Integrating Data Warehouses with Web Data: A Survey

Juan Manuel Pérez, Rafael Berlanga, María José Aramburu, and Torben Bach Pedersen, *Member*, *IEEE*

Abstract—This paper surveys the most relevant research on combining Data Warehouse (DW) and Web data. It studies the XML technologies that are currently being used to integrate, store, query, and retrieve Web data and their application to DWs. The paper reviews different DW distributed architectures and the use of XML languages as an integration tool in these systems. It also introduces the problem of dealing with semistructured data in a DW. It studies Web data repositories, the design of multidimensional databases for XML data sources, and the XML extensions of OnLine Analytical Processing techniques. The paper addresses the application of information retrieval technology in a DW to exploit text-rich document collections. The authors hope that the paper will help to discover the main limitations and opportunities that offer the combination of the DW and the Web fields, as well as to identify open research lines.

Index Terms—Data warehouse repository, XML/XSL/RDF.

1 Introduction

The Web is nowadays the world's largest source of information. It has brought interoperability to a wide range of different applications. This success has been possible thanks to XML-based technology [1], which provides a means of information interchange between applications, as well as a semistructured data model for integrating information and knowledge.

Information Retrieval (IR) [2] is also playing an important role in the Web, since it has enabled the development of useful resource discovery tools (e.g., Web search engines). Relevance criteria based on both textual contents and link structure have been shown to be very useful for effectively retrieving text-rich documents. Recently, information extraction techniques have been applied to detect and query the factual data contained in the documents (e.g., Question and Answering Systems). Finally, and more recently, the Web has been enriched with semantic annotations (e.g., RDF and OWL formats), allowing the retrieval and analysis of its contents in a more effective way in the near future.

During recent years, there has also been a large interest in Data Warehouse (DW) [3] and OnLine Analytical Processing (OLAP) [4] technologies. A DW system stores historical data integrated and prepared for being analyzed by OLAP and other tools. Many companies satisfy their needs for strategic information by applying these technologies to their structured databases.

The goal of this paper is to review how DW and Web technologies are being combined by current research efforts. From our point of view, the research in this field follows three main lines:

1. **The use of XML technology as an integration tool in distributed DW systems.** This research line includes work focused on both XML formats for exchanging multidimensional data and metadata [5], [6], [7] and new architectures that apply these XML languages and other XML-related technologies (e.g., XML query languages and transformation sheets) to integrate distributed heterogeneous DW systems [8], [9], [10], [11], [12], [13].
2. **The development of DWs for semistructured data.** The second research line covers the work on building warehouses for semistructured XML data (i.e., data-centric XML collections). We organize these papers into three groups:
   - The first group is oriented toward the construction of XML [14] or, more generally, Web document repositories [15]. They mainly address the efficient acquisition, storage, query, change control, and schema integration of data gathered from Web sources. However, these papers do not propose any mechanisms for analyzing these data.
   - The second group of papers is aimed at the design of multidimensional databases for XML data sources [16], [17], [18]. They study the problem of physically integrating XML data sources in a DW and propose different techniques to design the analysis schema, starting

- *J.M. Pérez and R. Berlanga are with the Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, Campus de Riu Sec, E-12071 Castelló de la Plana, Spain. E-mail: {JuanMa.Perez, berlanga}@lsi.uji.es.*
- *M.J. Aramburu is with the Departamento de Ingeniería y Ciencia de los Computadores, Universitat Jaume I, Campus de Riu Sec, E-12071 Castelló de la Plana, Spain. E-mail: aramburu@icc.uji.es.*
- *T.B. Pedersen is with the Department of Computer Science, Aalborg University, Selma Lagerlofsvej 300, DK-9220 Aalborg Ø, Denmark. E-mail: tbp@cs.aau.dk.*

Manuscript received 15 May 2007; revised 31 Oct. 2007; accepted 10 Dec. 2007; published online 21 Dec. 2007.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2007-05-0212.
Digital Object Identifier no. 10.1109/TKDE.2007.190746

```
<order id=''X123''>
   <customer account="10">
      <name>Thomas Edison</name>
      <telephone>123456123</telephone>
   ...
   </customer>
   <items>
      <item id=''CD1''>
         <name>cd player</name>
         <unitprice>60.6</unitprice>
         <quantity>1</quantity>
      </item>
   ...
   </items>
</order>
```

Fig. 1. Example data-centric XML document.

```
<business_newspaper date=''Dec.1,1998''>
<economy>
<article>
<headline>Financial Crisis Hits Southeast Asian
Market</headline>
<paragraph>
The financial crisis in Southeast Asian countries,
has mainly affected companies in the food
market sector. Particularly, Chicken SPC Inc. has
reduced total exports to $1.3 million during this
half of the year from $10.1 million in 1997.
</paragraph>
<paragraph> ...
</article> ...
</economy> ...
</business_newspaper>
```

Fig. 2. Example document-centric XML document.

from the DTDs provided by the data sources. Once the XML data is loaded into the DW, traditional OLAP techniques can be used to analyze and query the data.

- The third group of papers argues that since Web data is highly dynamic, the integration of the XML data sources at the logical level is a better option [19], [20]. They propose to extend traditional OLAP techniques to analyze online XML data, allowing the execution of OLAP operations on data contained in external XML sources. Nevertheless, their contributions only deal with highly structured XML data, and they are not suitable for exploiting the information described by document-centric XML collections.

3. **DWs and document-centric XML collections.** In the third research line, we consider the work that deals with unstructured data (i.e., document-centric XML collections) in DW systems. These papers combine DW and IR technologies in two different ways:

   - The first group of papers propose to apply multidimensional databases to implement IR systems [21], [22], that is, they use OLAP data cubes for querying a document collection. These systems apply OLAP operations at the document level and do not allow analysts to query the factual data described in the textual contents of the documents.
   - The second group of papers present the so-called contextualized warehouse [23], which is a new kind of decision support system that extends the architecture of the DW by adding a document repository. They propose a new type of OLAP cube where each fact is related to the set of documents that describe the dimension values that characterize the fact. IR queries are evaluated over the document collection in order to rank the facts of the cubes.

