML DTD to the data modelling tool and to the CWM. Meta Integration Works (MIW) will be use for Extract, Transform and Load (ETL) process between the data sources and the data warehouse.

Data mart will be produces from data warehouse to get on the subject-specific information before it could be send to the OLAP tools for produce OLAP cubes. Finally, end user application such as DSS and EIS could be developed from the OLAP cubes for the end user make the query, reporting, drill-down, and analysis.

### 5.1 XML Data Movement

This section will describe how to generate data migration and metadata integration that deal with XML data sources.

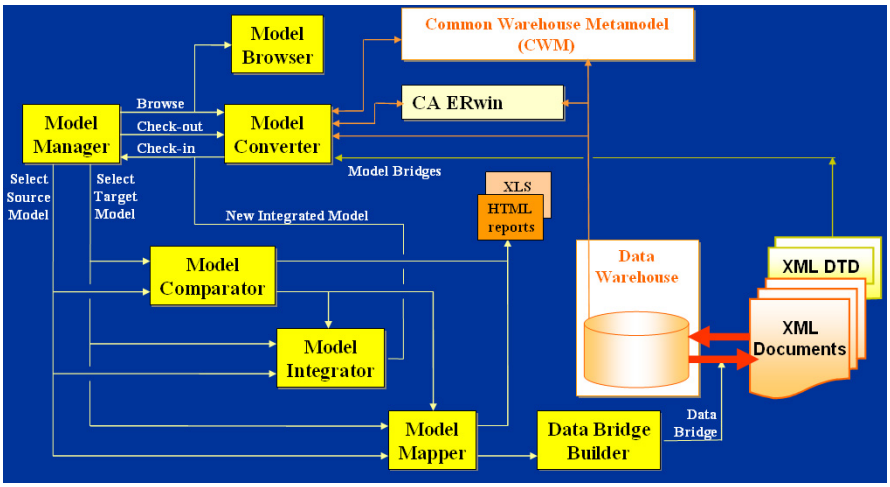


**Figure 7:** XML Data Movement Components for Data Warehouse and CWM.

Figure 7 shows more detailed architecture about XML data and metadata integration in the data warehouse by using Meta Integration Works as a tool for model conversion, model mapping, and code generation for XML data movement. We consider that the DTD (Document Type Definition) file is the

model of the XML file involved in the data movement because it describes the structure of the content of the XML file. The main idea is that we will load the DTDs in MIW as models. XML Elements will be represented as Classes, XML Attributes as attributes and PK/FK relationships will be created to represent the tree-like structure (parent and child elements) of the XML file. XML DTD will be imported to the model converter and will be viewed as a DTD model.

CWM XMI file from data warehouse will be imported by reverse engineering to the data modelling tools to integrate with metadata from XML DTD. The integrated model will be export to the CWM by using forward engineering. All metadata import and export process are using Meta Integration Model Bridge as a tool. After the mapping, we can generate the code by start the Data Bridge Builder. The XML documents could be integrated in the data warehouse after this operation. New integrated model that integrate metadata from CWM and XML DTD will be store in the CWM by using forward engineering.

## 6.0 CONCLUSIONS AND FURTHER WORK

Motivated by the increasing use of OLAP systems for analyzing business data and XML documents for exchanging information on the Internet, this paper proposed an architecture that enables OLAP systems to exploit XML data sources. This paper also looks at the metadata management aspect by using a central metadata repository that implements a common shared metamodel based on OMG's Common Warehouse Metamodel. CWM provides a standard language for defining the structure and semantics of sharing metadata.

The implementation of an architecture utilizing the aspects describing in this paper is the next immediate step to be taken. Future research will more focus on the XML data movement solution and metadata interchange in the CWM repository.

## REFERENCES


Agosta, L. (2001). Reports of the demise of meta data are premature. DM Review 3, URL: http:// www.dmreview.com/master.cfm. 12.12.2003.

Auth, G., and Eitil, V.,M. (2002). A Software Architecture for XML-based Metadata Interchange in Data Warehouse Systems.

Bremeau C. (2001). XML Data Movement Components for Teradata URL: www.metaintegration.net/ Partners/NCR-Partners2001-MetaIntegration.ppt

Do, H. H., Rahm, E., (2000). On metadata interoperability in data warehouses. Technical Report 1- 2000, Institute Informatics, University Leipzig.

Jensen, M. R., Møller, T.H., and Pedersen, T.B. (2001a). Specifying OLAP Cubes On XML Data. *Tech Report R-01-5003,* Department Of Computer Science, Aalborg University.

Jensen, M. R., and T. H. Møller (2001b). Constructing OLAP Cubes From XML Data. *Tech Report R-02-5003,* Department Of Computer Science, Aalborg University.

Koshafian, S., Abnous, R. (1995). Object Orientation – Concepts, Analysis, and Design, Languages, Databases, Graphical User Interfaces, Standard. 2nd ed. Wiley.

Lenz, H., (1997). Summarizability in OLAP and Statistical Databases, *Proceedings of the Ninth International Conference on Statistical and Scientific Database Management*, pp. 39-48.

Mimno, P. (2002). Succesful Real-Time Business Analytics: A Data Warehousing Strategy. White Paper. Informatica Carporation.

OMG (2001a) OMG: OMG Specifications. URL: http://www.omg.org/technology/documents/specifications.htm Last visited on 22/09/2003.

OMG (2001b) OMG: Common Warehouse Metamodel (CWM) Specification. URL:http://www.omg.org/cgi-bin/doc?ad/2001-02-01. Last visited on 28/09/2003.

Pledge, K., McGarry, J. (2001). Data Warehousing for Actuaries. Versi (1.2): 3-12.

Poole, J. (2000). The Common Warehouse Metamodel as a Foundation for Active Object Models in the Data Warehouse Environment. Position paper to ECOOP 2000 workshop on Metadata and Active Object-Model Pattern Mining – Cannes, France.

Rafanelli, M. (1990). STORM: A Statistical Object Representation Model, *Proceedings of the Fifth Conference on Statistical and Scientific Database Management*, pp. 14-29.

Shukla, A. (1996).Storage Estimation for Multidimensional Aggregates in the Presence of Hierarchies, *Proceedings of Very Large Databases*, pp. 522-531.

Thomsen, E., (1997). *OLAP Solutions: Building Multidimensional Information Systems*, John Wiley & Sons, Inc.