AN ARCHITECTURE FRAMEWORK
FOR COMPLEX DATA WAREHOUSES

Jérôme Darmont, Omar Boussaïd, Jean-Christian Ralaivao and Kamel Aouiche

ERIC, University of Lyon 2
5 avenue Pierre Mendès-France
69676 Bron Cedex
France
Contact email: jerome.darmont@univ-lyon2.fr

Keywords: Data warehousing, Complex data, XML warehousing, Metadata, Domain-specific knowledge.

Abstract: Nowadays, many decision support applications need to exploit data that are not only numerical or symbolic, but also multimedia, multistructure, multisource, multimodal, and/or multiversion. We term such data complex data. Managing and analyzing complex data involves a lot of different issues regarding their structure, storage and processing, and metadata are a key element in all these processes. Such problems have been addressed by classical data warehousing (i.e., applied to "simple" data). However, data warehousing approaches need to be adapted for complex data. In this paper, we first propose a precise, though open, definition of complex data. Then we present a general architecture framework for warehousing complex data. This architecture heavily relies on metadata and domain-related knowledge, and rests on the XML language, which helps storing data, metadata and domain-specific knowledge altogether, and facilitates communication between the various warehousing processes.

1 INTRODUCTION

Data warehousing and OLAP (On-Line Analytical Processing) technologies (Inmon, 2002; Kimball and Ross, 2002) are now considered mature. They are aimed, for instance, at analyzing the behavior of a customer, product, or company, and may help monitoring one or several activities (commercial or medical pursuits, patent deposits, etc.). More precisely, they help analyzing these activities under the form of numerical data. However, in real life, many decision support fields (customer relationship management, marketing, competition monitoring, medicine...) need to exploit data that are not only numerical or symbolic. We term such data complex data. Their availability is now very common, especially since the broad development of the World Wide Web. For example, a medical file is usually constituted of several pieces of data under various forms. A patient's medical history might be recorded as plain text. Various biological exam results might be indicated in many ways. The file could also include radiographies (images) or echographies (video sequences). Successive diagnoses and therapies might be recorded as text or audio documents, etc. Another example could be a collection of web documents concerning a given topic, which would be available under various formats (videos, images, sounds, texts, etc.).

Complex data might be structured or not, and are often located in different and heterogeneous data sources. Browsing these data necessitates an adapted approach to help collect, integrate, structure and eventually analyze them. A data warehousing solution is interesting in this context, though adaptations are obviously necessary to take into account data complexity. Measures might not necessarily be numerical, for instance. Data volumetry and dating are also other arguments in favor of the warehousing approach. Furthermore, complex data produce different kinds of information that are represented as metadata. These metadata, along with domain-specific knowledge, are essential when processing complex data and play an important role when integrating, managing, or analyzing them. Hence, metadata need to be given even more importance than in classical data warehousing.

The notion of complex data is not straightforward. To clarify this concept, we propose in this paper one definition that presents the different aspects we identify in complex data (Section 2). This definition is, to the best of our knowledge, somewhat complete and pertinent, but its scope could certainly be widened. We also propose a general architecture framework for warehousing complex data (Section 3). This model heavily relies on metadata and domain-specific knowledge. It also rests on the XML language that we use for different purposes: to store complex data, if necessary; to store metadata and knowledge about these complex data; and to facilitate communication between the different warehousing processes — ETL (Extract, Transform, Load) and integration, administration and monitoring, and analysis and usage. We finally conclude this paper and provide research perspectives (Section 4).