This paper covers all of these three research lines and is organized as follows: In Section 2, we introduce XML and Web technology. Section 3 summarizes the main concepts of DW and OLAP systems. In Section 4, we review the work on XML-based DW integration (research line 1). First, some XML formats to express DW data and metadata are presented. Then, the different XML-based architectures for DW integration that have been proposed in the literature are compared in terms of how they address heterogeneity conflicts and local DW work overload. Section 5 is dedicated to the research on semistructured DWs (research line 2). First, we describe some projects aimed at storing and querying XML data and Web data change control. Afterward, we introduce the work on the multidimensional design for XML sources. We conclude the section by studying the extensions of OLAP techniques in order to manipulate online data in XML format. In Section 6, we address unstructured data and DWs (research line 3). We present different papers that build an IR system over a multidimensional database and the research made on the integration of IR and OLAP techniques for the development of the contextualized warehouses. Finally, Section 7 provides conclusions and points to open research lines.

2 Web Data, XML, and the Semantic Web

According to the authors of the Xyleme project [14]: "The Web is huge and keeps growing at a healthy pace. Most data is unstructured, consisting of text (essentially HTML) and images. Some is structured, mostly stored in relational databases. All this data constitutes the largest body of information accessible to any individual in the history of humanity." However, in order to exploit all this information in applications, new flexible models are required.

In this context, semistructured data models and, in particular, the standardization of XML [1] for Web data exchange play an important role and open a wide new range of possibilities. Two main features of its semistructured data model are the (potential) lack of a predefined schema and its facilities for representing both the data contents and the data structure integrated into the same document. Other advantages of XML as a semistructured data format are its simplicity and flexibility. Moreover, XML is free, extensible, modular, platform independent, and well supported.

The structure of an XML document is given by the use of matching tag pairs (termed elements), and the information between matching tags is referred to as a content element. Furthermore, an element is permitted to have additional attributes, where values are assigned to the attributes in the start tag of the element. Figs. 1 and 2 show two example XML documents.

XML documents can be associated with and validated against a schema, e.g., a Document Type Definition (DTD).

The DTD of an XML document specifies the different elements that can be included in the document, how these elements can be nested, and the attributes they may contain.

Sometimes semistructured XML data sources provide us with irregular and incomplete data whose structure is frequently changing in an unpredictable way. In this case, when available, DTDs are complex and large, and it is not always clear if a specific XML document conforms to the corresponding DTD [24]. In many cases, the associated DTD consists of a sequence of alternatives so large that the DTD-based approach to XML data manipulation is not as useful as expected.

Based on their contents, XML documents are classified into two categories: *data centric* and *document centric* [25]. Data-centric XML documents are highly structured, like the XML document shown in Fig. 1. Document-centric XML are loosely structured and contain large text sections, like the one depicted in Fig. 2. Document-centric XML document collections are typically queried by using IR techniques.

A number of technologies are evolving around XML. These technologies include, among others, XML Schemas [26], an alternative to DTDs that improves data typing and constraining capabilities; the XPath language [27], which is used to refer to parts of XML documents; XQuery [28], the standard query language for XML documents, which provides powerful constructs for navigating, searching, and restructuring XML data; XPointer and XLink [29], which define linking mechanisms between XML documents; and XSL [30], which is a family of recommendations for defining XML document transformation and presentation rules.

Nowadays, the hot topic in Web research is the Semantic Web. The objective of this technology is to describe the semantics of Web resources in order to facilitate their automatic location, transformation, and integration by domain-specific software applications [31]. A number of languages have been proposed to describe the semantics of resources, namely, Topic Maps (XTM) [32], Resource Description Framework (RDF, RDF/S) [33], and Ontology Web Language (OWL) [34].

The World Wide Web Consortium (W3C) leads the development of the XML standard and related technologies. We refer the reader to the W3C Website (http://www.w3.org), where further details can be found.

3 DWs and OLAP

In the last years, there has been a great deal of interest in both the industry and research communities regarding DW and OLAP technologies. Three of the pioneers in the field were W.H. Inmon, R. Kimball, and E.F. Codd.

The classic definition of a DW by Inmon states that a DW is a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions [3]. Another widely accepted definition of a DW is the one by Kimball who defined a DW as a copy of transaction data specifically structured for query and analysis [35]. Thus, data warehousing involves the construction of a huge repository where an integrated view of data is given, which is optimized for analysis purposes. The main problems addressed by the DW technology, which make a DW different from traditional transactional database systems are surveyed in [36].

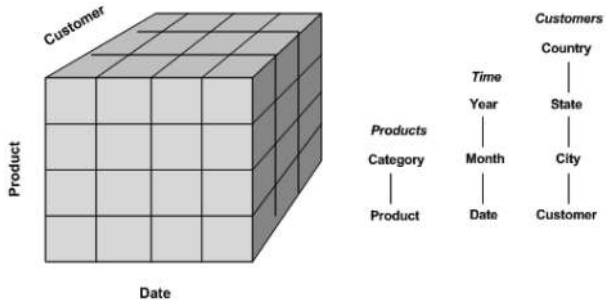

Fig. 3. Example multidimensional data cube.

