IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4855

Toward XML-Based Data Warehouse Architecture

Rami Rifaieh
R&D-TESSI INFORMATIQUE
72 bis Rue Bergson, 42000 Saint-Etienne,France
Email: rrifaieh@tess2i.fr

Nabila Aïcha Benkat
Department of Informatics, LIRIS-INSA de Lyon
69621 Villeurbanne, France
Email: nabila.benharkat@if.insa-lyon.fr

ABSTRACT

XML enables data to migrate from relational databases and other sources into future applications. It integrates structured and unstructured data to present new application and knowledge management opportunities. By another way, data warehousing is an essential element of decision support, which has increasingly become a focus of the database industry. Meta-data contains data dictionary and repository; it describes warehousing process, data storage and information delivery. This paper defines how XML affects and changes the concept of the data warehouse. It describes an XML based tool for a structured, reusable and more efficient data warehouse (DW). Otherwise, it shows the ability to specify the warehousing tools with rules defined in the meta-data dictionary.

1 INTRODUCTION

Data warehouse system is a collection of data used for decision-making. The success of data warehouses implementation for business intelligence activities implies an increasing demand for new concepts and solutions [2]. It includes growth across platforms, tools, and applications. In all cases, the integration of data from heterogeneous sources is the essential stage of warehousing process. These sources could arise from existing legacy systems, which continue to generate data from mainframes computers through client server architecture. Moreover, disparate Web applications such as data sharing, real time data interchange, e-business, B2B activities and systems as ERP provide information that may be used to populate the company data warehouse. Then, data warehouse should be improved to support new applications such as real-time data warehouse and data web-house.

Furthermore, meta-data is an essential information that defines the what, where, how, and why about the used data. It can range from the conceptual overview of the real world to detailed physical specifications for the particular database management system. Meta-data can be used by automated tools (e.g. indexing robots) to improve the data interpretation / exploitation. Specifically, meta-data of business warehouse contains management rules [2]. They define which elements are supposed to be decisional and how they are calculated.

XML enables data to migrate from relational databases and other sources into new applications. It includes the ability to exchange data between application programs and browsers, between application programs and other application programs etc.

The purpose of this paper is to show how XML and its standards support data warehousing process. Therefore, XML could be used in all *the construction process*, especially in *legacy data extraction*, *input transaction capture*, *cleansing procedures*, *direct storage of XML*, and *front-end information delivery*. As we also deal with the evolution of meta-data, the XSLT component associated with XML documents helps us to support progressive meta-data for management rules. XML schema can provide another interesting dimension to XML text by defining datatypes. Moreover, an automatic generation of XSLT transformation could be a solution for personalization process based on meta-data description.

We will focus on showing where and how the extensible markup language can reduce process complexity. We try to show how using XSLT makes it easier to acquire and transform data. We discuss also the automatic generation of XSLT from progressive meta-data, and show the usefulness of XML schema in the loading process.

The paper is organized as follows. In the next section, we will glance over DW and its construction process. The section 3 will detail the advantages of using XML for DWs. The section 4 will describe the design of XETL (XML-based Extraction, Transformation and Load tool). We will conclude our work in the section 5 with the future perspectives.

2 THE DATA WAREHOUSE SYSTEM

Data warehousing constitutes the background to enable business intelligence solution, which lets organizations access, analyze, and share information internally with employees and externally with customers, suppliers, and partners.

According to standard data warehouse architecture, the data warehouses systems include:

- ETL or warehousing tools: a set of tools which are responsible of preparing the data constituting the Warehouse database;
- Restitution tools: the diverse tools, which help the analysts to make their business decisions, and;
- Meta-data: it brings together the data about all the components inside the DW.

2.1 ETL Tools

ETL (Extraction, Transformation, and Load) represents the warehousing tools, or population tools. Warehousing tools have challenge to provide maintenance capability, availability, task management, and evolution support. Data integration and reuse possibilities are wide open but not yet very well realized. Although, some tools provide reused functions, these solutions still limited. Indeed, existing functions do not allow users to utilize an existing transformation plan and specify it with parameters to create a new data warehouse.

