Finding an Application-Appropriate Model for XML Data Warehouses

Franck Ravat, Olivier Teste, Ronan Tournier[*], Gilles Zurfluh

[*]=corresponding author
IRIT (UMR5505), Institut de Recherche en Informatique de Toulouse
Université de Toulouse, 118 route de Narbonne
F-31062 Toulouse Cedex 9, France
{ravat,teste,tournier,zurfluh}@irit.fr

Abstract. Decision support systems help the decision making process with the use of OLAP (On-Line Analytical Processing) and data warehouses. These systems allow the analysis of corporate data. As OLAP and data warehousing evolve, more and more complex data is being used. XML (Extensible Markup Language) is a flexible text format allowing the interchange and the representation of complex data. Finding an appropriate model for an XML data warehouse tends to become complicated as more and more solutions appear. Hence, in this survey paper we present an overview of the different proposals that use XML within data warehousing technology. These proposals range from using XML data sources for regular warehouses to those using full XML warehousing solutions. Some researches merely focus on document storage facilities while others present adaptations of XML technology for OLAP. Even though there are a growing number of researches on the subject, many issues still remain unsolved.

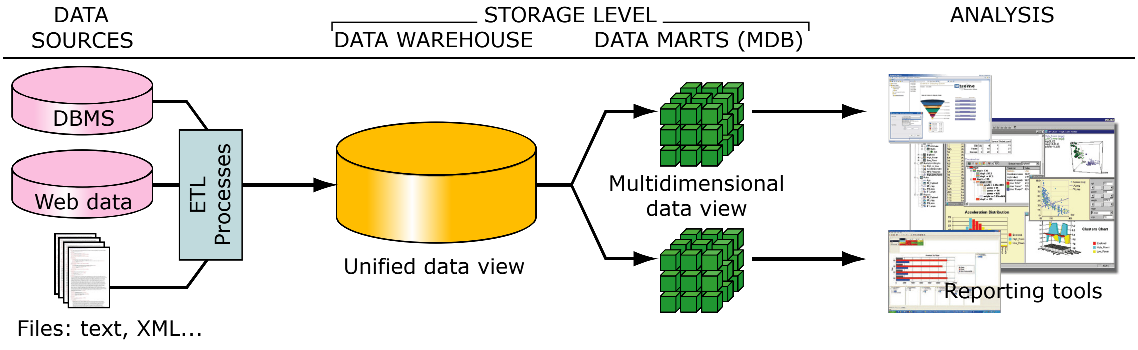
Keywords. Data warehouse; Document warehouse; XML; OLAP; Decision support system.

1 Introduction

Decision Support Systems (DSS), with the use of OLAP software (On-Line Analytical Processing), allow decision-makers to gain insight into corporate data by analysing aggregated historical business or scientific data [14]. Analysis data volumes have been increasing over the recent years and more and more data is available, notably using the XML (eXtensible Markup Language) exchange format. As a consequence, DSS have gradually been integrating this data. However, when designing an application within this context, many solutions exist.

1.1 Decision Support Systems and data warehouses

Decision support systems rest on a central repository composed of several storage spaces: the data warehouse and the data mart [36] (see Fig. 1). Corporate data sources are unified with ETL processes (Extracting, Transforming and Loading). The data warehouse presents a uniform view of corporate information system data. Analysis data is extracted from the warehouse and fed into data marts (i.e. Multidimensional DataBases—MDB). Data marts are multidimensionally structured according to analysis requirements —for example the analysis of the performance of a sales department. Decision makers run analyses on these data marts using reporting tools. It should be noted that some authors use the term data warehouse to designate the whole storage level.

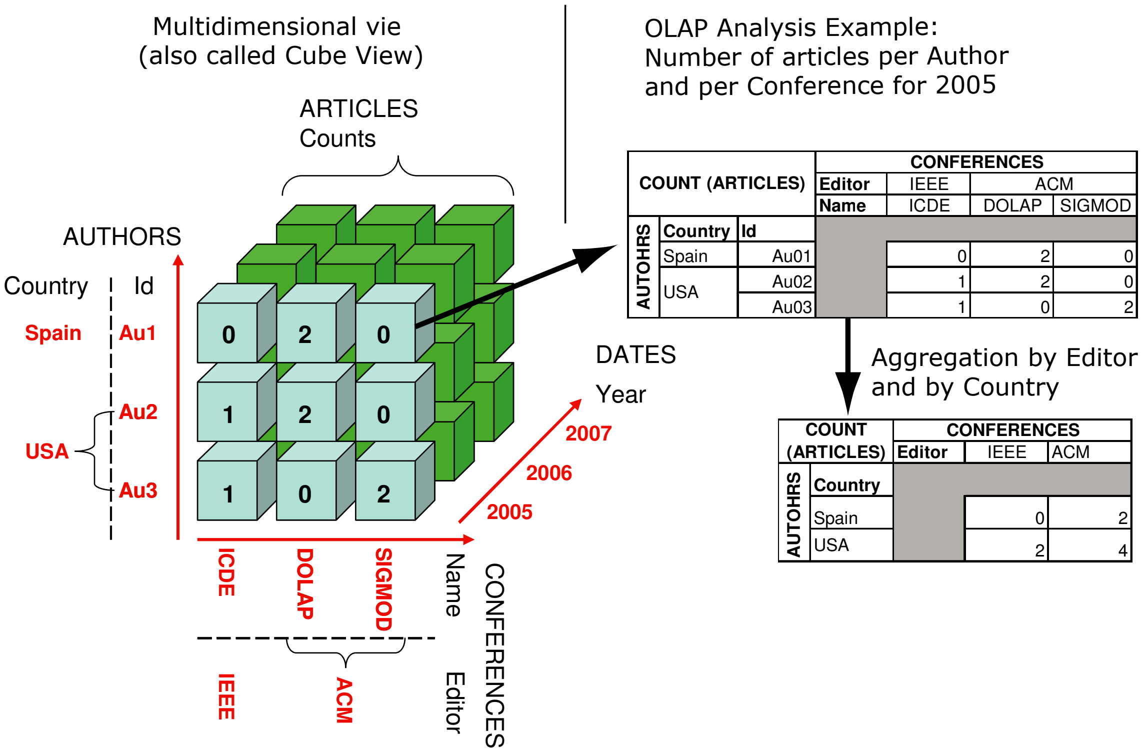


DBMS = DataBase Management System; **ETL** = Extracting Tranforming Loading; **MDB** = Muldimensional DataBase

Fig. 1 Decision Support System (DSS) architecture

Multidimensional modelling within data marts (or MDB) represents data as points in a multidimensional space, hence the *cube* or *hypercube* metaphor (see Fig. 2). To design MDBs, structures were created: *facts* model subjects of analysis (e.g. the number of articles published) and *dimensions* model analysis axes (e.g. when and who published an article). Facts and their associated dimensions form star schemas [36]. In this case, articles are linked to an author, a year of publication and a conference. Facts are conceptual groupings of analysis indicators, namely *measures*, such as the number of published articles. Dimensions are composed of hierarchically ordered *parameters* (or levels) which model the different levels of detail of the analysis axes [36], such as the name of a conference and the proceedings editor company.

Multidimensional analyses (or OLAP style analyses) are performed on the data stored in the data marts. These analyses are done by aggregating the factual data according to an aggregation function (COUNT in our example) and displaying the results according to two analysis axes at several detail levels (in the upper part of Fig. 2, results are displayed according to authors' id, authors' country of origin, conference name and conference editor). Then in a second step, data is aggregated once more and displayed according to the author's country and conference editors (lower right part of Fig. 2).



COUNT (ARTICLES)			CONFERENCES		
		Editor	IEEE	ACM	
		Name	ICDE	DOLAP	SIGMOD
AUTOHRS	Country	Id			
	Spain	Au01	0	2	0
	USA	Au02	1	2	0
		Au03	1	0	2

COUNT (ARTICLES)		CONFERENCES	
	Editor	IEEE	ACM
AUTOHRS	Country		
	Spain	0	2
	USA	2	4

Fig. 2 Example with the cube view (left) and an aggregation example (right)

As systems have evolved, more and more complex analysis data have been introduced within OLAP software. This paper presents an overview of the research literature that has addressed the case of mixing XML technology with decision support systems and more precisely into data warehouses. The next subsection introduces the XML data format.

1.2 The XML format

The W3C presents XML as: "*Extensible Markup Language (XML) is a simple, very flexible text format… Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.*"[1]

[1] **XML**: taken from http://www.w3.org/XML/ (November 2009).

The content of an XML document is encapsulated within *elements* that are defined by tags [91]. In the document, elements are hierarchically organised in a tree-like structure. Tags may contain additional descriptive information, namely attributes. Two formalisms may be used by XML documents to describe their own structure: DTD (Document Type Definition) and XSchema [94,95]. Elements within documents may be accessed with different languages: XPath [93] and XQuery [92]. XML documents may also easily be reorganised and modified with the XSL Transformation language [96].

Generally one speaks of XML data, but to be more precise, rather than speaking of XML files, the term *XML document* is used. More specifically, there are two types of XML document [20,33]:

- *Data-centric XML documents* are composed of very structured data and are similar to relational data. Mainly used by applications to exchange information (i.e. data representing transactions), these documents can be outputs of e-commerce applications, web logs, database “dumps”, etc.
- *Document-centric XML documents* are less structured at first glance. These are text-rich documents and are the electronic version of our traditional paper documents. They are not adapted for application to exchange information. Amongst these documents one may find scientific articles or proceedings, internal reports, e-books, Web pages with textual data, etc.

The major difference is the order of elements (see Fig. 3 for an example). In data-centric XML documents, the order is not important. E.g. in the case of sales transactions, it is not important whether the details concerning transaction number 1 is before or after those of transaction number 2. On the contrary, in the case of document-centric XML documents the order of the elements is critical. E.g., for an article, the order of paragraphs is important to understand the document.

```
<transactions>
   <transaction id=“t0001”>
      <customer id=“c21”>
         <name>Smith</name>
         <address>...</address>
      </customer>
      <products>
        <product>
         <name>LCD TV 52”</name>
         <qty>1</qty>
        </product>
        ...
      </products>
   </transaction>
   <transaction id=“t0002”>
   ...
   </transaction>
   ...
</transactions>
```

```
<is_journal>
   <issue>Volume 34, Issue 4-5</issue>
   <article>
      <title>Preface</title>
      <author> T. B. Pedersen</author>
      <Paragraph>This special section
      contains extended versions of the
      best papers from the ACM Tenth
      International Workshop on Data
      Warehousing and OLAP (DOLAP’07) which
      was held on November 9, 2007, in
      Lisbon, Portugal, as one of workshops
      associated with the ACM Sixteenth
      Conference on Information and
      Knowledge...</Paragraph>
      <Paragraph>...</Paragraph>
      ...
   </article>
   ...
</is_journal>
```

Fig. 3 Example of a data-centric (left) and a document-centric (right) XML documents. Elements are encapsulated between tags. Tags are defined by “<” “>” characters (closing tags start by “</”). The “id” in some tags is an example of attribute.

1.3 XML and Decision Support Systems

Initially there were two types of warehouses: First, *numerical warehouses* consider transactional data[2] and focus on analysis issues (see right column of Fig. 4). Second, *content warehouses* derived from data warehouses, were created to archive vast quantities of Web data mainly composed of text (see left column of Fig. 4). While numerical warehouses led to *Data Warehouses*, content warehousing led to data repositories (i.e. simple data storages) known as *Web warehouses* [77,1,86,9]. These were derived into *Text warehouses* when little structure was available and *Document warehouses* when structure was available [76]. Contrary to data warehousing, content warehousing is more related to data access and retrieval issues rather than analysis issues. Due to framework shortcoming, multidimensional analysis and OLAP style processing is not the major priority of content warehousing [16].

[2] **Transactional data**: data that could be contained in a database transaction. For example, the data in a database can spread over several tables that represent a transaction such as a sale. E.g. for the sale of a product: the amount and quantity of the product sold, the product description, the customer, the date of the sale, etc.

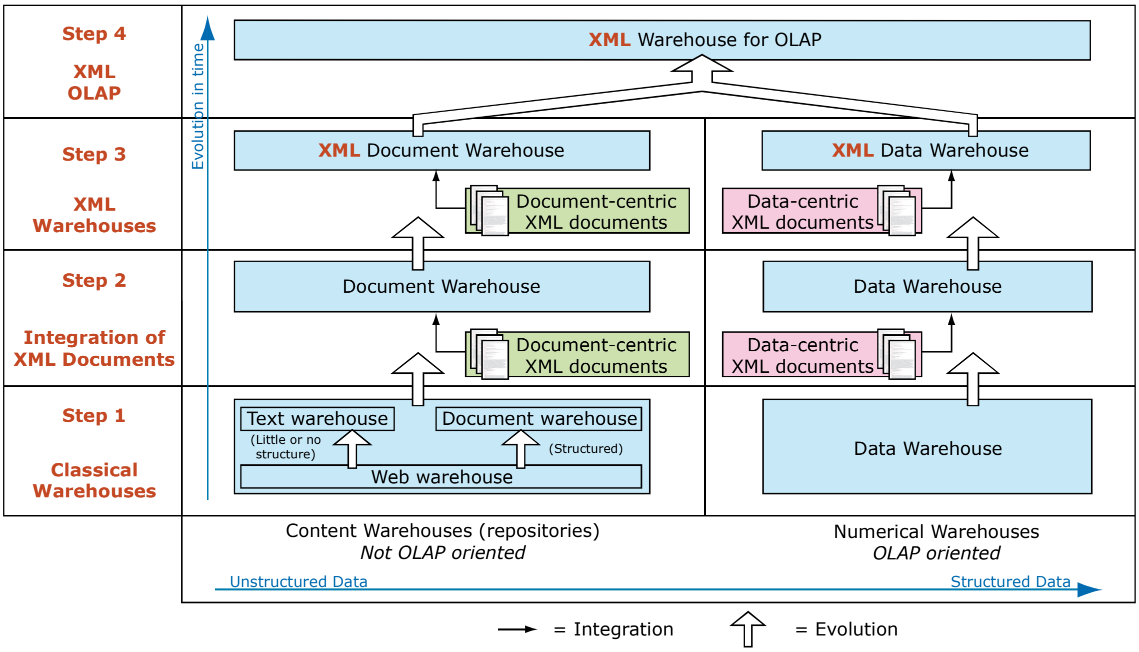


Fig. 4 The progressive use of XML technology with warehousing approaches.