The information stored in a DW is usually exploited by OLAP tools. The term OLAP was first coined by Codd. In [4], Codd presented 12 rules to evaluate OLAP systems and emphasized the main characteristic of OLAP: the multidimensional analysis. OLAP tools conceptually model the information as multidimensional cubes. Fig. 3 shows an example data cube. In the cubes, data is divided into *facts*, the central entities/events for the desired analysis (e.g., a sale), and *dimensions*, which provide contextual information for the facts (e.g., the products sold). Often, the dimensions are hierarchically organized into levels. For instance, products can be grouped into product categories. Typically, the facts have associated numerical *measures* (e.g., the quantity sold or the total price), and queries aggregate fact measure values up to a certain level (e.g., total profit by product category and month), followed by either roll-up (further aggregation, e.g., to year) or drill-down (getting more detail, e.g., looking at profit per day) operations.

Many commercial DW and OLAP products and services are available today. Multi-Dimensional eXpressions (MDX) [37] is the most used query language for reporting from multidimensional data stores. MDX was first introduced by Microsoft, and nowadays, many OLAP servers and DW client applications support MDX.

The cubes in a DW can be stored by following either a so-called Relational OLAP (ROLAP) and/or a so-called Multidimensional OLAP (MOLAP) approach. In ROLAP, the data is stored in relational tables. In order to map the multidimensional data cubes into tables, different logical schemas have been proposed. The *star* and the *snowflake* schemas are the most commonly used. The star schema consists of a *fact table* plus one *dimension table* for each dimension. Each tuple in the fact table has a foreign key column to each of the dimension tables and numeric columns that represent the measures. The snowflake schema extends the star schema by normalizing and explicitly representing the dimension hierarchies. Fig. 4 shows the differences between star and snowflake schemas. In the MOLAP alternative, special data structures (e.g., multidimensional arrays) are used for the storage instead. The combination of ROLAP and MOLAP is known as Hybrid OLAP (HOLAP). In the HOLAP approach, detailed data is usually stored in relational tables, whereas the MOLAP strategy is applied to manage aggregated data.

The interested reader can find an overview of DWs, OLAP, and multidimensional databases in [38] and [39].

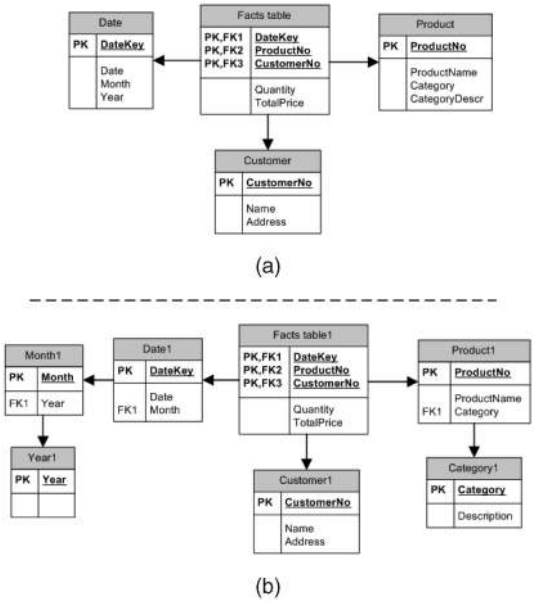

Fig. 4. Example of (a) star and (b) snowflake schemas.

4 XML-Based DW Integration

The Internet has opened an attractive range of new possibilities for DW applications. Companies can now publish some portions of their corporate warehouses on the Web. In this way, customers, suppliers, and people in general will be able to access this "public" corporate data by using Web client applications. The benefits of "plugging" the corporate warehouse into the company website are discussed in [40]. The authors of [3] and [35] study the development of e-commerce applications and click-stream analysis techniques to analyze the behavior of the clients when surfing a company online shop site and then to provide a user customized view of this website according to his/her preferences. An even more challenging issue is to apply Internet technology to provide interoperability between distributed heterogeneous warehouses and to build new (virtual) warehouses where the information available in these heterogeneous warehouses is exploited in a uniform homogeneous integrated way. In this context, XML plays an important role as a standard format of data interchange.

This section describes work focused on XML formats to represent multidimensional data and metadata. Afterward, it introduces the main issues addressed by the research on the integration of (general) data sources and the specific problems of the DW integration. Finally, it reviews the XML-based DW integration architectures proposed in the literature. These architectures use XML languages to express the metadata describing data sources or as a canonical language to transfer data between the different components of the system.

4.1 XML Formats Tailored to DW Interoperability

The first step on the road to interoperability and integration of heterogeneous warehouses is defining a common language for interchanging multidimensional data. With this objective, in [5], a set of XML document formats was proposed, including *XCubeSchema*, which describes the structure of a data cube by providing its measures and dimension schemata (hierarchy of levels in each dimension);

```
<cubeFacts version=''0.4''
xmlns=''http://www.xcube-open.org/XCubeFact_base.xcds''>
<cube id=''sale''>
   <cell>
      <dimension id=''product'' node=''LA-123''/>
      <dimension id=''time'' node=''2005-08-03''/>
      <fact id=''sales'' value=''10''/>
   </cell>
   <cell>
      <dimension id=''product'' node=''RS-133''/>
      <dimension id=''time'' node=''2005-08-03''/>
      <fact id=''sales'' value=''5''/>
   </cell>
...
</cube>
</cubeFacts>
```

Fig. 5. Example *XCubeFact* document [5].

XCubeDimension, which defines the members for each dimension level; and *XCubeFact*, which represents the cells of the data cube (i.e., how the dimension and measure values are linked). Fig. 5 presents the result of exporting a sales data cube in the XML format proposed in [5]. The figure shows a piece of an *XCubeFact* document depicting two cells with sales made on 3 August 2005 for the products LA-123 and RS-133, respectively.

The work presented in [6] also includes its own XML format to interchange data and metadata. This paper describes a Web service interface to evaluate MDX queries in a remote OLAP system. The main difference between the approaches in [5] and [6] resides in their underlying multidimensional model, which in the second case is tightly related to MDX [37]. The authors of [7] propose a UML-based multidimensional model along with its representation in XML. In this case, the XML format is only focused on metadata interchange.