Maintenance process is not better; for example, if a user wants to change a SKU (stock keeping unit) number definition from five digits to seven. How many programs need to be changed to affect this enhancement? For the most of existing tools, in order to enable this operation a query has to be formulated into the data dictionary. Then, the user has to update all the concerning programs. In the population process, our suggestion is to combine the ETL tools with XML, since our concern is first of all the diversity of data sources, data targets.

2.2 Restitution Process

Often the information in data warehouses is published to a company's intranet web site. HTML is actually used to build these sites. Thus, restitution tools should provide the ability to generate HTML pages from warehouse database.

Nevertheless, delivering operation results from restitution model to mobile portals is going to be a new feature for restitution tools. Wireless devices (Pocket and mobile phone) will be able to capture information and use it in a real time with geographically distant warehouse. Thus, new standard output format should be used such as WML (Wireless Markup Language).

2.3 Meta-data Management

Traditionally, meta-data is data about the data. The meta-data allows the various function areas in the warehouse to communicate. It encompasses all corporate resources: database catalogs, data dictionary, and data models [9]. In the common data architecture, data catalogs are needed to understand data sources. In general, the meta-data should cover the acquisition, access, and distribution of warehouse data and should be the key to provide the business user with a complete map of the data warehouse.

3 XML AND DATA WAREHOUSE SYSTEM

For information systems, XML includes the ability to exchange data between application programs and browsers, between application programs and other application programs etc.

3.1 How Can XML Improve Data Warehousing

XML can contribute in warehousing process: data integration, cleansing procedures, data storage, and front-end information delivery.

3.1.1 Source Data Integration

The integration of data from different sources is the essential stage of warehousing process. These sources could arise from existing legacy systems, which continue to generate data from mainframe computers through client server architecture. Moreover, disparate Web applications such as data sharing, real time data interchange, e-business, B2B activities and other systems such as ERP provide information that should be used to populate the company data warehouse.

If the systems submit data in XML format, then partners sharing a common XML schema can transmit and receive data. Although older legacy applications may need to be retrofitted with XML writers, many more modern systems are equipped to write data in XML format. Existing relational databases already support query output directly in XML form.

The ability to read remote data via XML greatly simplifies data extraction, because custom parsers for the remote data source do not have to be written. This, in turn, promotes a more distributed and opportunistic approach to a spread-out "data web-house". Data web-house is a data warehouse where sources data come from XML web documents; it constitutes an essential search engine for a collection of web data.

3.1.2 Native XML Data Storage

Native XML database [7] can be used to store the core of XML-based data warehouse. They provide the possibility to integrate semi-structured data inside the data warehouse. This direct XML storage can fulfil the needs of storage support for data web-house. In this category, we can identify *Tamino (*http://www.softwareag.com/tamino/), *Natix (*http://www.dataexmachina.de/natix.html), *and Lore (*http://www-db.stanford.edu/lore/). Hybrid scenarios in which data in XML form could be joined inside the database directly to data in conventional relational tables is also interesting. Indeed, market DBMS offer the capability to bulk load data inside relational table from XML sources. This step is done with the help of XML schema. Furthermore, an XML data type in a relational system [4] would open up completely new classes of hierarchical applications.

3.1.3 Front-end Information Delivery

More and more sophisticated tools are used to deliver information outside the data warehouse. The widespread deployment of XML will be helpful in removing query and reporting tools from end users' terminals. An XML data transfer plus an associated XSLT formatting specification is enough to produce any desired user interface presentation on a remote browser. The result business information could be published with HTML pages in the company's intranet web site.

Moreover, different end-user clients can use different XSLT formatting specifications while drawing data from the same database server. All that the server needs to do is expose its query results through XML and from the client side just a Web browser is needed.