2 A DEFINITION OF COMPLEX DATA

Many researchers in several communities start to claim they work on complex data. However, this emerging concept of complex data varies a lot, even within a single research community such as the database community. Hence, in a first step, we performed an extensive litterature study to identify all the different sorts of data researchers dealt with. We particularly, but not exclusively, focused on publications and events that explicitely mentionned the terms "complex data", which particularly emerge in the data mining field (Gançarski and Trousse, 2004). After compiling all this information, we were able to propose a first definition and concluded that data could be qualified as complex if they were: (1) *multiformat*, i.e., represented in various formats (databases, texts, images, sounds, videos...); and/or (2) *multistructure*, i.e., diversely structured (relational databases, XML document repositories...); and/or (3) *multisource*, i.e., originating from several different sources (distributed databases, the Web...); and/or (4) *multimodal*, i.e., described through several channels or points of view (radiographies and audio diagnosis of a physician, data expressed in different scales or languages...); and/or (5) *multiversion*, i.e., changing in terms of definition or value (temporal databases, periodical surveys...).

However, it appeared in subsequent meetings with fellow researchers that this first definition was not sufficient to cover the wide variety of complex data. It could indeed be viewed as an axis of complexity, among other axes dealing with data semantics or processing, for instance (Figure 1). Data volumetry could also be such an axis. Though data volume is not an expression of intrinsic complexity if viewed in terms of database tuples, it becomes a complex problem to deal with in statistics or data mining when it is the number of attributes that increases. In conclusion, we define in this section a framework that helps identifying what we term complex data. However, since this definition cannot be exhaustive, we leave it open to new axes of complexity.

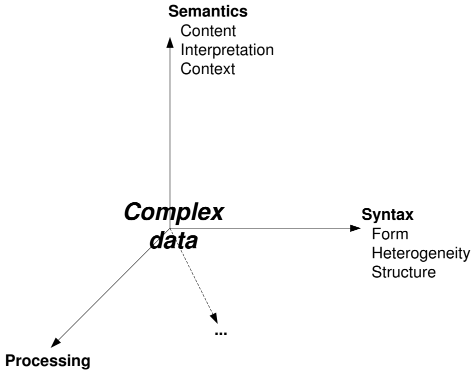

Figure 1: Axes of data complexity

3 COMPLEX DATA WAREHOUSE ARCHITECTURE FRAMEWORK

In opposition to classical solutions, complex data warehouse architectures may be numerous and very different from one another. However, two approaches seem to emerge.

The first, main family of architectures is data-driven and based on a classical, centralized data warehouse where data are the main focus. XML document warehouses (Xyleme, 2001; Baril and Bellahsène, 2003; Hümmer et al., 2003; Nassis et al., 2004) are a examples of such solutions. They often exploit XML views, which are XML documents generated from whole XML documents and/or parts of XML documents. A data cube is then a set of XML views.

The second family of architectures includes solutions based on virtual warehousing, which are process-driven and where metadata play a preeminent role. These solutions are based on mediator-wrapper approaches and exploit distributed data sources. These sources' schemas are one of the main information mediators exploit to answer user queries. Data are collected and multidimensionnally modeled (as data cubes, to constitute OLAP analysis contexts) on the fly to answer a given decision support need (Ammoura et al., 2001).

Whatever the type of architecture, the various processes used in data warehousing always deal with metadata and domain-specific knowledge, in order to achieve a better exploitation and good performances. Note that complex data are generally represented by descriptors that may either be low-level information (an image's size, an audio file's duration, the speed of a video sequence...) or relate to semantics (relationships between objects in a picture, topic of an audio recording, identification of a character in a video sequence...). Processing the data thus turns out to process their descriptors. Original data are stored, for instance as binary large objects (BLOBs), and can also be exploited to extract information that could enrich their own caracteristics (descriptors and metadata).

The architecture framework we propose for complex data warehousing (Figure 2) exploits the XML language. Using XML indeed facilitates the integration of heterogeneous data from various sources into the warehouse; the exploitation of metadata and knowledge (namely regarding the application domain) within the warehouse; and data modeling and storage. The presence of metadata and knowledge in the data warehouse is aimed at improving global performance, even if their actual integration is still the subject of several research projects (McBrien and Poulovassilis, 2001; Baril and Bellahsène, 2003; Shah and Chirkova, 2003).