4.2 Integration of Heterogeneous Data Sources

The integration of heterogeneous data sources is a traditional research area in databases whose purpose is to facilitate uniform access to several sources of heterogeneous data, distributed through a set of connected sites that work autonomously. An integrated system provides its users with a global schema where to define their views, along with the mechanisms needed to translate the elements of the global schema into the elements of the corresponding local schema and vice versa. The heterogeneity of the integrated sources usually cause some conflicts that must be solved by the translation mechanisms in order to produce global results that are correct and complete. Heterogeneity conflicts may occur at three different levels:

1. **Physical level.** The data sources to integrate can reside in different computer platforms that run distinct DataBase Management Systems (DBMSs) and operating systems, which provide different communications protocols, etc.
2. **Syntactic level.** Data sources may be based on different data models, support different data types, query languages, etc.
3. **Semantic level.** In different sources, different attribute names may be used for referencing the same data, the data values may be presented in different units (e.g., prices in dollars and euros), etc.

XML technologies and the standard data access Application Programming Interfaces (APIs) available nowadays allow us to manage the heterogeneity conflicts that appear at the physical and the syntactic levels. Many of the sources export their data in XML format today, and standard APIs like ODBC, JDBC, and SOAP [41] provide platform-independent interfaces for querying the data sources. Furthermore, the application of XML technologies to wrapper data sources allows us to solve some semantic heterogeneity problems. For example, by applying XSL transformations, the different attributes used to represent the same information at each local site can be translated to their common representation in the global schema. However, this way of treating semantic heterogeneity is very difficult to automate and also error prone, as any small change in the local or global schemas will require the revision of the transformations made by the wrappers of the involved data sources.

When the data sources to integrate are DWs, two additional issues should be considered:

1. The DWs to integrate must serve a common subject, that is, the analysis tasks that can be made at the global level will be similar to those that can be made at local level on a smaller scale. For example, consider two separated enterprise DWs: European and US sales. In the integrated DW, we could analyze the global sales. The dimensions and measures exported by each local DW will be quite compatible (e.g., both provide dimensions for time and product). However, the DWs may be implemented in different DBMSs, by following a ROLAP or a MOLAP approach, and the way in which facts and dimensions are represented may vary (e.g., product prices are expressed in dollars and euros). At the semantic level, the kind of problems that may appear when integrating DWs are different from the problems that may occur with traditional databases [42]. A classification of semantic heterogeneity conflicts specific to DW integration (e.g., different levels of detail for equivalent dimensions, mismatched dimension names, etc.) can be found in [10] and [13]. Syntactic heterogeneity conflicts can be addressed by applying XML-based formats as those previously described to provide DW interoperability [5], [6], [7].
2. The analysis tasks will produce complex queries over vast amounts of distributed data with hard requirements of time and space to be processed. Therefore, it is very important for the integrated system to have an architecture designed with the purpose of avoiding the overload of both the local sites by the execution of global queries and the network by the transmission of huge amounts of unprocessed data.

Next, we review three existing architectures that apply XML technology to the DW integration. We will compare them in terms of how they address the heterogeneity conflicts and the local DW overload.

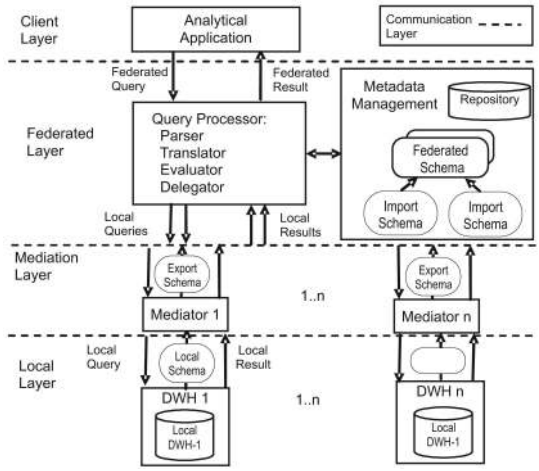

Fig. 6. Federated DW architecture [8].

4.3 XML-Based DW Integration Architectures

One of the first XML-based approaches to the integration of DWs was the architecture presented in [8] and [9]. As Fig. 6 shows, the proposed architecture is organized into four layers, namely, the local, mediation, federated, and client layers. The lower local layer consists of a collection of independent heterogeneous DW systems distributed over the Internet. In order to participate in the federation, each DW should provide its local schema to the corresponding mediator. In the federated layer, the queries of the client applications are first divided into subqueries that are issued to the corresponding mediators, and afterward, the resulting cubes are merged and returned to the client application. In this work, XML documents are used to represent the local, export, and federated schemata. Since these documents represent DW schemata, they are similar to the *XCubeSchema* documents proposed in [5]. The mapping between the federated and the import schemata is also specified in an XML document, in which we can find, for example, the correspondence between federated and local warehouse dimension names.

In [9], a different underlying canonical multidimensional model called *MetaCube* [43] is applied. The authors of this work define a new type of XML document called *MetaCube-X*, which is the XML expression of a *MetaCube* schema representing the export and federated schemata. None of the approaches [8], [9] address query evaluation or the use of XML for representing the results of the local and federated queries. They only focus on schema integration issues. However, as stated by the authors of [8], in order to completely overcome semantic heterogeneity in DW integration (e.g., the different local DWs may define different hierarchies for the same dimension), a deeper study of the mapping strategies is required.