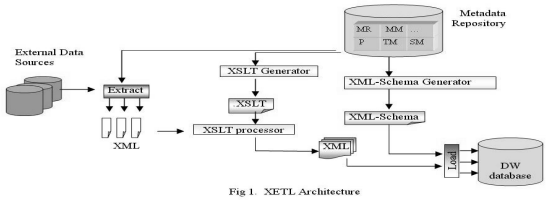

Fig 1. XETL Architecture

3.2 Advanced Features

3.2.1 Efficiency

XML is a neutral format, which can be used to enable the communication between different warehousing tools. Indeed, no owner format is needed to link and merge different warehousing tools, which is useful to divide the process and use a tool for extraction another one for transformation etc. The efficiency of the global process will be higher, especially when tools have good performances within different steps. We can chain a high extraction tool with XML generation capability and a high transformation tool with XML input, etc. Since XML is used in all the tools, no extra transformation between different owner tools formats is required.

3.2.2 Maintenance

The business rules of an organization are never the same, due to changes in the real world. So far, how do we maintain consistency when business rules change as a result of corporate reorganizations, regulatory changes, or other changes in business practices? How many places are impacted for each of these potential changes?

Due to the evolution of management rules, many programs need to be modified to enable new decision elements or new formulae for restitution. These rules define the decisional elements, equations, and parameters. For the most of existing tools, in order to realize this operation a query has to be formulated into the meta-data. Then, the user has to update all the concerning programs. To make maintenance process work better, the warehousing process has to be updated automatically. XML presents the ability to specify the transformations using XSLT. If the system is able generate automatically a new transformation, no extra update will be needed to enable the evolution.

3.2.3 Scalability

An important aspect to achieve the scalability is that changes in the implementation on either the application system side or the data side are not changing the transported data format between the two applications. This means that the client application or the server application does not need to be aware of that change.

One way of achieving that is to move the data in the standardized format between these components. Currently, the main format that is used in all the scenarios is XML. The idea is to take the data, in the data system or in the application system, move it into the standardized format of XML, both on the wire as well as using standardized grammar, which you can express in XML, to warp the data and transport it between the different systems.

3.2.4 Interoperability

Corporations are interacting on the web, blending suppliers, partners, and customers to form a virtual enterprise that function as the superset of the physical organizations. This e-business is performed through a myriad of real time information exchange technologies such as electronic document interchange (EDI), business-to-business exchange and e-business server applications.

Establish interoperability between these different sources is needed to integrate data inside DW. XML can play a major role in the integration of web sources inside the business DW, from corporate information portals, product catalogs, parts databases, and business-to-business document exchange etc.

3.3 How Meta-data Improve Data Transformation

For actual data warehouse architecture, few of tools communicate directly with meta-data repository. Meta-data are used to answer administrator queries. These queries offer information concerning the structure, models, and the warehousing process. Meanwhile, the set of meta-data is passive and query limited.

The existing tools do not deal with the evolution of management rules and reusability. Indeed, by one way, meta-data of business warehouse contains management rules. A passive use of meta-data implies two sorts of updates: meta-data repository and programs that handle data into warehouse. By another way, ERP systems [3] store data of different applications by a similar way on mainframe computers. In this case, data dictionary should provide more than just data attribute descriptions, range value, valid value, etc. It should supply the personalization process by applying the needed mapping to achieve the construction of different data warehouse. Thus, the same data element might be used by different entities of different applications to mean different things. Different data element names could also be used to represent the same things, potentially creating hundreds of instances of the same data all inconsistently named. In this case, data dictionary should provide more than just data attribute descriptions, range value, valid value, etc. It should supply the personalization process by applying the needed mapping to achieve the construction of different data warehouse. A generic plan is needed to establish mapping between source and target data. If meta-data contains all the generic plans for a target application, a personalization process can be used to extract data from source and apply the needed mapping.

The idea of making meta-data being ***active*** can improve warehouse systems. The solution can be performed with traditional transformation query with an automatic query generator or it can be a specific transformation language as used in XETL tool.

4 PRESENTATION OF XETL TOOL

In this section, we present XETL, an **X**ML based **E**xtraction, **T**ransformation and **L**oad tool (Fig.1). The basic idea of our tool consists of using the XML format to create new generation of data warehouse, where XML is used as a pivot format to perform the warehousing process. A full case study of using XETL with real scale commercial data is described in [12]. This study shows the effectiveness of using XETL as a warehousing tool to populate a relational database.