This architecture framework is essentially made of: the *data warehouse kernel*, which may be either materialized as an XML warehouse, or virtual (where cubes are computed at run time); *data sources*; *source type drivers* that notably include mapping specifications between the sources and XML; and a *metadata and knowledge base layer* that includes three submodules related to three management processes.

The three processes for managing a data warehouse are: the *ETL and integration* process that feeds the warehouse with source data from operational databases ($DS\ Op$) by using drivers that are specific to each source type (ST); the *administration and monitoring* process ($MD\&KR$) that manages metadata and knowledge (the administrator interacts with the data warehouse through this process); and the *analysis and usage* process that runs user queries, produces reports, builds data cubes, supports OLAP, etc. Each of these processes exploits and updates the metadata and the knowledge base. There are four types of flows: the *external flow*, which includes the ETL and integration flow and the exploitation (analysis and usage) flow (the warehouse may thus be considered as a black box); the *internal flow*, between the warehouse kernel and the metadata and knowledge base layer and between the metadata and knowledge base layer and the source type drivers; the *metadata and knowledge management and maintenance flow*, which acquires new knowledge and enriches existing knowledge; and the *reference flow*, which illustrates the fact that the external flow always refers to the metadata and knowledge base layer for integration, ETL, and analysis and usage in general.

Note that analysis results under the form of cubes, reports, queries, or any other intermediary results may constitute new data sources ($DS\ Res$) that may be reintegrated into the warehouse.

Though our proposal is only an architecture framework, it helps us formalizing the warehousing process of complex data as a whole. Thus, we are able to identify the issues to be solved. We can also point out the great importance of metadata in managing and analyzing complex data. Furthermore, piloting and synchronizing the data warehouse processes we identify in this framework is a whole problematic in itself. Optimization techniques will be necessary to achieve an efficient management of data and metadata. Communication techniques, presumably based on known protocols, will also be needed to build up efficient data exchange solutions.

4 CONCLUSION AND PERSPECTIVES

We addressed in this paper the problem of warehousing complex data. We first clarified the concept of complex data by providing a precise, though open, definition of complex data. Then we presented a general architecture framework for warehousing complex data. It heavily relies on metadata and domain-specific knowledge, which we identify as a key element in complex warehousing, and rests on the XML language, which helps storing data, metadata and knowledge, and facilitates communication between the various warehousing processes. This proposal takes into account the two main possible families of architectures for complex data warehousing (namely virtual data warehousing and centralized, XML warehousing). Finally, we rapidly presented the main issues in complex data warehousing, especially regarding data integration, the modeling of complex data cubes, and performance.

This study opens many research perspectives. Up to now, our work mainly focused on the integration of complex data in an ODS. Though we also worked on the muldimensional modeling of complex data, this was our first significant advance into the actual warehousing of complex data. In order to test and refine our hypotheses in the field, we plan to apply our proposals on three different application domains we currently work on (medicine, banking and geography). Such practical applications should help us devise solutions about the many issues regarding metadata management and performance, and experiment both the virtual and XML warehousing solutions.

One of our important perspectives deals with the selection of a representation mode for metadata and domain-specific knowledge. Knowledge related to the application domain is actually an operational information about complex data. It may be considered as metadata. In order to remain in the XML-based, homogeneous environment of our architecture framework, the formalisms that seem best-fitted to represent metadata are XML and RDF (Resource Descrip-