In the past few years, XML has provided a large data exchange framework within corporate information systems. Due to its flexibility and openness, an increasing quantity of XML data can be found on the Web or in corporate information systems, as outputs of e-commerce applications or simply as Web pages. This represents an important data source for decision support systems [22]. As a consequence, XML technology has slowly been included within decision support systems. Furthermore, according to a recent study [80], only 20% of decisional information comes from numerical data or highly organised data such as databases, data-centric documents, etc. In this way, integrating more complex documents (i.e. document-centric documents) into OLAP systems was also considered.

Warehousing these different types of XML documents (i.e. storing this type of data within data warehouses) leads to different warehousing concepts. Initially, integration approaches were developed. But these approaches only considered classical warehousing (data and document warehousing). Later, XML warehousing issues began to be addressed as well as the associated integration processes. The warehousing techniques differ (as well as integration techniques) as the XML documents may be either very structured data (data-centric documents) or much less structured data with a lot more text (document-centric documents). Two types of XML warehouses were designed (see middle part of Fig. 4): XML data warehouses and XML document warehouses. *XML data warehouses* are numerical warehouses that use data-centric XML documents; and *XML document warehouses* that use document-centric XML documents were derived from Web warehouses following the requirement of structuring Web data. It should be noted that document warehouse applications (such as Xyleme [1,98]) are content warehouses and are not adapted to OLAP style processing [16].

Recently an intermediate warehousing technique appeared: *XML document warehouse* ("*XML OLAP*"). OLAP style processing on XML documents can be performed (see top part of Fig. 4) and it is uses both data-centric documents and document-centric documents. Peter Fankhauser states that XML technology is mature enough to offer fairly challenging text mining applications [18] and Dan Sullivan [76] says that text mining should be used on document warehouses. Going further than these statements, this new category of warehouses tries to provide OLAP style processing on document warehouses. As a consequence, these environments benefit from all research done in the OLAP and data warehouse research fields. However, the main issue resides in how to handle textual data, as current OLAP systems cannot easily cope with such data types.

1.4 Organisation of the paper

Within this paper we shall provide a study using two sample data-centric and document-centric XML documents (see Fig. 3). This will be done whenever possible in order to provide a useful comparison between the different analysis frameworks. Moreover, the end of the paper presents guidelines to select a appropriate model when designing applications that combine XML and data warehouse technologies.

The organisation of the rest of this paper follows the steps in the left column of Fig. 4, i.e. it focuses on the usage of XML documents within warehouses and the evolutions of warehousing means in order to provide analysis capabilities. Section 2 states the initial researches that detail integrating XML documents within data warehouses and document warehouses. Section 3 describes solutions for XML data warehousing while section 4 focuses on XML document warehousing. Section 5 presents proposals that mix XML and OLAP technologies. Section 6 describes guidelines to help in choosing an application architecture. Finally section 7 concludes and presents research fields that have not drawn much attention so far.

2 Integration of XML Documents

Context: With the growing availability of XML documents, data contained in these documents where gradually considered as data sources for decision support systems. In this context, data is extracted from XML documents and converted into the native format of the data warehouse (see Fig. 5).

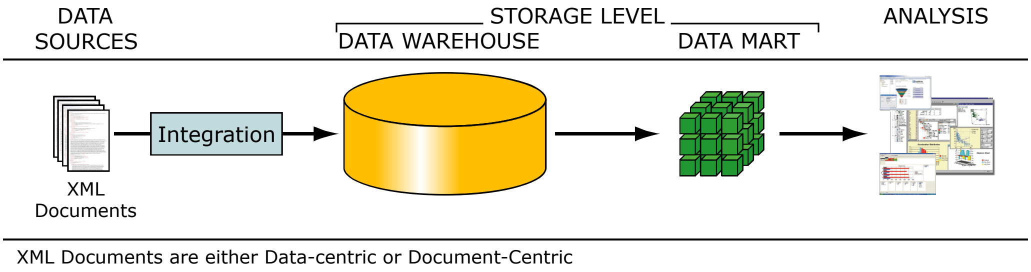


Fig. 5 Integration of XML documents within a data warehouse.

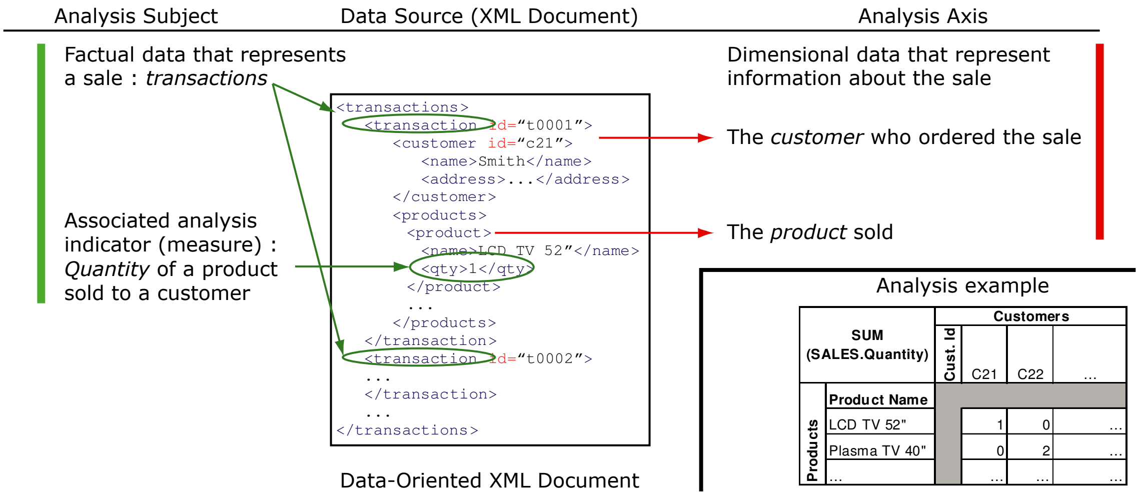
The main problem is the XML semi-structured format that allows many structures to coexist within a collection of documents; however this depends on the document type. On the one hand, data-centric XML documents are highly structured (i.e. similar to a database), their integration is similar to database integration (see subsection 2.1). On the other hand, document-centric ones are more loosely structured and their integration is more complex (see subsection 2.2). Moreover, their content is mostly textual data and it is not very well handled by existing OLAP systems. It should be noted that data-centric XML documents are also used as a communication means between different decision support systems (see subsection 2.3).

2.1 Integration of data-centric XML documents

Context and main issues: Integrating data-centric XML documents is similar to integrating highly structured data (e.g. data extracted from a database) within a data warehouse. Issues are also similar. An example is presented in Fig. 6 where factual data, (i.e. the analysis subject indicator data) and dimensional data (analysis axis data) are extracted from our sample data-centric XML document. In the multidimensional table provided as an analysis example (bottom right table of Fig. 6) all analysis data comes from the XML document. The major issues of this type of integration are due to specificities of the XML data type:

– Conversion of tree-like structured data to the native warehouse format (usually a relational database);

– Elimination of data redundancy (implied by the XML tree-like structure).


Analysis Subject
Data Source (XML Document)
Analysis Axis
Factual data that represents a sale : *transactions*
Associated analysis indicator (measure) : *Quantity* of a product sold to a customer
Dimensional data that represent information about the sale
The *customer* who ordered the sale
The *product* sold
Data-Oriented XML Document
Analysis example

```
<transactions>
   <transaction id="t0001">
      <customer id="c21">
         <name>Smith</name>
         <address>...</address>
      </customer>
      <products>
        <product>
         <name>LCD TV 52"</name>
         <qty>1</qty>
        </product>
        ...
      </products>
   </transaction>
   <transaction id="t0002">
   ...
   </transaction>
   ...
</transactions>
```

SUM (SALES.Quantity)		Customers			
		Cust. Id	C21	C22	...
Products	Product Name				
	LCD TV 52"		1	0	...
	Plasma TV 40"		0	2	...
	...		...	...	...

Fig. 6 Example of decisional data that can be extracted from a data-centric XML document (an output of a Web service that sells products).

Detailed presentation of researches. Initial researches use the Document Type Definition (DTD) to identify data source structures and then feed selected and transformed data to a data warehouse: Golfarelli et al. [22] did it with Web site traffic data and Pokorný [59] with sales data. However, the DTD does not respect the XML grammar. Its alternative, that respects the grammar (XSchema), was used. Using this, Vrdoljak *et* al. detail a purchase order analysis [82] along with the associated integration tool [83]. In the proposals, the authors use multiple XML files as data sources and data is linked together with the ID/IDREF XML mechanism. The logical layer, composed of several XML documents is very similar to a star schema [36] where data is structured in a multidimensional way.

Taking into account heterogeneous data sources, Jensen et al. [31,32], suggest fusing XML and relational data sources into an OLAP system, using UML as a pivot model in order to simplify data integration.

For the problem of space and time consumption when assembling data warehouses, Zhang et al. [100] build a warehouse from distributed XML data. Data is selected according to their access frequency by users within different distributed sources; hence infrequently accessed data is not loaded within the warehouse.

More recently, commercial OLAP suite designers have started to add XML integrator tools (e.g. Oracle, IBM, Microsoft, Business Objects, etc.)[3]. These tools operate easily on data-centric XML documents but they handle the more complex document-centric XML documents with difficulty.

List of unsolved problems. Most relevant issues still open:

- Multiple data structures for a set of XML documents implied by XML structure flexibility;
- No complete ETL framework specified;
- Many slow value-based joins for associating data due to the absence of keys in XML;
- The identification of numerical values and their possible relevance to be analysis indicators.

This last issue is even more important in the document-centric document integration.

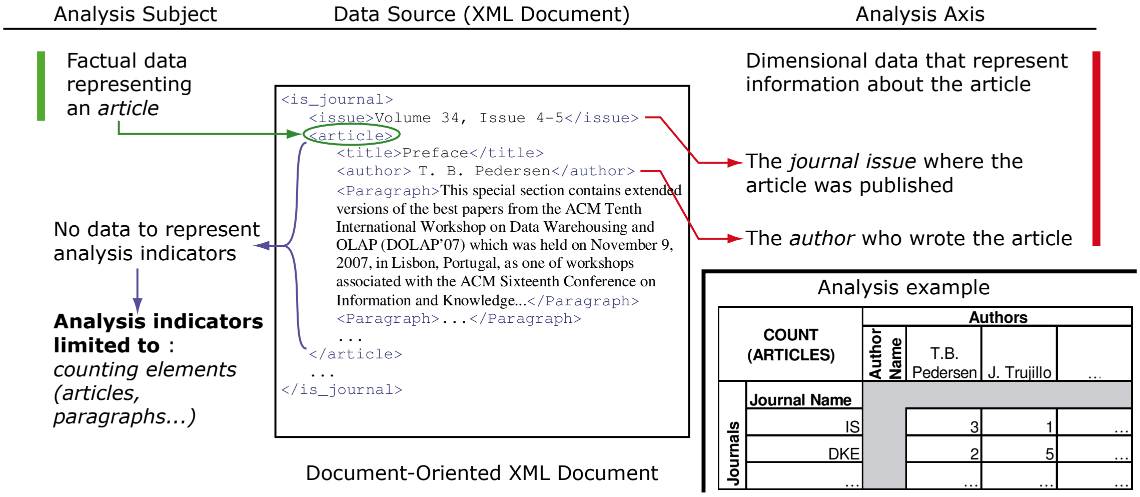
2.2 Integration of document-centric XML documents

Context and main issues: Document-centric XML documents are mainly composed of textual data and therefore barely compatible with actual OLAP environments, that require numerical data as an analysis indicator (quantities, amounts, etc.). The design process uses document metadata as dimensional data. For example, the metadata of a scientific article could be: journal issues, authors, etc. (see Fig. 7). But usually there is no relevant data to be used as analysis indicators. Hence, analyses are limited to counting the documents themselves (see bottom right table of Fig. 7). All the data from the displayed multidimensional table comes from the metadata part of the XML document. However, the large quantity of textual data contained within document-centric XML documents may not easily be integrated within data warehouses (in our example, the textual content of the article is not used and discarded).

The major issues are:

- XML documents loosely structured (much less than data-centric ones);
- XML documents mainly composed of textual data;
- Numerical data is scarce within these documents and this data is difficult to isolate in order to create numerical analysis indicators.

[3] See for example Oracle data integrator suite.