The architecture for integrated DWs proposed in [8] and [9] adapts a traditional architecture of federated databases [44]. The cornerstones of federated databases are the export schemata, the federated schema, and metadata. The export schemata are the expression of the local source schemata in a common canonical model. The federated schema imports the exported schemata and integrates them into a single global schema. Metadata is used to decide where to retrieve

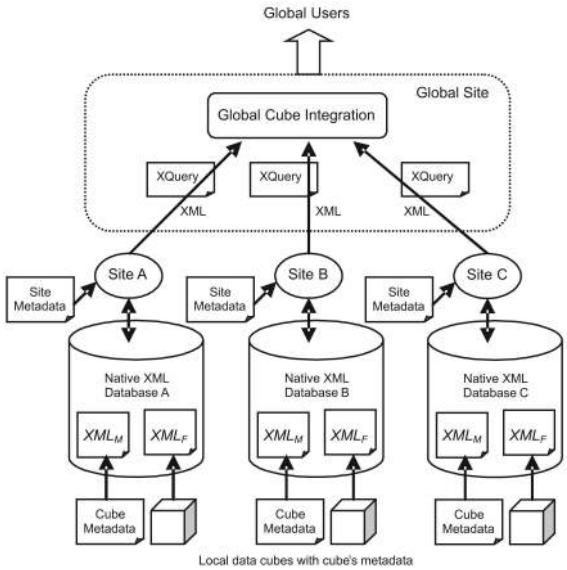

Fig. 7. Federated DW architecture [10].

the data required by global query processing. In the case of [8], all of these three elements are provided in XML format, and the mediators are introduced in the architecture to translate the federated subqueries into the local DW query language. It is important to notice that federated database architectures were designed for applications whose transactions consist of simple selection or update queries that are executed frequently. The case of DWs is quite different, as queries are read only, typically very complex and less frequently run.

Summarizing the contributions of the architecture proposed in [8], it provides a high degree of local autonomy, but it presents some limitations regarding how heterogeneity and load distribution are addressed. Heterogeneity issues are solved by manually programming the mediators of each local DW, making their maintenance difficult. Every change in the local or global schemas will need a human intervention to update the implicated mediators. Furthermore, as pointed out in [10], each time a query is executed, the corresponding local subqueries and the combination of local results must be reprocessed, because previous results cannot be reused. In other words, the local cubes that were merged to answer a query must be rebuilt each time they are needed. This produces unnecessary overload both at the local sites and at the global processor that builds the final cube.

The work made in [10] also proposes a federated architecture. In this case, the mediator components are replaced by native XML databases (see Fig. 7). Each native XML database stores the cubes available in the corresponding local warehouse along with their export schemata. Each local database manager provides its *site metadata*, which is a formal description of the dimensions and the semantics of the measures involved in the exported cubes. Heterogeneity conflicts between export schemata are solved semiautomatically by studying the *site metadata* and by designing and evaluating XQuery statements [28] to update the exported XML data cubes and their schemata. Finally, the resulting cubes are integrated into a global cube that can be analyzed by users. This approach also ensures the autonomy of the

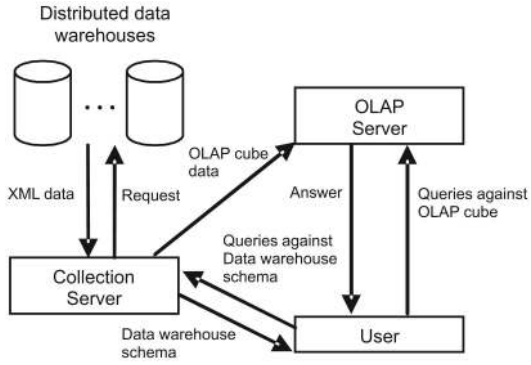

Fig. 8. Distributed DW architecture proposed in [11] and [12].

local DWs. The heterogeneity conflicts are solved semiautomatically. With the architecture proposed in [10], the local sites are not overloaded during the execution of global queries, as intermediate results can be retrieved from the corresponding XML databases. The major drawback of this approach is that the efficiency of XML databases to compute aggregations is still not comparable to the performance of the ROLAP or MOLAP-based OLAP systems.

A different architecture, based on Grid technology [45], is proposed in [11] and [12]. Fig. 8 shows the system architecture. Analysis tasks take place as follows:

1. A virtual universal DW schema representing all the data available in the local warehouses is presented to the user, who formulates an analysis query.
2. The *Collection Server* processes the query, sending requests to the relevant warehouses according to a distribution schema (i.e., how the data is distributed among the different warehouses).
3. The involved warehouses compute the selections and aggregations in parallel. A Grid-based distributed computing platform is used to perform this distributed data processing.
4. The *Collection Server* receives the data and performs a final aggregation, if needed and sends the resulting cube data to the OLAP Server.
5. The user analyzes the cube in the OLAP Server.

In [11] and [12], XML is used to represent the universal cube schema, the initial user query, the distribution schema, the data returned by each DW, and the final analysis cube data. The authors of this work propose to transform the XML data returned by the warehouses into a format suitable for the OLAP server by applying standard XML tools like XSLT [30]. The use of the XSLT transformation sheets is conceptually interesting here, but it seems not efficient enough to manage huge amounts of data. Unfortunately, the authors of [11] and [12] do not provide an evaluation of the performance of their system. Their main contribution is the use of Grid technology to distribute the computation needed in the construction of huge cubes of data coming from a large number of autonomous sites. However, they do not study how to solve semantic heterogeneity conflicts, which is an important limitation for integrating very autonomous sites.