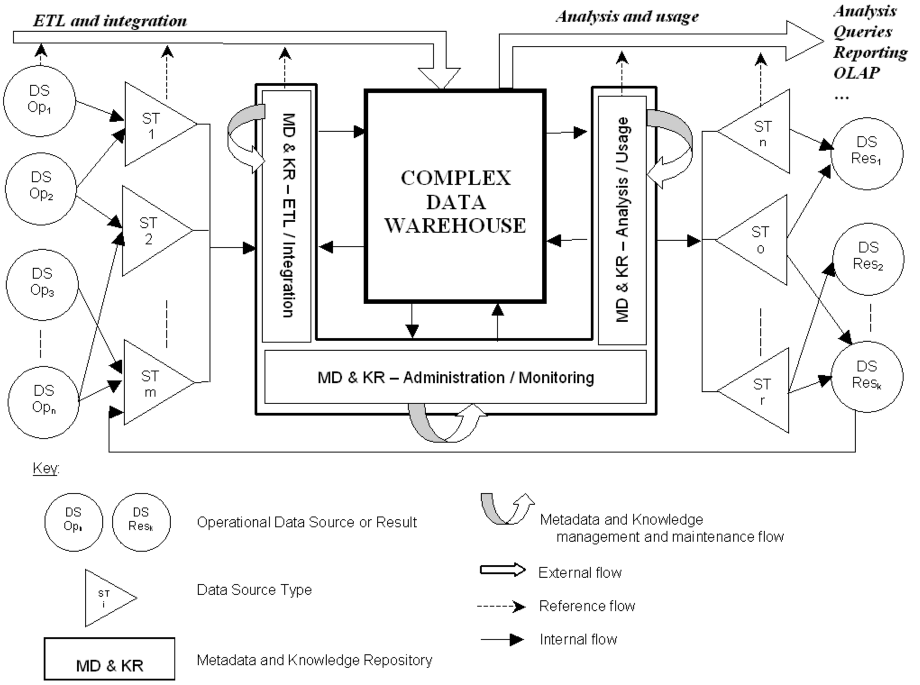

Figure 2: Complex data warehouse architecture framework

tion Framework) schemas. These tools are appropriate to represent both low-level and semantic descriptors. Furthermore, they are adapted to reasoning for metadata exploitation. The Common Warehouse Metamodel (CWM), an OMG standard for data warehouses (OMG, 2003), could also help us managing metadata and knowledge. But can the CWM metamodels integrate the performance factors of a complex data warehouse? Should these metamodels be extended or would it be more interesting to propose new submodels instead? These are largely open questions.

REFERENCES

Ammoura, A., Zaiane, O. R., and Goebel, R. (2001). Towards a Novel OLAP Interface for Distributed Data Warehouses. In *3rd International Conference on Data Warehousing and Knowledge Discovery (DaWaK 01), Munich, Germany*, volume 2114 of *LNCS*, pages 174–185.

Baril, X. and Bellahsène, Z. (2003). *Designing and Managing an XML Warehouse*, pages 455–473. XML Data Management. Addison Wesley.

Gançarski, P. and Trousse, B., editors (2004). *Complex data mining in a KDD process (EGC 04 workshop), Clermont-Ferrand, France*.

Hümmer, W., Bauer, A., and Harde, G. (2003). XCube: XML for data warehouses. In *6th ACM International Workshop on Data Warehousing and OLAP (DOLAP 03), New Orleans, USA*, pages 33–40.

Inmon, W. H. (2002). *Building the Data Warehouse*. John Wiley & Sons, third edition.

Kimball, R. and Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, second edition.

McBrien, P. and Poulovassilis, A. (2001). A Semantic Approach to Integrating XML and Structured Data Sources. In *13th International Conference on Advanced Information Systems Engineering (CAiSE 01), Interlaken, Switzerland*, volume 2068 of *LNCS*, pages 330–345.

Nassis, V., Rajugan, R., Dillon, T. S., and Rahayu, W. J. (2004). Conceptual Design of XML Document Warehouses. In *6th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 04), Zaragoza, Spain*, pages 1–14.

OMG (2003). *Common Warehouse Metamodel (CWM) Specification version 1.1*. Object Management Group.

Shah, A. and Chirkova, R. (2003). Improving Query Performance Using Materialized XML Views: A Learning-Based Approach. In *1st International Workshop on XML Schema and Data Management (XSDM 03 - ER 03 Workshops), Chicago, USA*, volume 2814 of *LNCS*, pages 297–310.

Xyleme, L. (2001). A Dynamic Warehouse for XML Data of the Web. *IEEE Data Engineering Bulletin*, 24(2):40–47.