Analysis Subject
Data Source (XML Document)
Analysis Axis
Factual data representing an *article*
No data to represent analysis indicators
Analysis indicators limited to : *counting elements (articles, paragraphs...)*
Dimensional data that represent information about the article
The *journal issue* where the article was published
The *author* who wrote the article
Document-Oriented XML Document
Analysis example

```
<is_journal>
   <issue>Volume 34, Issue 4-5</issue>
   <article>
      <title>Preface</title>
      <author> T. B. Pedersen</author>
      <Paragraph>This special section contains extended
      versions of the best papers from the ACM Tenth
      International Workshop on Data Warehousing and
      OLAP (DOLAP'07) which was held on November 9,
      2007, in Lisbon, Portugal, as one of workshops
      associated with the ACM Sixteenth Conference on
      Information and Knowledge...</Paragraph>
      <Paragraph>...</Paragraph>
      ...
   </article>
   ...
</is_journal>
```

COUNT (ARTICLES)		Authors		
	Author Name	T.B. Pedersen	J. Trujillo	...
Journals	Journal Name			
	IS	3	1	...
	DKE	2	5	...
	...	...	...	...

Fig. 7 Example of decisional data that can be extracted from a document-centric XML document (a scientific article here).

Detailed presentation of researches: As a consequence of the last issue above, document-centric XML documents have mainly been integrated into non OLAP oriented warehouses, i.e. content warehouses (see for example researches around the WhoWeDa project [86] that focus on Web data integration within a content warehouse, but not within the XML context). Document warehouses such as Xyleme [1,98] come with a "document integrator" or a similar tool to facilitate such integration.

However, despite the fact that Document-centric XML documents are barely compatible with actual OLAP environments, some researches have focussed on integrating some of the data contained within these document-centric XML documents. Document-centric XML documents contain a small set of structured data: descriptive metadata, such as the authors, the date of publication, etc. (see the Dublin Core Metadata Initiative [15] for examples on document metadata). The integration of these metadata within an OLAP framework offer some (limited) OLAP analysis. McCabe et al. [41] analyse the location of public document archives, Mothe et al. [44], analyse the content of a document collection in order to enhance keyword based information retrieval and Keith et al. [34], does some lexicography on textual documents. These approaches use a star schema [36] to represent the multidimensional structure and they offer the analysis of documents by counting the number of times a specific term (a word within the text) is used within them. In these approaches, the document content is lost and reduced to a small set of keywords represented through a clustered "keyword dimension" (see [5] for implementation suggestions). Maintenance of this dimension is barely possible as new documents may not easily be added within the system (the whole clustering process might have to be carried out each time). Finally, this dimension is not regular and requires a complex structure (see [40] for a survey on handling these complex dimensions).

List of unsolved problems. Most relevant issues still open:

- Document have multiple metadata structures;
- There is no formal methodology for the integration process (i.e. ETL processing);
- The keyword dimension is complex, reductive and with a complex maintenance on data warehouse updates;
- Document content is systematically left aside.

To conclude, these approaches cannot analyse document-centric XML document contents. Indeed, they reduce document-centric XML documents to data-centric ones that structure the metadata. The documents specificities (large text blocs) are not used during the analysis (see section 5.3 for recent solutions on this issue). Hence only part of the information of the XML document is contained in the data warehouse.

2.3 Importing and exporting multidimensional data with XML

Context and main issues: XML has also been suggested as a mediator between the different storage components but also between the storage components and the analysis tools. The idea is to define common structures with the XML grammar as the Common Warehouse Model (CWM[4]) did for data warehouse structures. In this environment, XML data-centric documents with specific structures are destined to be used by components of OLAP systems to communicate with each other (see Fig. 8).

The major issues are:

- Decisional data exchange;
- Schema description exchange;
- Analysis data transmission.

[4] CWM, Common Warehouse Model, from the OMG: http://www.omg.org/technology/documents/formal/cwm.htm

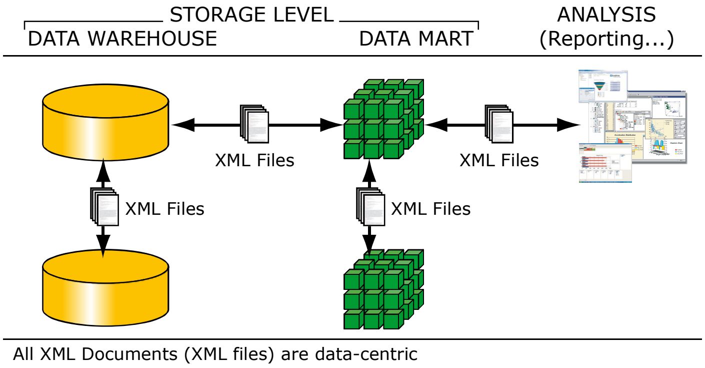


Fig. 8 Communication between DSS components with XML documents.

Detailed presentation of researches: The initial proposals were based on an XML document structure to unify the multidimensional structure used by different systems: XCubeSchema in Hummer et al. [28], and Metacube in Nguyen et al. [48**,**49].

Following the same idea, Niemi et al. [50] assemble, on user request, XML data cubes from distributed data. They use XML as transaction formalism between remote sources and a data warehouse system that stores the cube. Similarly, an interesting idea, although lacking formal specification, is presented by Huang et al. [27], where the authors wrap the export of a data cube in an XML cube structure for an internet-based data warehouse system.

All these papers present systems that generate data-centric XML documents, which may then be integrated within an OLAP system such as a relational OLAP architecture [36]. The most advanced researches wrap decisional info within an XML structure.

Rusu et al. [71] use the XQuery language [92] to present a set of typical queries and the associated data reorganisation necessary to organise XML data into multidimensional XML files to simplyfy the data integration within multidimensional databases.

Coming from heterogeneous database integration, Tseng [79], is one of the rare authors that use XML files during the ETL process that precedes the assembly of a data warehouse. In this proposal, data is extracted from sources into XML files and is merged into a common format with XSLT [96].

List of unsolved problems. Most relevant issues still open:

– No multiple XML document structures;

- No common format either for data or metadata exchange among all the works;
- Lack of advanced functionalities such as handling complex data [13]; in this case, decisional data that would not be limited to text and numbers but could also be graphs, images, etc. (as in medical data).

However, since 2000, XMLA[5] is a standardisation attempt driven by Microsoft and Hyperion and joined by SAP and SAS more recently. XMLA allows client applications to communicate with multi-dimensional or OLAP systems using XML files. Until now neither IBM nor Oracle support this standard attempt. But with the recent acquisition of Hyperion by Oracle and the similarity between IBM's DB OLAP and Hyperion ESS, the situation should evolve in the future.

2.4 Integration of XML documents so far…

Overall summary: Researches on data-centric XML document integration have mainly focused on the use of existing multidimensional structures within XML documents to extract data from these structures and integrate them within a multidimensional database. Concerning document-centric XML documents, the rare researches limit themselves to integrating document metadata that can only provide very limited document content analysis.

	Sources		Decision Support System	Objectives
	Document Type	Structure		
[22,59,100,82,83]	Data-centric documents	DTD, XSchema	Data warehouse	Data analysis
[31,32]	Data-centric documents	DTD + Relational schema	Virtual Data warehouse	Data analysis
[41,44,34]	Document-centric documents	DTD (or no structure at all)	Data warehouse	Metadata analysis
[28,48,49,50,27,71]	Data-centric documents	Predefined structures based on DTD or XSchema	Data warehouse or data mart	Interoperability (no analysis)
[79]	Data-centric documents	XSchemas and associated transformations (XSLT)	Data warehouse	ETL* processing
* ETL = Extraction Transformation Loading				

Table 1 The different researches that integrate XML documents within data warehouses.

Special remark: Although all these researches concern data integration, they rarely speak of ETL (Extraction Transformation Loading) processes, see [81] for a detailed application of ETL processes (but not in the XML context). Moreover, there is no methodology for defining an XML document integration process. According to a recent overview of data warehousing research [69], ETL processes, with traditional data, are not complete and what could be called "XML ETL processes" have not drawn much attention so far. This may be due to the complexity associated with data integration or to the availability of numerous partial "data integrator suites" in commercial tools.

Analysis framework comparison: using our sample XML documents (see Fig. 3), the researches presented in subsection 2.1 are capable of integrating the data-centric XML document and allow the corresponding sales analysis (Fig. 6). All the systems convert the document into their underlying database format. XML specificities are lost and documents must have identical structures. However, these systems cannot easily use the document-centric XML document. On the contrary, all researches in subsection 2.2 are capable of integrating our sample document-centric XML document (see Fig. 3). The document metadata is integrated as dimensional data and the analysis framework uses simple measures such as counting the documents themselves or subparts of the document (sections, paragraphs, etc.), see Fig. 7. All documents must have an identical structure defining their metadata. It should be noted that these systems cannot easily use data-centric XML documents.

Unsolved issues: The integration of XML documents introduces some specific issues such as *heterogeneous structures*. These issues are not handled although some solutions exist for example in schema matching [63] or model management [7]. A solution could be to consider schema relaxation [3]. However this fails when the documents structures are very heterogeneous.

The keyword type dimensions have been investigated but only in terms of how to handle them in the OLAP querying environment [62].

Moreover, numerous semantic issues, inherent to data integration problems have not been solved. With the growing researches on data integration using ontologies [51] or similar knowledge representations, solutions should appear in the near future.

Rather than integrating XML data into a specific multidimensional system, some researches have considered using XML warehouses. This is detailed in the two next sections, section 3 for the case of data-centric XML documents and section 4 for document-centric XML documents.

[5] XMLA: XML for Analysis from http://www.xmlforanalysis.com

3 XML Data Warehouse

Context: In the previous section, XML documents are integrated within traditional data warehouse systems. In this case, data-centric XML documents are integrated within an XML warehouse. When integrating these data-centric XML documents, two approaches are possible:

- Logical integration (or federation), where XML documents are stored apart from the warehouse;
- Physical integration where XML documents are directly stored within the warehouse.

This following subsections detail each approach.

3.1 XML Data Federation: logical integration of XML Documents with a Data Warehouse

Context and main issues: In some cases, it may not be possible to integrate the XML data within the data warehouse. This may be due to legal constraints such as the impossibility to of hosting specific data due to ownership rights or privacy constraints. But this may also be due to physical constraints such as data that changes too quickly and therefore these changes are incompatible with an efficient warehouse refreshing processes (e.g. stock market values). The solution consists in federating two systems: the first hosts the data warehouse and the second (a repository) hosts the XML data. A federated query engine distributes the queries among the two systems (see Fig. 9) and the results are fused together within a multidimensional analysis.

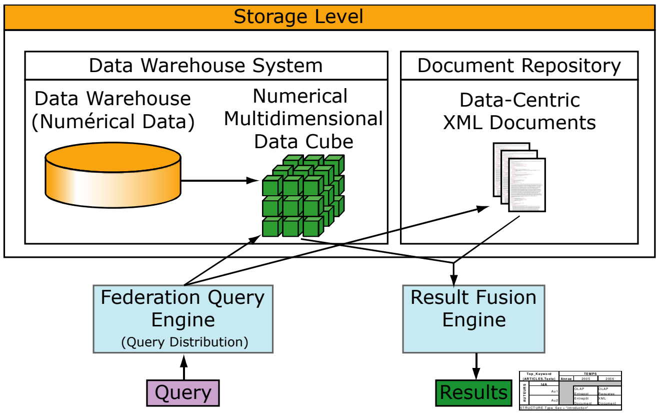


Fig. 9 The XML data federation architecture

The major issues are:

- The fusion of XML and data warehouse data as a result of the OLAP queries;
- The unified view of the federation (a unique schema).

Detailed presentation of researches: Pedersen et al. [53] describe a federation composed of a traditional data warehouse and a set of XML documents. This system uses links between the multidimensional data of the warehouse and the XML data. In [99], they describe the federation implemented within the Targit[6] system. In this system, XML data may be factual data [55] or dimensional data [53]. XML data may also be federated within an existing dimension. This is called "decorating" the dimension with additional information coming from XML sources. Complementary to this approach, in [54], the same authors consider handling changes in the XML document sources.

In [37], Li et al. present a federation between an OLAP system and a repository of XML documents whose structure is defined with XSchema. The authors use UML to translate the XSchema definition into a multidimensional snowflake schema [36]. As a consequence the OLAP system queries the XML repository through a logical layer that represents the multidimensional structures with the snowflake schema.

List of unsolved problems: Most relevant issues still open:

- ETL data integration: as in the previous section, a complete formal framework is still missing;
- Data fusion (required for assembling federated query results); however, it should be noted that a recent overview of data fusion [11] provides valuable information and hints on how to solve some issues.

To conclude, all researches stated in this section suffer from performance issues. Indeed, XML technology is still not as mature as relational databases. Numerous performance issues are still unsolved.

3.2 XML data warehouse: physical integration of XML data

Context and main issues: The objective of such warehouses is to operate on a native XML database and allow OLAP style queries to be specified on the XML warehouse (see Fig. 10). The XML data selected from the warehouse is structured in a multidimensional cube (i.e. a data mart) and OLAP style queries are run on this cube.

[6] Targit Business Intelligence Suite from http://www.targit.com/

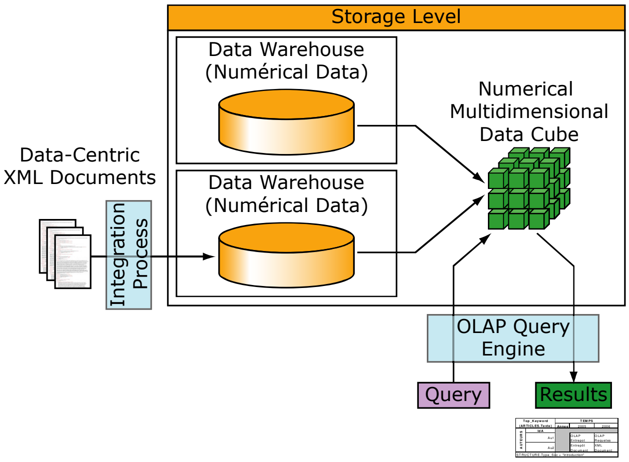


Fig. 10 The XML data warehouse architecture

The major issues of this approach are:

- The multidimensional logical model used within the data mart (i.e. the cube);
- Performance issues, due to the XML database technology less mature than relational databases.