The application of XML has meant a great advance toward DW integration, since it provides a common data model that solves the syntactic heterogeneity conflicts. However, semantic heterogeneity discrepancies between

DW schemata are still handled manually [8] or semiautomatically [10]. Trying to automatically address these conflicts, Semantic Web languages have been applied to describe DW conceptual schemata. The work in [13] also proposes a federated architecture and applies Topic Maps to solve semantic heterogeneity issues in DW integration. Topic maps have been already used for modeling metadata in XML format (XML Topic Maps [32]). In the approach in [13], the measures, dimensions, and hierarchy dimension levels of each site are represented by their local topic maps. Association relations are used for modeling the fact structure (i.e., the dimensions and measures that constitute the fact) and the roll-up relationships between dimension levels. Afterward, at the federated layer, an ontology-based integration process builds the global topic map that provides the integrated view of the local schemas and deals with the semantic conflicts between them. For example, consider the *Time* dimension defined in two different local schemas. These dimensions include two equivalent levels "*day*" and "*tag*" (day in German). In the global topic map, there will be only one topic, "*day*," with two scopes, *English* and *German*. Then, each scope will be linked to the corresponding dimension. The major contribution of [13] is a framework for the semantic integration of the DW schemata. They do not address the aspects of query evaluation and load distribution.

4.4 Summary and Open Research Lines

As we have discussed in this section, the first step on the integration of heterogeneous DWs is defining a common language for multidimensional data exchange. In this context, XML plays an important role as the standard format of data interchange. In the literature, we have found different works that propose XML formats tailored to multidimensional data. The formats proposed in [5] and [6] are suitable for expressing multidimensional data and metadata, whereas the XML format presented in [7] is only focused on multidimensional metadata interchange. The one that seems to be most widely adopted by the DW and OLAP industry is [6], since it is based on the data model of the standardized MDX OLAP query language [37].

Regarding the architectures for the integration of DWs, in this section, we have introduced different papers that propose a federated architecture [8], [9], [10], [13] or apply Grid computing technology [11], [12] and use multidimensional XML formats for solving the syntactic heterogeneity conflicts among the local DWs. The work presented in [11] and [12] does not address semantic heterogeneity conflicts. The semantic discrepancies are handled manually in [8], by implementing a mediator component over each local DW, and semiautomatically in [10], by running XQuery statements on the exported XML data cubes and metadata. The authors of [13] address the semantic conflicts by defining an ontology that relates the local schemas, which are represented by topic maps. The architecture of [8] and [9] overloads the local DWs, since each global query requires querying the local DWs. The local DW overload problem is solved in [10] by storing a copy of each exported data cube in a native XML database. The major drawback of [10] is the low performance of XML databases to evaluate OLAP operations.

Current approaches to the integration of DWs are oriented to specific scenarios, where local schemata are well known, and export schemata are manually built to comply with the global schema rules. These approaches clearly cannot be applied to the emerging large-scale scenarios, where DWs can become public by simply providing them as Web services. The big challenge here is to handle high levels of semantic heterogeneity. The Semantic Web is the result of the research made to bring knowledge to the Web in order to facilitate the location of concepts and to improve interoperability between applications. Other examples of the most important outcomes of this research are standard languages to specify shared knowledge in the form of *domain ontologies* (e.g., RDF/S and OWL). In the same way that XML tries to solve syntactic heterogeneity problems, these languages try to solve the semantic ones. The main idea behind the use of *domain ontologies* is to set up a common terminology and logic for the concepts involved in a particular domain. In the case of DWs, these concepts could describe the facts, dimensions, categories, and values implied in analysis subjects. Thus, publicly available DWs should provide a semantic description of their resources (i.e., schema, data, and operations) according to the adopted *domain ontology*.

It is worth mentioning that the application of Topics Maps presented in [13] does not follow the idea of *domain ontology*, since they are rather used as a pairwise mapping description among a small set of DW schemata. Notice that *domain ontologies* are devised as rich logical descriptions that allow users and applications to manage a large variety of resources, not only DWs. Here, the main issue is how *domain ontologies* can help DWs to interoperate, not only between them, but also with other information-provider applications.

5 Utilization of XML Data in DWs

With the emergence of XML as the lingua franca of the Web, semistructured information is now widely available, and several solutions have been proposed to build warehouses for XML data. This section first introduces work oriented toward the construction of XML Web data repositories, then presents the research done on the design of multidimensional databases for XML data, and finally focuses on the extension of OLAP techniques to XML data.

5.1 XML Web Data Repositories

The problem of gathering and querying Web data is not trivial, mainly because data sources are dynamic and heterogeneous. In this context, some papers are focused on the construction of repositories for XML [14] or Web documents [15]. The main issues of this research area include the efficient acquisition, storage, indexing, query processing, change control, and schema integration of data extracted from dynamic and heterogeneous Web sources. This section summarizes the main results of two important projects: Xyleme [14] and Warehouse of Web Data (Whoweda) [15].

Xyleme [14] was an ambitious project aimed at building a warehouse for all the XML data available on the Web. The Xyleme system runs on a network of distributed Linux PCs.

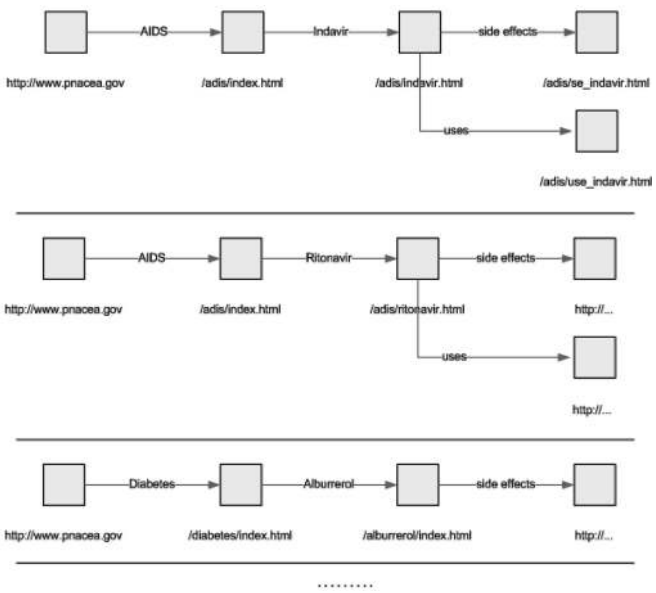

Fig. 9. Example of a Web table in WHOM [46].