4.1 XETL and Existing Tools

XETL takes advantage of XML ability to realize interoperability, scalability, maintenance, efficiency, and data integration opportunity. We consider that data transformation process is generated directly from meta-data. Thus, the designer will not be called to verify the consistency of the process, since this process uses parameters extracted from meta-data repository. At the same time, traditional request over meta-data is accessible for system's administrator. As we are dealing with XML format, we have tried to perform this functionality by generating XSLT transformation from meta-data.

4.2 XETL Architecture

XETL's architecture (Fig.1) integrates data from different and heterogeneous sources. Moreover, XETL is an active ETL; it interacts with meta-data to give parameters for extraction programs, to generate the transformation, and to specify the loading with target schema. Hence, it optimizes the flow of data, reduces the update of warehousing process by making automatic the creation of valid transformation, and generates schema from meta-data repository. The different components of XETL tool are described below:

4.2.1 The Meta-Data Components

The Meta-data repository is the essential element in the system. It includes a set of information of which:

- The mapping model (MM): this model is used to describe the mapping expressions. By mapping expressions, we mean the needed information to identify how a target field could be mapped from a set of source fields. Fig.2 represents an example of MM used in [12].
- The source model (SM) contains the model of the source data. It covers relational, object oriented and semi-structured data modelling [11].
- The target model (TM) is alike to (SM); it describes the target data model. Moreover, it could cover the multidimensional model used by OLAP processing.
- Management rules (MR) is the set of rules defined by the administrator in order to fulfil business requirements.

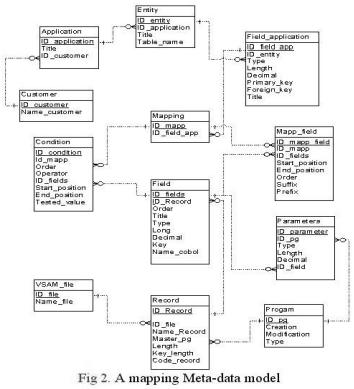

Fig 2. A mapping Meta-data model

4.2.2 The Extraction Process

The role of these programs consists in converting to a common XML format, the different sources. These sources could be traditional databases or inner digital documents that are produced by applications in enterprise or even web documents published by other partners. This first stage supplies the extraction program with the needed parameters from meta-data repository.

4.2.3 The XSLT Generator

The XSLT generator is a module, which can read useful parameters, rules, and mapping specification from meta-data repository to create a style sheet transformation. Concretely, a query is performed on the mapping model (MM). The result file is used by the Generator to produce the style sheet file. Then, an XSLT processor permits to execute the transformation on XML source documents. It generates a new collection of clean XML documents.

Indeed, the selection and filters get rid of superfluous data. During this process, all control and check are applied to data. Thus, the XSLT is more than cosmetic change for XML data; it tackles the content, structure and valid values. The architecture is not limited by the using of XSLT language; any other XML query language can be used to establish such process. For example, X-Query can be useful because it is more adapted for documents databases query and it is more optimized. On the other hand, the stability of XSLT (1.0) and the multi implementation is an advantage over the X-Query.

Model Title	TYPE	Table	Source	CONDITION	TITLE	POSITION
ID_Customer	char(10)	Header_After_sale _customer	element.txt	TOP3 = "E" AND DONNEEB2="CSAV" AND CODENTITE="CLIENT"	CLELEMENT	
ID_HASC	char(5)	Header_After_sale _customer	element.txt	TOP3 = "E" AND DONNEEB2="CSAV" AND CODENTITE="CLIENT"	CLEENTITE	
Date_HASC	YYYYMM DD	Header_After_sale _customer	element.txt	TOP3 = "E" AND DONNEEB2="CSAV" AND CODENTITE="CLIENT"	DATE1	
ID_LASC	num(10)	Line_After_sale _customer	element.txt	TOP3 = "L"	CLELEMENT	1-10
ID_HASC	num(4)	Line_After_sale _customer	element.txt	TOP3 = "L"	CLELEMENT	11-14
Type	char(10)	Line_After_sale _customer	element.txt	TOP3 = "L"	DONNEEB2	
Description	char(14)	Line_After_sale _customer	element.txt	TOP3 = "L"	DONNEEA1	1-14
Characteristic	char(30)	Line_After_sale _customer	element.txt	TOP3 = "L"	LIBELEM	1-30
Line_statue	char(2)	Line_After_sale _customer	element.txt	TOP3 = "L"	DONNEEA5	1-2
Weight	num(5,2)	Line_After_sale _customer	element.txt	TOP3 = "L"	DONNEEA5	3-7
Location	char(10)	Line_After_sale _customer	element.txt	TOP3 = "L"	DONNEEA3	