Detailed presentation of researches: Pokorný [60] builds an XML data warehouse resting on a star schema [36] with explicit dimension hierarchies. The author also addresses the problem of the necessity of approximate matching between similar data-centric XML documents that have to be queried together. Similarly, Rusu et al. [72] also build a data warehouse on top of a star schema.

Some researches offer to process complex data, i.e. data-centric XML documents with complementary files such as images. Complex data require specific transformations in order to be integrated within warehouses. Boussaid et al. [13] use a snowflake schema for an XML data warehouse built from breast cancer patient files. Within this approach, the authors use an XML documents to store a fact instance and its associated dimensional data. This approach keeps XML documents close to their original form and they are not split into several documents. The authors have also provided an operator based on data mining techniques [43]. However this operator does not respect closure ensured by all OLAP operators (see [2] or [24] for formal algebraic aspects of OLAP operators). That is, once the operator is used there is no guarantee that other OLAP operators such as drilldown or rollup (zooming in or out of the data) will still operate.

Baril et al. [6] use views to build a warehouse on top of an XML repository. The physical level uses a mapping between the XML data and a relational database. However, the authors do not specify a framework for multidimensional analysis based on their warehouse. Nassis et al. [45] define a conceptual model composed of xFACTs (XML Facts) and VDims (Virtual Dimensions), based on an object-oriented environment. xFACT and VDim are complex structures defined on top of transactional data. As in [6], the authors use XML views to define dimensions over the XML warehouse data. This work has been complemented in [46] and an associated methodology for the definition of user requirements has been defined in [47].

The xFACT structure, more complex than a traditional fact, could allow new analyses, but the framework lacks all reference to an associated manipulation language or algebra that could allow the specification of new multidimensional analyses.

List of unsolved problems: Most relevant issues still open:

– Handling multiple document structures;

– Manipulation operators to be associated to the described multidimensional models;

Strangely, query processing is completely absent from the previously presented researches. Moreover, performance issues are also still a problem due to the lack of maturity of XML databases.

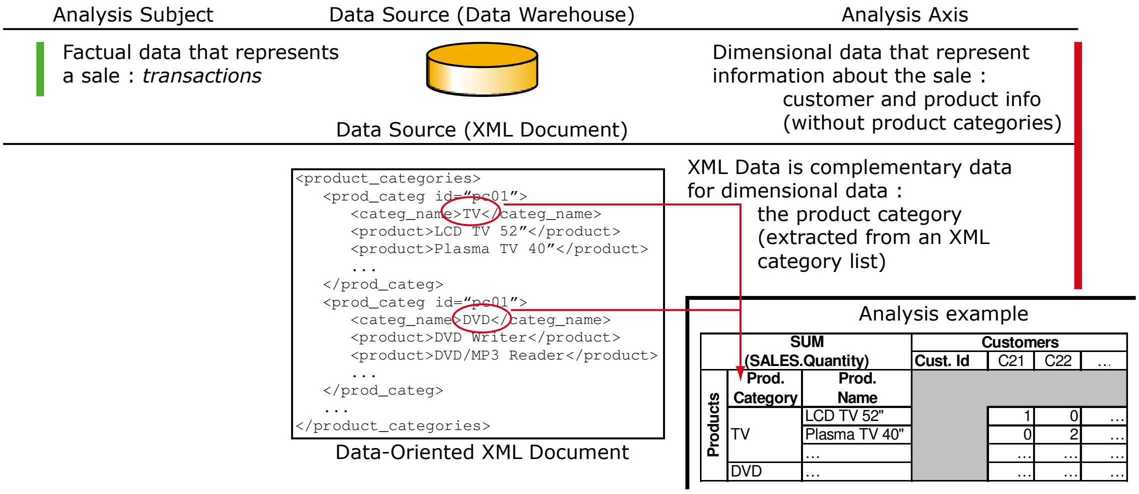
3.3 XML Data Warehouses so far…

Overall summary: A summary of the contributions in the category of XML data warehouses is provided in Table 2. XML data warehouses, as with data warehouses, are based on numerical factual data. Reusing the data warehouse environment, these researches benefit from ten years of research. They focused mainly on the conceptual level (i.e. defining models) and tried to take advantage of the native hierarchical structure of XML documents.

Logical Integration (Federation)				Physical Integration			
	Sources		Analysis Specification (Manipulation Operators)		Sources		Analysis Specification (Manipulation Operators)
	Data Warehouse Schema	XML files contain			Data Warehouse Schema	File Type	
[53,55,99]	Star	Factual data	Decoration Operator	[60]	Star	XML Files	Not detailed
		Dimensional Data		[72,73]	Star	XML Files	Not detailed
		Complementary Data		[13,43]	Snowflake	XML Files with Complex Data	Data Mining Operators
[37]	Snowflake	1 XML File per element of the schema	Ad-Hoc	[6]	Relational Database	XML Files	Not detailed
			Through UML Class Diagram	[45,47]	Specific : xFACT	XML Files	Not detailed

Table 2 Summary of the characteristics of XML data warehousing approaches

Analysis framework comparison: Within the researches based on a federation approach, the documents are not analysed, but their content is used as complementary information for an ongoing analysis whose data comes from the data warehouse system (see Fig. 11). Thus in the multidimensional table provided (bottom right corner of Fig. 11) only the data contained in the column Prod. Category comes from the XML document. The rest comes from the data warehouse. These data-centric XML documents have to be structured according to a specific format. Moreover, they require value based joins to be associated to the data warehouse data (in our example product names). Document-centric XML documents can only be handled in a limited way as their content is much looser and also mostly composed of textual data. However, using their metadata as complementary analysis data could be considered. But no research has focused on this case. Moreover, the document contents would still be discarded.


Analysis Subject
Data Source (Data Warehouse)
Analysis Axis
Factual data that represents a sale : *transactions*
Dimensional data that represent information about the sale : customer and product info (without product categories)
Data Source (XML Document)
XML Data is complementary data for dimensional data : the product category (extracted from an XML category list)
Data-Oriented XML Document
Analysis example

```
<product_categories>
   <prod_categ id=“pc01”>
      <categ_name>TV</categ_name>
      <product>LCD TV 52”</product>
      <product>Plasma TV 40”</product>
      ...
   </prod_categ>
   <prod_categ id=“pc01”>
      <categ_name>DVD</categ_name>
      <product>DVD Writer</product>
      <product>DVD/MP3 Reader</product>
      ...
   </prod_categ>
   ...
</product_categories>
```

	SUM (SALES.Quantity)		Customers			
			Cust. Id	C21	C22	…
	Prod. Category	Prod. Name				
Products	TV	LCD TV 52"		1	0	…
		Plasma TV 40"		0	2	…
		…		…	…	…
	DVD	…		…	…	…

Fig. 11 Example of decisional data extracted from a data-centric XML document in a federation environment.

With the physical integration approach, the content of data-centric XML documents can be analysed. It is simple if the documents possess a structure similar to the XML data warehouse system as only minimal structural changes have to be performed. Elements of the document content are split into factual data and dimensional data (see Fig. 6) with little or no transformation. However, researches have focussed on modelling issues rather than adapted manipulation languages and therefore most of these researches lack means of specifying analyses. Document-centric XML documents cannot be handled. It should be noted that some proposals, [13,43], operate on complex data that resembles document-centric documents. However, these researches lack adequate means to handle textual contents.

Unsolved issues: Several issues are still open:

- XML hierarchical structure;
- Formal manipulation specification;
- Performance issues;
- Logical transition between the conceptual view and physical implementation.

This hierarchical structure is, by nature, not very well adapted to representing all multidimensional structures of the OLAP environment (e.g. complex hierarchies, such as those formalised in [38]). However, graph-like XML structures are currently being developed [30].

Concerning manipulation, despite numerous researches on the subject (see [64] for recent comparative studies), most traditional models lack the associated manipulation algebra that take advantage of the model's concepts in the OLAP environment. So far, in this context and to our knowledge, OLAP manipulation algebras [64] have been neither reused nor extended to take into account XML multidimensional data.

Curiously, although there have been numerous research on performance issues within data warehouses, there is no thorough study on XML data warehouse performance. However, numerous solutions could be borrowed from previous researches, such as materialised views [26,101], compressed cubes [17], etc.

Actual solutions provide mainly a conceptual view and the logical transition towards the physical level still requires to be completed as in traditional warehouses [69]. Research could focus on this logical layer and allow easier physical implementations. This would also benefit researches on performance issues as some of the optimisations may be described at the logical level.

Due to the fact that numerical data warehouses cannot cope easily with textual data, content warehouses (i.e. text or document warehouses) have to be used when warehousing document-centric XML documents. Therefore, XML document warehouses were created.

4 XML Document Warehouse

Context: When integrating document-centric XML documents within warehouses specific repositories are used: document warehouses [76] (e.g. Xyleme[7]). These systems provide a framework for content warehousing, but they are neither analysis nor OLAP oriented [1,16]. Indeed, document-centric XML documents are more complex and less structured than their data-centric cousins. Moreover, as they are mainly composed of textual data, the identification of numerical data destined to be aggregated within factual indicators is very difficult and often impossible. Due to this necessity of numerical data, actual OLAP processing can only operate in a very limited way on such documents.

Problems and issues: The major issues are:

- Possible loose XML structures;
- Multiple document structures;
- Large quantities of textual data;
- Numerical factual data is difficult to locate and identify.

So far, the major focus of XML document warehouses is information retrieval issues, i.e. to get the right document that a user wants. However, some researches have tried to marry the information capacity of XML document warehouses with an OLAP analysis environment.

Two approaches currently exist for using document-centric XML documents within document warehouses:

- A contextualisation approach by leaving the document outside of the data warehousing system;
- An integration approach where only document-centric XML document metadata is integrated within the warehousing system.

4.1 Contextualisation of a data warehouse with XML documents

Context and main issues: The goal of this approach is to provide textual documents to the decision maker as complementary information for an ongoing analysis.

The major issues are:

- The association between an OLAP analysis “context” and the documents;
- Handling the textual content of the document warehouse within the analysis environment.

Detailed presentation of researches: Coming from the information retrieval community, Pérez et al. [56,57] propose to associate two elements: 1) an information retrieval engine on a document warehouse and 2) an OLAP engine on a traditional multidimensional data mart. This approach could be seen as an XML document warehouse federation with a data warehouse. The association between the OLAP analysis context and the documents is performed by using relevance modelling techniques and language models [61].

When the analyst executes a multidimensional query on a data mart, keywords are used to specify an information retrieval query on the document warehouse associated to the data mart (see Fig. 12). The analyst retrieves the results of the analysis as well as documents (mails, new articles, reports, etc.) relevant to the keywords used in his query (e.g. words indicating location, time, etc.). However, the retrieved documents are not analysed by the system and the user has to read each one individually.

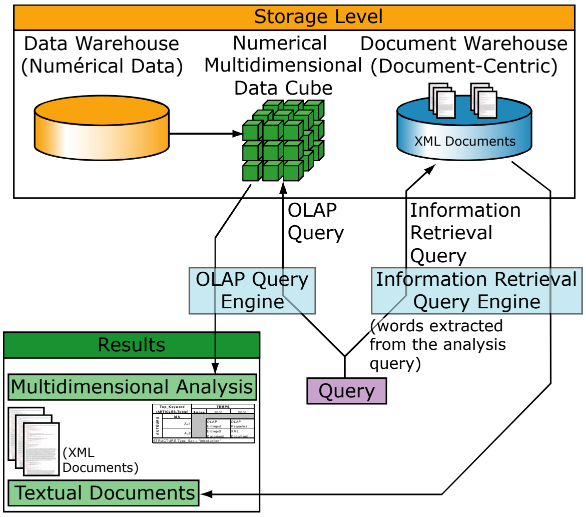

Storage Level
Data Warehouse (Numérical Data)
Numerical Multidimensional Data Cube
Document Warehouse (Document-Centric)
XML Documents
OLAP Query
Information Retrieval Query
OLAP Query Engine
Information Retrieval Query Engine
(words extracted from the analysis query)
Query
Results
Multidimensional Analysis
(XML Documents)
Textual Documents

Fig. 12 The XML document warehouse used as analysis context information architecture

List of unsolved problems: With the lack of document content analysis, the following issues are still open:

- Information overload: if the systems returns too many documents for a single analysis query;

[7] Xyleme Server from http://www.xyleme.com/page/xml_server/

– Time consuming: the decision maker has to read every document returned by the system.

Based on researches presented in section 2.2, some approaches have tried to provide the analysis of document metadata using XML document warehouses,

4.2 Building a data mart from XML document warehouse metadata

Context and main issues: A recent idea is to build adapted data marts based on some of the metadata held within document-centric documents stored in a document warehouse. It is possible to extract information concerning their structure or specific characteristics such as the metadata suggested by the Dublin Core Metadata Initiative [15] (see Fig. 13) and to use these in order to build a data mart.