In order to store such a huge amount of XML data, a hybrid approach is proposed to keep the tree structure of XML documents in a traditional DBMS until a certain depth and then store the pieces of documents under the selected depth as byte streams. Thus, the upper part of the XML document structure is always available, but the lower sections require parsing to obtain the structure. Query processing is based on an algebra operator that returns the set of documents that satisfy a given tree pattern. Xyleme partitions the XML documents into clusters corresponding to different domains of interest (e.g., tourism, finance, etc.), which allows indexing each cluster on a different machine. Since the documents in a cluster may follow different DTDs, an abstract DTD for the cluster along with the mappings to the original DTDs is inferred. In this way, the user queries the cluster by using the abstract DTD. In order to acquire the XML documents, several crawlers run in parallel. The refreshment of a copy is performed depending on the importance of the document or its estimated rate change or under the request of the owner of the document (i.e., in a notification/subscription basis).

The Whoweda project is also aimed at warehousing relevant data extracted from the Web [15]. This project is mainly focused on the definition of a formal data model and an algebra to represent and manage Web documents [46], their physical storage [47], and change detection [48]. In the data model, called Warehouse Object Model (WHOM) [46], a Web warehouse is conceived as a collection of the so-called *Web tables*. A Web table is a special construct of the WHOM that represents sets of interlinked documents of the WWW. The tuples of the Web tables are multigraphs where each node represents a document, and the edges depict hyperlinks between documents. Fig. 9 shows some tuples of an example Web table with documents about "drugs." In order to manage the data stored in the Web tables, a set of algebraic operators is provided (e.g., global Web coupling, Web join, Web select, etc.). For example, the global Web coupling operator retrieves a set of interlinked documents satisfying a query with conditions on the metadata, content, structure, and hyperlinks of the documents. The result of the operation is a new Web table where each new tuple matches a portion of the WWW satisfying the constraints of the query. In the Web join operator, the tuples from two Web tables containing identical nodes are "concatenated" into a single joined Web tuple. Two nodes are considered identical if they represent the same document with the same URL and modification date.

XML data change control is an important issue that has spawned a lot of research. Xyleme [14] allows users to subscribe to changes in an XML document [49]. When such a change occurs, subscribers receive only the changes made, called *deltas* [50], [51], and then incrementally update the old document. This approach is based on a versioning mechanism [51] and an algorithm to compute the difference between two consecutive versions of an XML document [50]. The Whoweda project addresses change detection over sets of interlinked documents, instead of over isolated XML documents. The global coupling algebra operator may be used to state a set of relevant interlinked documents to "watch." Given two versions of this set of interlinked documents materialized in two different Web tables, the differences between these two versions are calculated by applying the Web join and the Web outer join algebra operators. The authors of [52] considered a more general problem by studying how to update materialized views of graph-structured data when the sources change. In [53], an adaptive query processing technique for federated database environments was proposed. Finally, [54] and [55] consider adaptivity in a federation of XML and OLAP data sources (see Section 5.3).

5.2 XML Multidimensional Database Design

This section surveys the most relevant research on multidimensional design for XML data. Specifically, the work of Golfarelli et al. [16], Pokorný [17], and Jensen et al. [18] are studied.

The authors of [16] argue that existing commercial tools support data extraction from XML sources to feed a warehouse, but both the warehouse schema and the logical mapping between the source and target schemas must be defined by the designer. They show how the design of a data mart can be carried out starting directly from an XML source and propose a semiautomatic process to building the DW schema.

Since the main problem in building a DW schema is to identify many-to-one relationships between the involved entities, they first study how these relationships are depicted in the DTD of the XML documents. Such relationships are modeled by subelements nesting in DTDs. ID/IDREF attributes of the DTDs are not considered, since IDREFs are not constrained to be of a particular element type. For example, if ID attributes are defined for the elements `car` and `manufacturer` and an IDREF attribute is stated for an `owner` element, the IDREF attribute of the `owner` element may reference either a `car` or a `manufacturer` element in an instance XML document. The authors provide an algorithm that represents the structure modeled by the DTD as a graph and, starting from a selected element (the analysis fact), semiautomatically builds the multidimensional schema by including the dimension and dimension

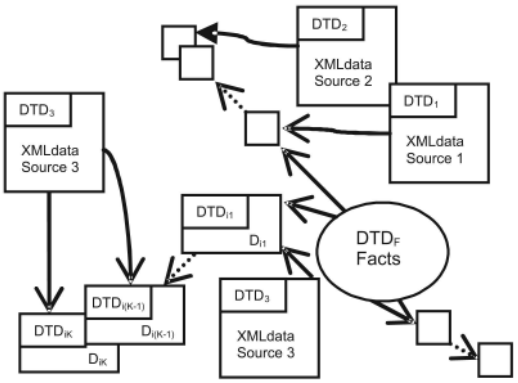

Fig. 10. XML-star schema proposed in [17].

levels depicted by the many-to-one relationships found between the elements and attributes of the graph. In order to understand why the designer participation is needed, consider the following example. In a DTD, the definition `owner(car*)` states that an owner may have many cars. However, the cardinality of the inverse relationship is not stated in the DTD. That is, the same car may belong to several owners. The problem is solved by querying the document instances and asking the user.

In [16], it was assumed that the schema of the source XML data is provided by a single DTD. In [17], a different approach is followed, by considering that when the source XML data is gathered from different sources, then each source will provide its particular DTDs. Thus, the author of [17] defines an XML-start schema by modeling the dimensions as sequences of logically related DTDs (see Fig. 10). It is assumed that a single DTD describes the measures of the facts. In order to build the dimension hierarchies, this approach defines a sub-DTD as the portion of a source DTD that characterizes the structure of a dimension hierarchy member. Then, XML view mechanisms are applied to select the members of each dimension hierarchy. The concept of referential integrity for XML data is introduced to establish the hierarchical relationships between the dimension members. Referential integrity control is performed by checking whether the XML tree of the child dimension member can be embedded within the XML tree of the parent dimension member.