Fig 3. The result of query into mapping meta-data (MM)

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" version="1.0" encoding="UTF-8" omit-xml-declaration="no" indent="no"
media-type="text/html"/>

<xsl:template match="Import">
 <Import>
    <xsl:for-each select="Row">
      <xsl:if test="TOP3='E'">
          <xsl:if test="(substring(DONNEE2,1,4)='CSAV' and
                                substring(CODENTITE,1,6)='CLIENT')">
              <Header_After_Sale_Customer>
                  < ID_HASC >
                    <xsl:value-of select="substring(CLEELEMENT,1,10)"/>
                  </ID_HASC>
                  < ID_Customer >
                        <xsl:value-of select="substring(CLEENTITE,1,5)"/>
                  </ID_Customer>
                < Date_HASC >
                  <xsl:if test="not(DATE1='      ')">
                    <xsl:value-of select="concat(substring(DATE1,7,2),
                      '/',substring(DATE1,5,2),'/',substring(DATE1,1,4))"/>
                    </xsl:if>
                  </Date_HASC>
              </ Header_After_Sale_Customer >
            </xsl:if>
          <xsl:if test="(substring(DONNEE2,1,7)='FSAVCDE' and
                    substring(CODENTITE,1,10)='FOURNISSEU' and
                    (not(substring(CLEELEMENT,11,1)='A')))">
                  <Header_After_Sale_Supplier>
                  <ID_HASS>
                    <xsl:value-of select="substring(CLEELEMENT,1,10)"/>
                  </ID_HASS >
                  <ID_supplier>
                    <xsl:value-of select="substring(CLEENTITE,1,5)"/>
                    </ ID_supplier >
                  <Date_HASS>
                    <xsl:if test="not(DATE1='      ')">
                      <xsl:value-of select="concat(substring(DATE1,7,2),
                        '/',substring(DATE1,5,2),'/',substring(DATE1,1,4))"/>
                            </xsl:if>
                    </ Date_HASS >
                </ Header_After_Sale_Supplier >
            </xsl:if>
          </xsl:if>
      </xsl:for-each>
    </Import>
  </xsl:template>
</xsl:stylesheet>
```

Fig 4. XSLT transformation generated by ***XSLT-Generator***

A *XSLT-Generator* prototype written in C++ permits to create formatted XSLT transformations, this prototype is described with a case study in [12]. The *XSLT-Generator* communicates with meta-data to read needed mapping. Indeed, a query is performed on MM, which is a part of the meta-data, and the result Fig.3 is used to generate the transformations Fig.4.

4.2.4 The XML Schema Generator

The part of the meta-data used to produce the XML Schemas is the Target Model (TM). The generator reads the target model and creates an XML schema. In particular, to each entity of the target model is associated an XML schema. The most interesting point in this generation is that it supplies the loading process by providing schemas that include data types.

4.2.5 The Loader

The fourth step consists on loading data to warehouse database with schema description. The result XML documents should be valid to XML schema. The loader deals with the validation of these documents. If an error occurs, the loading process is interrupted, and the error is searched for upstream. Therefore, the integrity of target DW database will be preserved.

5 RELATED WORKS

Our work differs from the work done in Xyleme project [10]. Xyleme is XML base web-house, it stores XML pages in order to constitute an XML data repository. Xyleme deals with storage of huge quantities of XML data, query processing, and data acquisition strategies to update repositories. It provides control with services such as query subscription and semantic data integration to free users from having to deal with many specific DTD when expressing queries [1]. Our approach deals with different problematic. We are not studying the issue of building a data warehouse for the web (data web-house). We discussed the issue how XML can change the existing ETL tools and the advantage of such solution.