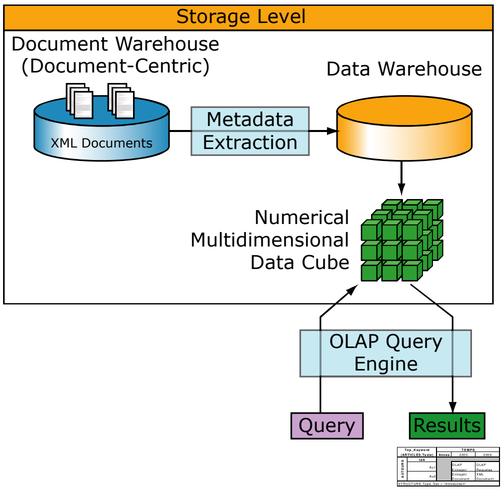


Fig. 13 The analysis of XML document-centric document metadata architecture

Detailed presentation of researches: Khrouf et al. [35] use a document warehouse to store documents grouping them according to their structure. They offer multidimensional analysis of document structure statistics and metadata.

Several other proposals aim at analysing document collections by building adapted data marts (see section 2.2 for more details). Recently, Tseng et al. [80], detail how to build a star schema for different analyses of document metadata and the authors go further and associate their data mart to an XML document warehouse and therefore, it is possible to see the documents concerned with an ongoing analysis.

List of unsolved problems: In all these researches, the document content is not considered. Analysis indicators are limited to simple counts of documents (see analysis example in Fig. 7). Some researches have tried reducing the document content to a keyword dimension (e.g. [44]), but this approach raises many issues (see 2.2).

4.3 XML document warehouses so far…

Overall summary: Frameworks for integrating document-centric XML documents that use XML technology for warehousing the documents have not really focussed on data analysis issues (see Table 3 for a summary). Although all frameworks provide manipulation languages for analysis specifications, analyses are rarely based on the document-centric XML document data.

XML Document Warehousing					
Contextualisation Approach			Document Metadata Integration		
	Storage	Manipulation		Storage	Manipulation
[56,57]	2 spaces : - Data Warehouse - Document Warehouse	Combination : - OLAP (on data) - IR[1] (on documents)	[35]	- Document warehouse - Multidimensional data mart (star schema)	Statistics on document structures (and metadata)
			[80]	- XML files (1 per document) - Multidimensional data mart	Multidimensional queries on metadata

[1] IR = Information Retrieval

Table 3 Summary of the characteristics of XML document warehousing approaches

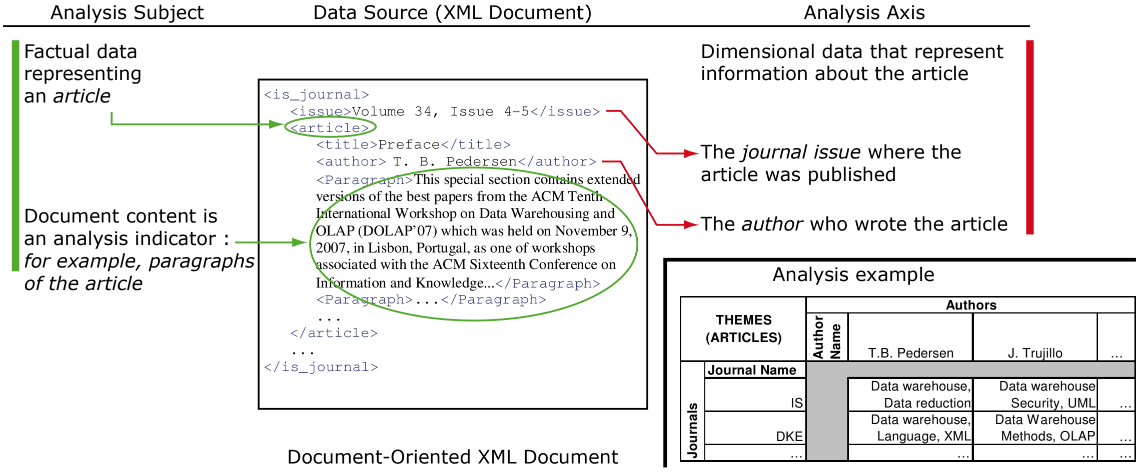
Analysis framework comparison: Contextualisation approaches do not provide any document analysis environment. The contents of the XML documents are read manually by the user as they are returned. However, some authors provide document analysis. Khrouf [35] provides XML structure analysis and Tseng [80] provides document metadata analysis.

Unsolved issues: However these frameworks perform content warehousing and do not consider analysis on the document contents. Even when integrating OLAP style analyses, the indicators are limited to numerical measures (simple counts).

As stated in introduction, XML document warehouses are not OLAP oriented. But some recent researches, detailed in the next section, have tried fusing XML document warehouses with data warehouse OLAP engines.

5 XML Warehouse for OLAP

Context: Providing analysis (OLAP style querying) on either data-centric or document-centric XML documents raises many issues. Notably, analysis on document-centric XML documents could allow the analysis of the textual content of the documents within an OLAP environment (see Fig. 13 for an analysis example that provides the major topics of the analysed documents). In this example, document metadata is used as dimensional data and the textual document content comprises the data of the measure. Here in the analysis example (the multidimensional table displayed in the bottom right corner of Fig. 14) all the data comes from the XML document. Contrary to the example presented Fig. 7 where the document content is not used; this content is now aggregated using a specific aggregation function. This function operates on text as an average aggregation function would operate on numerical data by computing the average sales value. Here, the TOPIC function (inspired by [52]) returns the topics of a fragment of text.


Analysis Subject
Data Source (XML Document)
Analysis Axis
Factual data representing an *article*
Document content is an analysis indicator : *for example, paragraphs of the article*
Dimensional data that represent information about the article
The *journal issue* where the article was published
The *author* who wrote the article
Document-Oriented XML Document
Analysis example

```
<is_journal>
   <issue>Volume 34, Issue 4-5</issue>
   <article>
      <title>Preface</title>
      <author> T. B. Pedersen</author>
      <Paragraph>This special section contains extended
      versions of the best papers from the ACM Tenth
      International Workshop on Data Warehousing and
      OLAP (DOLAP'07) which was held on November 9,
      2007, in Lisbon, Portugal, as one of workshops
      associated with the ACM Sixteenth Conference on
      Information and Knowledge...</Paragraph>
      <Paragraph>...</Paragraph>
      ...
   </article>
   ...
</is_journal>
```

THEMES (ARTICLES)		Authors			
		Author Name	T.B. Pedersen	J. Trujillo	…
Journals	Journal Name				
	IS		Data warehouse, Data reduction	Data warehouse Security, UML	…
	DKE		Data warehouse, Language, XML	Data Warehouse Methods, OLAP	…
	…		…	…	…

Fig. 14 Decisional data that can be extracted from document-centric XML documents by taking into account document textual content.

Problems and issues: in these approaches major issues are mainly linked to XML as well as textual data:

- Data-centric XML documents require adapted querying on the XML data to handle document structures;
- Document-centric XML documents require adapted data marts (i.e. adapted multidimensional structures);
- Text, which is a major component of document-centric XML documents, has to be handled, therefore adapted aggregation processes have to be provided.

The following subsections describe these three issues.

5.1 Multidimensional modelling with XML data

Context and main issues: On the one hand, XML data is organised according to a tree-like structure and the data of standard dimensions is also organised in the same way (i.e. dimensions with strict hierarchies [38]). Hence, the XML structure is used to logically represent the multidimensional structure of OLAP analysis data. On the other hand, several logical architectures were designed: ROLAP, MOLAP and HOLAP[8]. In a similar way what could be named "X-OLAP logical architectures" were defined.

The major issues are:

- The usage of a tree like structure to logically represent all multidimensional data;
- Absence of key-like structures to link data implying denormalisation of multidimensional data (i.e. as in Kimball's star schema [36]).

Detailed presentation of researches: Either in the case of data-centric XML documents or document-centric XML documents, several solutions were addressed.

Boussaid et al. [13] stores multidimensional instances together. Factual data as well as the associated dimensional data are stored together in the same XML element. This produces large volumes of data due to denormalisation. The opposite has also been conceived by splitting instances. In this case, factual data and dimensional data are stored in separate XML files (as in relational systems, using R-OLAP architectures, where factual data and dimensional data are stored in separate tables). Park et al. [52] use an index file with the equivalent of database foreign keys (not detailed in the article) to link factual and dimensional data in the different files.

Specific XML structures have also been used, for example, Wiwatwattana et al. [89,90] where the authors use "coloured XML trees" defined in [30]. Finally, others map XML source data to a multidimensional model, i.e. maintain source XML files and produce a mapping to designate factual and dimensional data within the XML files. It should be noted that this approach may be combined with the splitting instances approach when using views. This generates a virtual data mart (e.g. [45]).

[8] ROLAP, MOLAP, HOLAP : Relational, Multidimensional, Hybrid OLAP architectures (i.e. implementations).

List of unsolved problems: These approaches have several unsolved drawbacks:

- A structural issue;
- A space consumption issue;
- A query issue.

In terms of structure, the tree like structure of XML files is adapted to dimensions composed of classical hierarchical data (i.e. strict hierarchies [38]). Unfortunately other types of hierarchical relations are more complex to handle and these types generate summarisability issues. Mazón et al. [40] presents an overview of such issues and the different solutions that have been proposed so far.

In terms of space, Kimball's denormalised star schema [36] usually requires a lot more space due to the use of XML to represent multidimensional data (partly due to the UTF-8 or 16 encoding used in XML).

In terms of query, the use of XML as a logical architecture requires complex (multidimensional) queries expressed in languages such as XQuery. As XQuery is not adapted to multidimensional query expressions, some researches have proposed to extend XQuery with a grouping mechanism (see [8,12] in the next subsection). Moreover, current XML databases lack numerous optimisations that have been implemented in relational databases.

Associated to multidimensional modelling is the manipulation of the multidimensional structures in order to define OLAP analysis queries.

5.2 OLAP manipulation of XML data

Context and main issues: Very recently, some researches have focused on the problems related to complex data integration into warehouses according to two fields of research: firstly, adapting standard manipulation operations to handle XML data and secondly, adapting the OLAP aggregation principle for working on textual data rather than just numerical data.

Detailed presentation of researches: In order to express multidimensional queries on XML data, query languages such as the XML Query language (XQuery) [92] are used. However, these languages lack operators that would simplify multidimensional query expressions. As a first step, Beyer et al. [8] and

Bordawekar et al. [12], have specified a grouping mechanism for the XQuery language (similar to the SQL *Group By*) simplifying OLAP query expressions.

A recent approach offers some algebraic operators [25] based on the TAX [29]. However, this early proposal does not detail implementation issues neither possible optimisations.

For aggregating grouped data, the *XAggregation*, function defined by Wang et al. [85], aggregates XML structures. It is associated to the XOLAP framework [84] in order to specify more complex queries over XML cubes. Very recently, Wiwatwattana et al. [88] modified Gray's *cube* operator [23] to handle XML data in a flexible environment.

OLAP analyses are based on the aggregation of numerical indicators. The analysis of document-centric document data (i.e. textual data) requires more advanced analysis processing than a simple aggregation of XML structures. It is necessary to provide aggregation functions for textual contents, i.e. non numerical analysis indicators. As a first steps to a more global framework, aggregation functions adapted to textual data were defined. Park et al. [52] introduce such functions based on text mining techniques while Ravat et al. [67] specify a function based on information retrieval weighing techniques.

List of unsolved problems: Most relevant issues still open:

- Lack of a complete manipulation framework;
- Partial adaptation to text and complex data;
- More advanced aggregation functions.

Firstly, manipulation of XML data, lacks a complete adapted language for multidimensional structures and OLAP style analysis queries (such as in [2,24]). Secondly, although the OLAP aggregation process is undergoing changes, only a few researches try to handle textual data or complex data (i.e. numerical data that is more complex than counts or amounts such as perimeters, spectral data, etc.). Finally aggregation has been studied in terms of summarizability (see [40] for a recent overview) and now requires adaptation for the new data type possibilities (e.g. text).

5.3 XML Document Warehouse OLAP oriented

Context and main issues: The goal of this approach is to provide a document warehouse equivalent to a data warehouse. The major issue of this approach is the analysis of document contents. As document-centric XML documents are mainly composed of textual data, adapted analysis has to be provided.

Detailed presentation of researches: Park et al. [52] describe a framework for analysing document-centric XML documents within an XML environment. Dimensions are represented through XML's hierarchical structure and facts use the specific xFACT structure [45]. This work suggests the use of aggregation functions inspired by text mining techniques for the analysis of the text rich contents of the documents but lacks formal specification and implementation of the framework.

Ravat et al. [66] define a multidimensional conceptual model for the analysis of document-centric XML documents and associate a set of adapted manipulation operations to the model. The architecture rests on a combination of Relational OLAP and XML using specific SQL/XML [42].

The analysis of text-rich XML documents lack numerical measures, thus traditional aggregation functions do not operate. In order to allow aggregation over textual data, adapted aggregation functions are required (such as those suggested in [52]). Recently, an adapted aggregation function based on information retrieval weighing techniques [67] has been proposed. However, as these functions are complex, they still require optimisation.

List of unsolved problems: Most relevant issues still open:

- Complete manipulation algebraic language;
- Complete conceptual formal model;
- Performance enhancements.

Firstly, manipulation algebra linked to XML warehouses for OLAP style data warehouses can still be enhanced with more advanced manipulation operators (see [64] for some multidimensional operators). Secondly, XML data does not only consist of textual documents and as more data types will be handled within XML OLAP environments, there is a need for a generalised conceptual model that will have to take into account the specificities of the different data types. Finally, these data warehouses, as traditional ones, require optimisation in order to gradually solve performance issues.

5.4 XML for OLAP so far…

Overall summary: Providing an OLAP analysis framework for document contents is complex and only in its early years. Initial approaches created or adapted models in order to cope with XML structure (see multidimensional modelling with XML category in Table 4) but did not focus on the analysis environment. Others have modified the analysis environment to enable OLAP manipulation of XML documents, but were restricted to data-centric XML documents (see OLAP manipulation category in Table 4). Finally, recent researches consider both and focus on the analysis of document-centric XML document contents (see "Document OLAP" category in Table 4).

XML Warehouses for OLAP			
Categories		**Multidimensional modelling**	**Manipulation (Operator Type)**
Multidimensional Modelling with XML	[13,43]	XML snowflake schema - 1 XML file Measures associated with Dimensional Data	*Specific data mining operator (out of the scope of the OLAP framework)*
	[45,47]	Specific: xFACT - 1 XML View per fact (xFACT) - 1 XML View per Dimension (or dimension level)	
OLAP Manipulation	[8,12]		GroupBy Operator for XML Query Language (XQuery)
	[25]		Algebraic operators
	[84,85]		Aggregation of XML tree-like structures
	[88]		Extension of the Cube Operator [23] for XML Structured Data
Document OLAP	[52]	Specific (full XML): - 1 XML file per Dimension - 1 XML file per Fact (xFACT) - 1 index linking data in each XML file (not described)	OLAP Aggregation of XML Textual Data (not detailed)
	[6566,67,68]	Specific (ROLAP and XML): Conceptual OLAP Document Analysis Model (Galaxy)	OLAP Aggregation of XML Textual Data

Table 4 Summary of the characteristics of "XML OLAP" approaches

Analysis Framework Comparison: The initial researches that provide XML-based multidimensional modelling [13,43,45,47] can handle more complex data than previous proposals (i.e. complex data-centric XML documents). These researches consider the same approach as when analysing document-centric document metadata. However, they have the same drawback, i.e. analyses are limited to classical measures (e.g. simple counts). They can easily handle data-centric XML documents, but are limited when it comes to document-centric ones.

The researches that have focussed on OLAP manipulation [8,12,84,85,88] are all limited to data-centric XML documents. They cannot take into account the specificities of the more complex document-centric ones. However, all these researches provide valuable modifications to specific XML manipulation languages that will be necessary in the future.

Finally, the two approaches that offer a more global framework [52,65,66,67,68] are capable of analysing document-centric XML documents (both can also handle data-centric ones). In these proposals, the analysis capacities, although linked to algebraic operators [66], are still limited. So far, among all the complex data types only textual data is handled.

Unsolved issues: The major unsolved issues are:

- Performance issues;
- Lack of a logical translation from the conceptual level to the physical implementation;
- How to handle textual data.

Firstly, using XML as a multidimensional logical structure for data marts ("XML multidimensional modelling") has numerous consequences. Among these, XML databases and multidimensional XML database lack maturity compared to relational database management systems and numerous optimisations are still required to obtain reasonable processing time for OLAP queries.

Secondly, some researches have considered conceptual modelling adapted to the analysis of document-centric XML documents. The translation from the conceptual level to the logical one is incomplete in traditional warehouse architectures [69] and there is no exception for the XML OLAP environments. This is partly due to the fact that 1) XML environments (i.e. XML databases) still lack maturity; 2) specific optimisations have been neither specified nor implemented; and 3) there is no specification of a physical architecture for such systems.

Thirdly, analysing document contents raises two major issues: how to handle textual data and performance issues. Solutions such as text mining techniques could be applied, (see [75] for a recent overview). Moreover, data warehouses based on textual data will be slower than their numerical cousin, hence there is a necessity of increased performance. This research line could benefit from 10 years of optimisation using views and indexes in numerical data warehouses and in databases.

6 Finding a application-appropriate model

The four previous sections presented an overview of the different researches that mix XML and data warehouse technologies. When designing applications mixing these technologies, one may become easily lost with the numerous solutions. The goal of this section is to provide guidelines to choose which model (or architecture) is relevant depending on the application requirements and the available XML documents as data sources.

The four models (or architectures) that may be used have been presented in each of the four previous sections:

- XML data integration;
- XML data warehouses;
- XML document warehouses;
- XML warehouses for OLAP.

The question that has to be answered when designing such applications is: "what are the XML documents intended for?" The answer helps to describe the application requirements.

If the documents are intended to be read by users, the requirements are *information retrieval*-like. The idea is to query the document storage space and to retrieve a particular set of documents. When querying data-centric XML documents, the queries can be done on XML element content that act as columns in a database. In case of document-centric XML documents, the query will be done on a partial content of the document. For example, it is usual to reduce the content of textual documents to a set of keywords.

If the documents are intended to be analysed, there are three cases that depend on how the analysis is carried out. First is the case of the document that contains data for a *complementary analysis*. In this case, the data is to be associated to an already existing analysis framework (for example, XML files that contain additional geographical information that will be associated to country names stored in a data warehouse). In the second case, the document data will be used for *OLAP style analysis*. In this case, two alternatives exist: structure analysis and content analysis. In the third case an analysis engine is intended to be used (e.g. data mining, text mining, etc.).

Finally, if the document is supposed to be a common pivot language for information, there are two cases. First the document is an input for the *data integration* process, that is, the document is a common format for integrating data in a data warehouse. And second, the document is a *communication* means between heterogeneous decision support systems.

The Table 5 summarises the different solutions for the application requirements previously presented depending on the XML document type.

Application Requirements		Data Sources	XML data integration	XML data warehouses	XML Document Warehouses	XML Warehouses for OLAP
Information Retrieval		Data-centric or Document-centric			X	
Complementary Analysis		Data-centric (1)	X	X		
		Document-centric (2)	X	X		X
OLAP Analysis	Structure	Data-centric or Document-centric			X	
	Content	Data-centric (3)	X	X		
		Document-centric				X
Other Analyses		Data-centric		X		
		Document-centric			X	
Data Integration		Data-centric (4)	X	X		
		Document-centric (4)			X	X
Communication between DSS		Data-centric or Document-centric (5)	X	X	X	X

Table 5 the right model for the right application.

In some cases, the table provide several alternative solutions for a same application requirement and document type. These cases are highlighted by numbers in brackets. The following paragraphs discuss these alternatives:

(1) In case of complementary analysis with data-centric XML documents, the solution depends on how easily are converted the documents. If it is easy, then XML data integration should be considered. Otherwise, XML data warehouses should be preferred.

(2) When requirements are complementary analysis with document-centric documents, the solution depends on what should be analysed within the document. First of all, if the textual content of the documents is to be analysed, then XML warehouses for OLAP should be used. Otherwise, there are two alternatives. XML data integration should be used if the analysis information is easily extracted from the documents and if not, XML data warehousse should be considered.

(3) The case of analysing the content of data-centric XML documents is similar to case (1) and the same arguments hold.

(4) When requirements concern data integration, it is necessary to decide whether the document should be converted into another format or not. If conversion is possible, then an XML data integration architecture should be considered. Otherwise, if the conversion if too difficult or too costly, XML architectures should be considered; XML data warehouses when using data-centric documents and XML document warehouses when using document-centric XML documents.

(5) In case of communication between DSS, it should be noted that XML data integration architecture is fine when there is no consideration for OLAP analysis of the documents. Otherwise, one of the analysis solutions should be considered.

Finally when combining several requirements, it is possible to fall on incompatible solutions. In these cases, it should be noted that some solutions may be replaced by more general ones, although this will require extra modifications. In this way, XML data warehouses may be a more general alternative to XML data integration; and XML warehouses for OLAP are a more general alternative to both XML data warehouses and XML document warehouses.

It should also be noted that if semantic issues are the major requirement of the application, recently, Pérez *et al.* [58] have written a complementary survey on the integration of Web data and data warehousing. In this survey the fourth section particularly highlights the semantic issues However, it should be noted that their survey is information retrieval oriented. The reader is also invited to consult [51] for more precisions on semantic issues.

To summarise, XML data integration is used when a conversion of the data sources is possible; XML data warehouses are used when analysis is required and document-centric XML document contents are not used; XML document warehouses are used when no analysis is required; Finally, XML warehouses for OLAP are used when the content of document-centric XML documents has to be analysed.

7 Conclusion and challenges that still arise

XML has several advantages that have made this technology attractive to data warehouses. It is useful for defining analysis messages in case of interoperability (such as XMLA). As XML is a self describing format, it combines data and meta-data easing the definition of integration processes (ETL processes).

Finally, its expressive power allows the definition of OLAP structures (such as cubes). As a consequence XML warehousing simplifies early steps of data cleansing and enrichment but also the final steps of interactive analysis.

However, XML has also some limitations. Firstly: in terms of performance, such as query response time which is critical for OLAP applications. Indeed, these applications need to maintain the line of the though of the analyst while performing different analyses. Secondly in terms of complexity in managing links such as references (similar to foreign keys in relational databases). Indeed, these links require many value-based joins. And thirdly, in terms of input data types as XML documents may contain more than text and numbers (e.g. images). Moreover, XML documents are not limited to recording transactional data, thus the global environment is more general and more flexible but also a lot more complex to handle.

7.1 Summary

XML warehouses are an active research topic. The researches on this topic presented in this paper have been divided into four categories. These categories correspond to technical or conceptual solutions for implementing applications using XML and data warehouses. Each category was described and compared: integrating XML data in current OLAP systems (detailed in section 2); building data warehouses or document warehouses on top of XML data but with limited OLAP functionalities (see sections 3 and 4); and providing OLAP framework for XML warehouses, notably content warehouses (presented in section 5).

Depending on the type of warehouse, different application requirements may be met. As a consequence, the previous section (section 6) described the correspondence between application requirements and the categories in order to ease the selection of the appropriate model for the application.

In order to be more specific, this section continues with a comparative study of the different categories.

7.1.1 Global Comparative Study

A summary of the researches cited in this paper is presented in Table 6 where the different papers are organised according to the 4 central sections of this paper. The first conclusions that may be drawn are the fact that most works rest on data-centric XML documents. And despite the existence of numerous multidimensional models (see [78] for surveys) nearly all settle for Kimball's star schema [36].

	Sources (Document Type)		Data Integration				Data storage						OLAP Manipulation of data				Design	
							Conceptual			Logical								
	Data-centric XML Document	Document-centric XML Document	Logical (Federation)	Physical (with conversion into a non XML system)	Physical (native XML system, with no conversion)	Interoperability	Constellation Schema [36]	xFACT Schema [45]	Galaxy Schema [66]	Star Schema [36]	Snowflake Schema [36]	R-Cube Schema [57]	Algebraic Manipulation Operators	OLAP Aggregation Functions	Document Metadata Analysis	Document Textual Content Analysis	Design Method	Document Versions
XML Data Integration																		
[22,82,83]	X			X						X								
[59]	X			X						X								
[100]	X			X														
[50]	X			X						X								
[27]	X			X						X								
[79]	X			X						X								
[28]	X					X				X								
[48,49]	X					X				X								
[41]		X		X						X					X			
[44,5]		X		X						X					X			
[34]		X		X						X					X			
XML Data Warehouses																		
[31,53,55,99]	X		X							X								
[37]	X		X								X							
[60]	X				X													
[13]	X				X						X							
[6]	X				X					X								
[45,47]	X	X			X			X							X		X	
[72,73]		X			X					X					X		X	X
XML Document Warehouses																		
[56,57]		X	X									X						
[35]		X			X					X								
[80]		X		X						X					X			
XML Warehouses for OLAP																		
[8,12]	X				X					X			X					
[25]	X				X					X			X					
[84,85]	X				X					X			X					
[88]	X				X					X			X					
[52]		X			X			X						X*	X	X		
[65,66,67,68]	X	X		X	X		X		X				X	X	X	X	X	
Notes: *=no formal specification																		

Table 6 Summary of the different scientific contributions

In Table 6, two types of physical integrations are distinguished: the physical integration of XML data by converting the data into the local system (non XML) and integrating XML data in a native XML system. Some researches offer both possibilities but use the native XML system only to store the XML data to allow the user to consult it. In those cases, the analysis data, i.e. the data mart, is stored in a non XML system (e.g. a standard ROLAP system). On the contrary, researches in the category of XML warehouses for OLAP combine both systems for flexibility and performance.

Surprisingly enough, few academic research address querying and analysis issues. This may be due to the evolving XML query languages. For example, only recently do researches have access to efficient XQuery frameworks. And finally, there are even fewer design methods and proposals that handle document evolution.

7.1.2 Analysis Framework Comparison

Data warehousing is usually used to perform OLAP style analyses on data. This subsection summarises the analysis capabilities (in terms of on-line analytical processing) provided by the four different approaches, using the sample XML documents presented in introduction (see Fig. 3).

1) Within the *XML data integration* category, the proposals that operate on data-centric XML documents allow the analysis of our sample data-centric document and those that operate on document-centric ones allow the analysis of the meta data of our sample document-centric document (see Fig. 6 and Fig. 7 for illustrations). The principle is the same for both approaches, elements of interest are extracted from the XML documents, converted into the data warehouse format and loaded into the warehouse.

2) Using *XML data warehouses*, the federation approach provides only partial analysis of the data-centric XML document content (see Fig. 11). And this is only in the case of factual data being the XML data. The other approaches can provide the analysis of our sample data-centric XML document (see Fig. 6). The researches of this category that focus on document-centric XML documents create an XML data warehouse from metadata extracted from these documents. Using our sample document-centric XML document, the analysis provided is similar to the one presented in Fig. 7. However, document contents are systematically discarded.

3) In the *XML document warehouses* category, contextualisation approaches provide no document analysis. Only one work provides a metadata analysis of document-centric XML documents. With our

sample document, this analysis is similar to the one presented in Fig. 7. However, document contents are still discarded.