Other studies show the usefulness of XML for DW restitution, were defined in [6]. A notion of XML-star schema is defined; it also provides the possibility to explore dimensions with XML data. Our approach can fit together with the work done in [6]. Indeed the two approaches supply data warehouse with clean XML document extracted from different sources with XETL to be delivered after as star model with XML data dimensions.

6 CONCLUSION AND FUTURE WORK

Enterprise data warehouse systems will evolve into federated structures. The next generation of data warehouses should be able to handle changing business requirements including real-time warehouse, integration data from web sources, etc.

The idea of this paper consists in associating XML with data warehousing. XML is a neutral format; it is possible to convert to XML format different data sources such as flat file, relational tables, web relational, etc. Every application requires translation and exchange should benefit from XML suppleness and capability. For that, converting many databases sources into XML format before integration of these data inside our warehouse overlaps the using of owner tools format. This is easy with an open data source such as XML. For this, we proved that XML is in the right place inside this architecture to integrate different data sources. The second issue is about creating an XML based ETL tool. This means how to merge all the data coming from different sources together, clean data, and load it into the repository of our data warehouse. The last point for using XML in this architecture consists on data restitution by end users. We investigated the design of an XML based ETL tool. This includes translation of data source to XML, cleansing the sources XML files with XSLT transformations, and loading XML results documents to user application with their schemas.

Future work will be held on the meta-data level to create a more suitable and efficient model for describing meta-data of data warehouse. Such issues were briefly studied in [5] and [8]. Extending these notions, investigating usability, and defying ontology-based model will be the core of our future study.

7 REFERENCES

[1] A.Marian, S.Abiteboul: “Change-Centric Management of Versions in an XML Warehouse”, VLDB 2001.

[2] C.Quix: “Repository Support for Data Warehouse Evolution”, proceedings of the International Workshop on Design and Management of Data Warehouses, DMDW’99, Heidelberg Germany 1999.

[3] E.Alsène, “The Computer Integration of the Enterprise”, IEEE Data Engineering Management, Vol.46, pp.26-35 February 1999.

[4] G. Kappel, E. Kapsammer, W.Retschitzegger: “ XML and Relational Database Systems: A Comparison of Concepts”, Proceedings of the International Conference on Internet Computing, IC’2001, Las Vegas, USA.

[5] J. Van Zyl, D. Corbett, Millist W. Vincent: “An Ontology of Metadata for a Data warehouse Represented in Description Logics”, CODAS’99, Wollongong, Australia, March 27-28, 1999. Springer, Singapore, ISBN 9814021644, pp. 27-38.

[6] J.Pokorny, “Modelling Stars Using XML”, ACM 4th International Workshop on Data Warehousing and OLAP, DOLAP 2001, Atlanta, GA, USA, November 2001.

[7] R.Bourret: “XML and Databases”, http://www.rpbourret.com/xml/XMLAndDatabases.htm

[8] R.Rifaieh, N.A.Benharkat: “A Translation Procedure to Clarify the Relationship between Ontologies and XML Schema”, 2nd Internet Computing Conference IC2001, Las Vegas, USA June 2001.

[9] T.Stöhr, R.Müller, E.Rahm: “An integrative and Uniform Model for Metadata Management in Data Warehousing Environments”, in proceeding of the International Workshop on Design and Management of Data Warehouse DMDW’99, Germany June 1999. pp.40.

[10] Xyleme: “A Dynamic Data Warehouse for XML Data of The Web”, IEEE Data Engineering, June 2001 Vol.24 No.2

[11] Müller, R., Stöhr, T., Rahm, E.: “An Integrative and Uniform Model for Metadata Management in Data Warehousing Environments”. Proc. Workshop on Design and Management of Data Warehouses (DMDW’99), Heidelberg, June 1999

[12] R.Rifaieh, N.A.Benharkat: “XETL: XML-based Data Warehousing Tool”, in proceedings of International Conference on Information Integration and Web-based Applications and Services IIWAS 2002, September 2002, Bandung, Indonesia.