4) Finally, in *XML warehouses for OLAP*, the researches that focus on manipulation are still in an early stage of research, because XML query languages are still young. Using our sample data-centric XML document, the approaches provide the manipulation to express the associated sales analysis (Fig. 6). However, document-centric XML documents are not taken into account. On the contrary, the approaches that focus on document-centric XML documents can use the data-centric (respectively, the document-centric) sample XML document to provide the associated sales analysis (respectively the metadata analysis). These researches can avoid discarding the document content and provide a new kind of analysis: analysing the document-centric XML document content (see Fig. 14).

7.1.3 General comments

Despite numerous conceptual models, there is no widely agreed upon conceptual model for representing data cubes [69] (data mart multidimensional structures). Moreover the link between the data warehousing conceptual and logical levels is not yet complete. Similarly, the same issues are present in XML warehousing. However, [22] defines a link between the XML logical level and the DFM (Dimensional Fact Model [21]) and [60] does the same with a star schema [36]. Finally, as XML systems are not mature enough, the physical level is far from being specified apart from a few research prototypes. However, recent database and data warehouse advances that use XML (such as [42,90]) should allow the design of new applications.

Handling XML documents within an analysis framework requires also handling content and structure evolution linked to document evolution but also schema versioning as analysis requirements may evolve as documents do. Analysing document-centric XML documents implies manipulating and summarising large quantities of text. Finally, XML document warehouses OLAP oriented fall into the category of large warehouses, they require adapted processing and optimisation as would very large databases.

7.2 What could future trends be

In addition to the open issues stated at the end of the previous sections, we have identified three major topics for future research in this domain: 1) design approaches for XML associated to data warehousing; 2) Storage issues for XML decisional data; and 3) Analysis restitution. These topics should be observed carefully when designing applications as research should evolve. Each is detailed in the following paragraphs and the last one states other research possibilities.

7.2.1 From a methodological point of view

The first question that could be asked is how to model XML OLAP environments? There is a need for models to be able to represent multidimensional XML data and the associated structures in order to perform OLAP. Some solutions have been proposed, but as detailed previously, these solutions are specific or still in their early years of research.

Design approaches are not sufficient and most of them still require to be completed for traditional warehousing systems [69]. They also need to be extended in order to handle specific characteristics of document-centric XML documents (a first proposal has been introduced in [68]). Moreover, if the problem is taken from the modelling side, modified OLAP concepts will require an adapted tomalism, thus an adapted design approach. Currently approaches such as goal oriented requirement engineering [70] and model driven engineering [39] are being developed. Adaptations to XML context could be considered.

Globally speaking, several issues may be considered:

- At the *conceptual level*, how to simplify the dialog with decision makers while still being able to represent the specificities that have to be modelled by an XML OLAP system? There is a need for models to be able to represent multidimensional XML data and the associated structures in order to perform OLAP.
- At *logical level*: How to support on-line analysis associated to XML? Is the classic ROLAP architecture allied with the uprising SQL/XML technologies [42] enough? It should be noted that at this level, as stated before, the logical translation between conceptual and physical levels is not complete yet [69].

– At *physical level*: is the definition of specifically adapted ETL processes required for handling XML documents? If so, it would be possible to benefit from pre-existing environments (e.g. [81]) and extend them. Moreover, would specific XML transformation processes adapted to XML documents be useful?

7.2.2 Storing and optimising XML data for OLAP

Along with the open issues stated in [69], many challenges still remain. XML documents have specific characteristics [66]. Are the basic physical models of traditional warehouses expressive enough to represent these characteristics? As XML documents characteristics differ depending on whether they are data-centric or document-centric, should there be different physical models? Different storage solutions exist nowadays. On may find content warehouses that manage documents (such as Xyleme Server), traditional DBMS extended with the management of XML data (such as Oracle XMLDB[9]) and pure XML DBMS such as Tamino[10].

Document-centric XML documents are also concerned with document evolution issues. Existing techniques could be used such as handling data warehouse versions (see [97] for references) or deltas (e.g. [10]). Recently in [73,74] the authors tackle the problem of dynamic documents. In a near future, this could lead to "XML warehouse versioning".

Moreover, XML data is large due to the usage of UTF encoding. However, numerous optimisations could be considered such as compression (e.g. compressed cubes) or indexes. Pre-computations and pre-aggregations would also speed up the system. But, do materialised views require to be extended? If so, should they be defined in a specific way? For example, what would an "XML OLAP cube" correspond to? what about an "XML iceberg cube"[11]? Etc.

7.2.3 Analysis based on XML documents

Firstly analysis must be fast. In order to maintain a line of thought of decision makers, it is commonly accepted that OLAP systems require answering analysis queries within approximately 10 seconds. Optimising of the physical storage would increase the response time and, as in traditional systems, indexes

[9] XMLDB, module of Oracle Database (since Oracle Database *10g*) http://www.oracle.com/database/index.html
[10] Tamino: from SoftwareAG, http://www.softwareag.com/corporate/products/wm/default.asp

could also be used (bitmap, trees or their numerous variants, etc.), but at the cost of space and complex update processing.

In traditional OLAP systems, dashboards and pivot tables (bi-dimensional tables) restitution modes are widely used. But when displaying the result of document-centric XML document analysis, are those analysis interfaces enough?

In the past ten years, numerous researches have focused on how to specify OLAP queries. These researches base themselves on the definition of algebras. However, when performing XML OLAP, do all the previously specified operators still make sense? Is there a need for specific XML operators or XML aggregation functions? It should be noted that some works argue that OLAP analyses on document-centric XML document contents require adapted aggregation functions to handle textual contents. Moreover, XML views require adaptations to be used as "materialised XML views" [4] and the previously stated new aggregation functions will require materialised views [101] for the optimisation of OLAP queries.

7.2.4 XML in DSS so far…

Fankhauser, [18], stated that XML technology was mature enough to allow the design of challenging text-mining applications. We believe that adapting the OLAP framework for XML opens a new window on decision support. This new flexible architecture will allow the combination of different analysis techniques such data mining or text mining while maintaining the widely used OLAP analysis framework.

References

1. S. Abiteboul, S. Cluet, G. Ferran, M.-C. Rousset, The Xyleme project, j. Computer Networks, 39(3), Elsevier, 2002, pp. 225–238.
2. R. Agrawal, A. Gupta, S. Sarawagi, Modeling Multidimensional Databases, in 13th Int. Conf. on Data Engineering (ICDE), IEEE Computer Society, 1997, pp. 232–243.
3. S. Amer-Yahia, S. Cho, D. Srivastava, Tree Pattern Relaxation, in proc. 8th Intl. Conf. on Extending Database Technology (EDBT), LNCS 2287, Springer, 2002, pp. 496–513.

[11] see [19] for more details on iceberg cubes.

4. A. Arion, V. Benzaken, I. Manolescu, Y. Papakonstantinou, Structured Materialized Views for XML Queries, 33rd Intl. Conf. on Very Large Data Bases (VLDB), ACM, 2007, pp. 87–98.

5. N. Aussenac-Gilles, J. Mothe, Ontologies as Background Knowledge to Explore Document Collections, in RIAO 2004 conf. proceedings: coupling approaches, coupling media and coupling languages for information retrieval, 2004, pp. 129–142.

6. X. Baril, Z. Bellahsène, Designing and Managing an XML Warehouse, in Chaudhri A.B., Rashid A., Zicari R. (Eds.), XML Data Management: Native XML and XML-Enabled Database Systems, Addison Wesley, 2004, pp. 455–474.

7. P. A. Bernstein, H. Ho, Model Management and Schema Mappings: Theory and Practice. Tutorial of the 33rd Int. Conf. on Very Large Data Bases (VLDB), 2007, pp. 1439–1440.

8. K.S. Beyer, D.D. Chamberlin, L.S. Colby, F. Ozcan, H. Pirahesh, Y. Xu, Extending XQuery for Analytics, in Proceedings of the ACM Int. Conf. on Management of Data (SIGMOD), ACM Press, 2005, pp. 503–514.

9. S.S. Bhowmick, W.K. Ng, S.K. Madria, Web Schemas in WHOMEDA, in 3rd ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2000, pp. 17–24.

10. S.S. Bhowmick, S.K. Madria, W.K. Ng, Detecting and Representing Relevant Web Deltas in WHOWEDA, IEEE Transactions on Knowledge and Data Engineering (TKDE), 15(2), IEEE Computer Society, 2003, pp. 423–441.

11. J. Bleiholder, F. Naumann, Data Fusion, ACM Computing surveys, 41(1), ACM Press, 2008.

12. R. Bordawekar, C.A. Lang, Analytical processing of XML documents: opportunities and challenges, SIGMOD Record 34(2), ACM Press, 2005, pp.27–32.

13. O. Boussaid, R.B. Messaoud, R. Choquet S. Anthoard, X-Warehousing: An XML-Based Approach for Warehousing Complex Data, in 10th East European Conf. on Advances in Databases and Information Systems (ADBIS), LNCS 4152, Springer, 2006, pp. 39–54.

14. G. Colliat, OLAP, relational, and multidimensional database systems, ACM SIGMOD Record, 25(3), ACM Press, 1996, pp. 64–69.

15. Dublin Core Metadata Initiative (DCMI): Dublin Core Metadata Element Set, Version 1.1, ISO Standard 15836, from http://dublincore.org/documents/dces/ (sept. 2008).

16. F-X. Dudouet, I. Manolescu, B. Nguyen, P. Senellart, XML Warehousing Meets Sociology, in *IADIS Intl. Conf. WWW/Internet* (*ICWI*), IADIS Press, 2005.

17. M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, J.D. Ullman, Computing Iceberg Queries Efficiently, in 24th Int. Conf. on Very Large Data Bases (VLDB), Morgan Kaufmann, 1998, pp. 299–310.

18. P. Fankhauser, T. Klement, XML for Data Warehousing Chances and Challenges, (extended abstract), in 5th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 2737, Springer, 2003, pp. 1–3.

19. L. Findlater, H.J. Hamilton, Iceberg-cube algorithms: An empirical evaluation on synthetic and real data, in j. of Intelligent Data Analysis, 7(2), IOS Press, 2003, pp. 77–97.

20. N. Fuhr, K. Großjohann, XIRQL: A Query Language for Information Retrieval in XML Documents, in 24th int. ACM conf. on Research and development in information retrieval (SIGIR), ACM Press, 2001, pp.172–180.

21. M. Golfarelli, D. Maio, S. Rizzi, The Dimensional Fact Model: a Conceptual Model for Data Warehouses, invited paper, in Int. J. of Cooperative Information Systems (ijCIS), 7(2&3), 1998, 215–247.

22. M. Golfarelli, S. Rizzi, B. Vrdoljak, Data Warehouse Design from XML Sources, in 4th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2001, pp. 40–47.

23. J. Gray, A. Bosworth, A. Layman, H. Pirahesh, Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total, in 12th Int. Conf. on Data Engineering (ICDE), IEEE Computer Society, 1996, pp. 152–159.

24. M. Gyssens, L.V.S. Lakshmanan, A Foundation for Multi-dimensional Databases, in 23rd Int. Conf. on Very Large Data Bases (VLDB), Morgan Kaufmann, 1997, pp. 106–115.

25. M. Hachicha, H. Mahboubi, J. Darmont, Expressing OLAP operators with the TAX XML algebra, 3rd Intl. Workshop on DAtabase Technologies for handling XML Information on the Web (DataX), ACM, 2008, pp. 61–66.

26. V. Harinarayan, A. Rajaraman, J.D. Ullman, Implementing Data Cubes Efficiently, in ACM Int. Conf. on Management of Data (SIGMOD), ACM Press, 1996, pp. 205–216.

27. S-M. Huang, C-H. Su, The Development of an XML-Based Data Warehouse System, in 3rd int. conf. on Intelligent Data Engineering and Automated Learning (IDEAL), LNCS 2412, Springer, 2002, pp. 206–212.

28. W. Hümmer, A. Bauer, G. Harde, XCube: XML for data warehouses, in 6th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2003, pp. 33–40.

29. H.V. Jagadish, L.V.S. Lakshmanan, D. Srivastava, K. Thompson, TAX: A Tree Algebra for XML, 8th Intl. Workshop on Database Programming Languages (DBPL), LNCS 2397, Springer, 2001, pp. 149–164.

30. H.V. Jagadish, L.V.S. Lakshmanan, M. Scannapieco, D. Srivastava, N. Wiwatwattana, Colorful XML: One Hierarchy Isn't Enough, in Proc. of the ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD), ACM, 2004, pp. 251–262.

31. M.R. Jensen, T.H. Møller, T.B. Pedersen, Specifying OLAP Cubes On XML Data, in 13th Int. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, 2001, pp.101–112.

32. M.R. Jensen, T.H. Møller, T.B. Pedersen, Specifying OLAP Cubes on XML Data, in journal of Intelligent Information Systems, 17(2-3), Springer Netherlands, 2001, pp. 255–280.

33. J. Kamps, M. Marx, M. de Rijke, B. Sigurbjörnsson, Best-Match Querying from Document-Centric XML, in Proc. 7th Int. Workshop the Web and Databases (WebDB ’04), 2004, pp. 55–60.

34. S. Keith, O. Kaser, D. Lemire, Analyzing Large Collections of Electronic Text Using OLAP, in *APICS 29th Conf. in Mathematics, Statistics and Computer Science*, Acadia University, 2005, pp. 17–26.

35. K. Khrouf, C. Soulé-Dupuy, A Textual Warehouse Approach: A Web Data Repository, in M. Mohammadian (Ed.), Intelligent Agents for Data Mining and Information Retrieval, Idea Publishing Group, 2004, pp. 101–124.

36. R. Kimball, The Data Warehouse Toolkit. John Wiley, 1996; 2nd edition: R. Kimball, M. Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, Wiley and Sons, Chichester, 2003.

37. Y. Li, A. An, Representing UML Snowflake Diagram from Integrating XML Data Using XML Schema, in Int. Workshop on Data Engineering Issues in E-Commerce (DEEC), IEEE Computer Society, 2005, pp. 103–111.

38. E. Malinowski, E. Zimányi, Hierarchies in a multidimensional model: From conceptual modeling to logical representation, in j. Data & Knowledge Engineering (DKE), 59(2), Elsevier, 2006, pp. 348–377.

39. J.-N. Mazón, J. Trujillo, An MDA approach for the development of data warehouses, Decision Support Systems (DSS), 45(1), Elsevier, 2008, pp. 41–58.

40. J.-N. Mazón, J. Lechtenbörger, J. Trujillo, A survey on summarizability issues in multidimensional modelling, in Data & Knowledge Engineering, Elsevier, 2009 (accepted manuscript), doi: 10.1016/j.datak.2009.07.010

41. C. McCabe, J. Lee, A. Chowdhury, D.A. Grossman, O. Frieder, On the design and evaluation of a multi-dimensional approach to information retrieval, in 23rd Int. ACM Conf. on Research and Development in Information Retrieval (SIGIR), ACM Press, 2000, pp. 363–365.

42. J. Melton, S. Buxton, Querying XML, XQuery, XPath, and SQL/XML in Context, Elsevier Inc., 2006.

43. R.B. Messaoud, O. Boussaid, S. Rabaséda, A new OLAP aggregation based on the AHC technique, in 7th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2004, pp. 65–72.

44. J. Mothe, C. Chrisment, B. Dousset, J. Alau, DocCube: Multi-dimensional visualisation and exploration of large document sets, J. of the American Society for Information Science and Technology (JASIST), 54(7), Wiley Periodicals, 2003, pp. 650–659.

45. V. Nassis, R. Rajugan, T.S. Dillon, J.W. Rahayu, Conceptual Design of XML Document Warehouses, in 6th int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 3181, Springer, 2004, pp. 1–14.

46. V. Nassis, R. Rajagopalapillai, T.S. Dillon, J.W. Rahayu, Conceptual and Systematic Design Approach for XML Document Warehouses, Int. J. of Data Warehousing & Mining (ijDWM), 1(3), Idea Group Publishing, 2005, pp. 63–87.

47. V. Nassis, T.S. Dillon, R. Rajagopalapillai, J.W. Rahayu, An XML Document Warehouse Model, in 11th Int. Conf. on Database Systems for Advanced Applications (DASFAA), LNCS 3882, Springer, 2006, pp. 513–529.

48. T.B. Nguyen, A.M. Tjoa, O. Mangisengi, Meta Cube-X: An XML Metadata Foundation for Interoperability Search among Web Data Warehouses, in 3rd Int. Workshop on Design and Management of Data Warehouses (DMDW), CEUR Workshop Proceedings 39, CEUR-WS.org, 2001, p. 8.

49. T.B. Nguyen, A.M. Tjoa, O. Mangisengi, MetaCube XTM: A Multidimensional Metadata Approach for Semantic Web Warehousing Systems, in 5th int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 2737, Springer, 2003, pp.76–88.

50. T. Niemi, M. Niinimäki, J. Nummenmaa, P. Thanisch, Constructing an OLAP cube from distributed XML data, in 5th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2002, pp.22–27.

51. N.F. Noy, Semantic Integration: A Survey Of Ontology-Based Approaches, in SIGMOD Record 33(4), ACM, 2004, pp. 65–70.

52. B-K. Park, H. Han, I-Y. Song, XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses, in 7th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 3589, Springer, 2005, pp. 32–42.

53. D. Pedersen, K. Riis, T.B. Pedersen, XML-Extended OLAP Querying, in 14th Int. Conf. on Scientific and Statistical Database Management (SSDBM), IEEE Computer Society, 2002, pp.195–206.

54. D. Pedersen, T.B. Pedersen, Achieving adaptivity for OLAP-XML federations, in Proc. of the 6th ACM intl. workshop on Data warehousing and OLAP (DOLAP), ACM, 2003, pp. 25–32.

55. D. Pedersen, J. Pedersen, T.B. Pedersen, Integrating XML Data in the TARGITOLAP System, industrial paper in 20th Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, 2004, pp. 778–781.

56. J.M. Pérez, R. Berlanga Llavori, M.J. Aramburu, T.B. Pedersen, A relevance-extended multi-dimensional model for a data warehouse contextualized with documents, in 8th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2005, pp. 19–28.

57. J.M. Pérez, R. Berlanga Llavori, M.J. Aramburu, T.B. Pedersen, Contextualizing data warehouses with documents, in *Decision Support Systems* (*DSS*), Elsevier, 45(1), 2008, pp. 77–94.

58. J.M. Pérez, R. Berlanga, M.J. Aramburu, T.B. Pedersen, Integrating Data Warehouses with Web Data: A Survey, in IEEE Transactions on Knowledge and Data Engineering (TKDE), 20(7), 2008, pp. 940–955.

59. J. Pokorný, Modelling Stars Using XML. 4th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2001, pp.24–31.

60. J. Pokorný, XML Data Warehouse: Modelling and Querying, in 5th Int. Baltic Conf. on Databases and Information Systems (BalticDB&IS), vol.1, Institute of Cybernetics (Tallin Technical University), 2002, pp. 267–280.

61. J.M. Ponte, W.B. Croft, A language modeling approach to information retrieval, 21st Intl. Conf. on Research and Development in Information Retrieval (SIGIR), ACM Press, 1998, pp. 275–281.

62. Y. Qi, K.S. Candan, J. Tatemura, S. Chen, F. Liao, Supporting OLAP operations over imperfectly integrated taxonomies, in Proc. of the ACM Intl. Conf. on Management of Data (SIGMOD), ACM, 2008, pp. 875–888.

63. E. Rahm, P.A. Bernstein, A survey of approaches to automatic schema matching, VLDB Journal, 10(4), Springer-Verlag, 2001, pp. 334–350.

64. M. Rafanelli, Operators for Multidimensional Aggregate Data, in M. Rafanelli (Ed.), Multidimensional databases: Problems and solutions, Idea Group Publishing, 2003, pp. 116–165.

65. F. Ravat, O. Teste, R. Tournier, OLAP Aggregation Function for Textual Data Warehouse, in 9th Int. Conf. on Enterprise Information Systems (ICEIS), vol. DISI, INSTICC Press, 2007, pp. 151–156.

66. F. Ravat, O. Teste, R. Tournier, Z. Zurfluh, A Conceptual Model for Multidimensional Analysis of Documents, in 26th Int. Conf. on Conceptual Modeling (ER), LNCS 4801, Springer, 2007, pp. 550–565.

67. F. Ravat, O. Teste, R. Tournier, Z. Zurfluh, Top_Keyword: An Aggregation Function for Textual Document OLAP, in 10th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWak), LNCS 5182, Springer, 2008, pp. 55–64.

68. F. Ravat, O. Teste, R. Tournier, Z. Zurfluh, Designing and Implementing OLAP Systems from XML Documents, in Kozielski, Stanislaw; Wrembel, Robert (Eds.), Annals of Information Systems vol. 3, special issues on New Trends in Data Warehousing and Data Analysis, Springer, 2008, pp. 295–315 (to appear).

69. S. Rizzi, A. Abelló, J. Lechtenbörger, J. Trujillo, Research in data warehouse modeling and design: dead or alive? in 9th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2006, pp.3–10.

70. C. Rolland, Towards Engineering Purposeful Systems: A Requirements Engineering Perspective, 19th Intl. Conf. on Database and Expert Systems Applications (DEXA), LNCS 5181, Springer, 2008, pp. 1–4.

71. L.I. Rusu, J.W. Rahayu, D. Taniar, On Building XML Data Warehouses, in 5th Int. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL), LNCS 3177, Springer, 2004, pp. 293–299.

72. L.I. Rusu, J.W. Rahayu, D. Taniar, A Methodology for Building XML Data Warehouses, Int. J. of Data Warehousing & Mining (ijDWM), 1(2), Idea Group Publishing, 2005, pp. 23–48.

73. L.I. Rusu, J.W. Rahayu, D. Taniar, Warehousing Dynamic XML Documents, in 8th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 4081, Springer, 2006, pp.175–184.

74. L.I. Rusu, J.W. Rahayu, D. Taniar, Storage Techniques for Multi-versioned XML Documents, in 13th Int. Conf. on Database Systems for Advanced Applications (DASFAA), LNCS 4947, Springer, 2008, pp. 538–545.

75. A. Stavrianou, P. Andritsos, N. Nicoloyannis, Overview and semantic issues of text mining, SIGMOD Record 36(3), ACM Press, 2007, pp. 23–34.

76. D. Sullivan, Document Warehousing and Text Mining, Wiley John & Sons, 2001.

77. X. Tan, D.C. Yen, X. Fang, Web warehousing: Web technology meets data warehousing, Technology in Society (TechSoc), 25(1), Elsevier, 2003, pp. 131–148.

78. R Torlone, Conceptual Multidimensional Models, in Multidimensional Databases: Problems and Solutions, Maurizio Rafanelli (Ed.), Idea Publishing Group (IGP), 2003, pp. 69–90.

79. F.S.C Tseng, XML-Based Heterogeneous Database Integration For Data Warehouse Creation, in 9th Pacific Asia Conf. on Information Systems (PACIS), 2005, pp. 590–603.

80. F.S.C. Tseng, A.Y.H. Chou, The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence, J. of Decision Support Systems (DSS), 42(2), Elsevier, 2006, pp. 727–744.

81. P. Vassiliadis, A. Simitsis, P. Georgantas, M. Terrovitis, S. Skiadopoulos, A generic and customizable framework for the design of ETL scenarios, J. of Information Systems (IS), 30(7), 2005, pp. 492–525.

82. B. Vrdoljak, M. Banek, S. Rizzi, Designing Web Warehouses from XML Schemas, in 5th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 2737, Springer, 2003, pp.89–98.

83. B. Vrdoljak, M. Banek, Z. Skocir, Integrating XML Sources into a Data Warehouse, in 2nd int. Workshop on Data Engineering Issues in E-Commerce and Services (DEECS), LNCS 4055, Springer, 2006, pp. 133–142.

84. H. Wang, J. Li, Z. He, H. Gao, Xaggregation: Flexible Aggregation of XML Data, in 4th int. conf. on Advances in Web-Age Information Management (WAIM), LNCS 2762, Springer, 2003, pp.104–115.

85. H. Wang, J. Li, Z. He, H. Gao, OLAP for XML Data, in 5th Int. Conf. on Computer and Information Technology (CIT), IEEE Computer Society, 2005, pp.233–237.

86. WhoWeDa (WareHouse Of WEb Data), from the Web Warehousing & Mining Group (http://mandolin.cais.ntu.edu.sg/~whoweda/index.htm), june 2009.

87. WhoWeDa: Warehousing of Web Data, from http://mandolin.cais.ntu.edu.sg/~whoweda/.

88. N. Wiwatwattana, H.V. Jagadish, L.V.S. Lakshmanan, D. Srivastava, X^3: A Cube Operator for XML OLAP, in IEEE 23rd Int. Conf. on Data Engineering (ICDE), IEEE Computer Society, 2007, pp. 916–925.

89. N. Wiwatwattana, H.V. Jagadish, A Query Processing Architecture for an XML Data Warehouse, in Proc. 24th Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, 2008, pp. 1513–1515.

90. N. Wiwatwattana, XML Data Warehousing: Modeling, Design and Analysis. VDM Verlag, 2008.

91. World Wide Web Consortium (W3C) XML, eXtensible Markup Language 1.0 (4th Edition), W3C Recommendation 16 August 2006 from http://www.w3.org/TR/xml/

92. World Wide Web Consortium (W3C) XQuery 1.0, XML Query Language, W3C Recommendation 23 January 2007 (current working draft: XQuery 1.1 July 2008), from http://www.w3.org/TR/xquery/

93. World Wide Web Consortium (W3C), XML Path Language (XPath) 2.0, W3C Recommendation 23 January 2007 from http://www.w3.org/TR/xpath20/

94. World Wide Web Consortium (W3C), W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures, W3C Candidate Recommendation 30 April 2009.

95. World Wide Web Consortium (W3C), W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Candidate Recommendation 30 April 2009.

96. World Wide Web Consortium (W3C), XML Transformations (XSLT) version 2.0, W3C Recommendation 23 january 2007 from http://www.w3.org/TR/xslt20/

97. R. Wrembel, A Survey on Managing the Evolution of Data Warehouses, Intl. J. of Data Warehousing and Mining (ijDWM), IGI Publishing, 5(2), 2009, pp. 24–56.

98. Xyleme server from http:// www.xyleme.com.

99. X. Yin, T.B. Pedersen, Evaluating XML-extended OLAP queries based on a physical algebra, in 7th ACM Int. Workshop on Data Warehousing and OLAP (DOLAP), ACM Press, 2004, pp. 73–82.

100. J. Zhang, T.W. Ling, R.M. Bruckner, A.M. Tjoa, Building XML Data Warehouse Based on Frequent Patterns in User Queries, in 5th Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK), LNCS 2737, Springer, 2003, pp.99–108.

101. Y. Zhuge, H. Garcia-Molina, Graph Structured Views and Their Incremental Maintenance, in Proc. of the 14th Intl. Conf. on Data Engineering (ICDE), IEEE Computer Society, 1998, pp. 116–125.