The work in [18] deals with the conceptual design of multidimensional databases in a distributed environment of XML and relational data sources. This approach uses UML diagrams [56] to describe the structure of the XML documents, as well as the relational schema. For relational databases, commercial reverse engineering tools can be applied to build the corresponding UML diagrams. For XML documents, they propose an algorithm [57] that builds the UML diagram from the DTDs of the XML sources. They also provide a methodology to integrate the source schemata into an UML snowflake diagram and take special care in ensuring that XML data can be summarized. For example, they study how XML elements with multiple parents, ID references between elements, or recursive element nesting should be managed. The resulting UML schema can be applied for the integration of sources in a multidimensional database.

All the approaches reviewed in this section assume that the logical structure of XML documents is described by DTDs. In [16], the multidimensional schema is built starting from a single DTD. That is, the authors of [16] assume that the complete multidimensional schema is represented within a single DTD and that the hierarchy dimension members and measures can be found in a collection of XML documents that conform to this DTD. The approach presented in [17] is more flexible, in the sense that it allows the designer to combine different portions of several DTDs in order to construct the dimension hierarchies. Thus, several XML document sources can be used for populating the data cubes. The algorithm proposed in [57] transforms a single DTD into a UML diagram. However, in the framework proposed by the same authors [18], different classes taken from UML diagrams of different DTDs or relational schemas can be mixed to design the final multidimensional schema. In this case, the user is responsible for specifying the relations (and their cardinalities) that exist between the classes that come from different diagrams (i.e., DTDs or relational schemas). The three approaches [16], [17], [57] use the nesting relationships that appear between the document elements for establishing hierarchical relationships between the different dimension levels. In addition, the algorithm presented in [57] also use the ID/IDREF attributes for the same purpose. This usage of the ID/IDREF attributes is only valid for XML documents in which all the IDREF attributes of a given element type only reference the ID attributes of a particular element type. Other formalisms for describing the structure of XML documents, more powerful than the DTD grammar, exist, e.g., the XML Schemas [26]. In [58], the work presented in [16] is extended to design the multidimensional schema starting from an XML Schema.

5.3 XML-Based Extension of OLAP Techniques

This section mainly studies the work of Pedersen et al. on the extension of OLAP techniques to XML data [19], [20]. Pedersen et al. argue that the dynamicity of today's business environments are not handled well by current OLAP systems, since physically integrating data from new sources is typically a long time-consuming process, making logical integration the better choice in many situations. Thus, by considering the increasing use of XML for publishing Web data, they aim their work at the logical federation of OLAP and XML data sources. Their approach allows the execution of OLAP operations that involve data contained in external XML data. In this way, XML Web data can be used as dimensions [19] and/or measures [20] of the OLAP cubes.

The OLAP-XML federations proposed in [19] and [20] use links for relating dimension values of a cube to elements of an XML document (e.g., linking the values of a Store-City-Country dimension to a public XML document with information about cities, such as state and population). Thus, a federation consists of a cube, a collection of XML documents, and the links between the cube and the documents. The most fundamental operator in OLAP-XML federations is the *decoration operator* [59], which adds a new dimension to a cube based on the values of the linked XML elements. This work presents an extended multidimensional

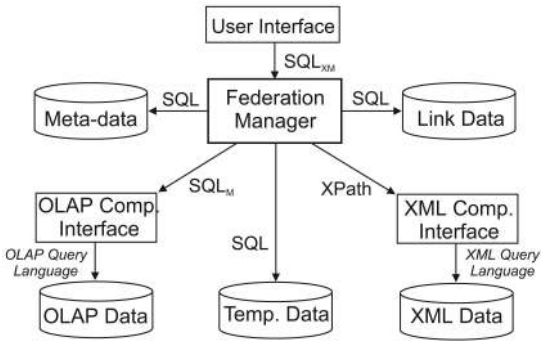

Fig. 11. OLAP-XML federation architecture [19].

query language called SQL_{XM} that supports XPath expressions and allows linked XML data to be used for decorating, selecting, and grouping fact data. For example, the following query computes the total purchase quantities grouped by the city population that is found only in the XML document:

```
SELECT SUM(Quantity), City/Population
FROM Purchases
GROUP BY City/Population
```

Fig. 11 shows the architecture of the system proposed in [19]. Along with the Federation Manager, it includes an OLAP component (i.e., a commercial OLAP server able to evaluate multidimensional queries) and an XML component (i.e., an XML database system with an XPath interface). In Fig. 11, SLQ_M depicts regular multidimensional OLAP queries (e.g., MDX queries [37]), whereas SQL_{XM} represents the XPath-extended multidimensional queries proposed in [59]. The Federation Manager receives SQL_{XM} queries and coordinates their execution in the OLAP and the XML databases. The metadata, link data, and temporary data databases (e.g., traditional relational databases) are also managed by the Federation Manager component.

The unoptimized approach to process an SQL_{XM} query is described as follows: First, any XML data referenced in the query is fetched and stored in a temporary database as relational tables. Second, a pure OLAP query (SQL_M) is constructed from the SQL_{XM} query, resulting in a new table in the temporary database. Finally, these temporary tables are joined, and the XML-specific part of the SQL_{XM} query is evaluated on the resulting table along with the final aggregation.

Pedersen et al. provide both rule-based and cost-based optimization strategies focused on reducing the amount of data moved from the OLAP and XML components to the temporary database. The rule-based optimization algorithm partitions an SQL_{XM} query tree, meaning that the algebra operators are grouped into an OLAP part (SQL_M), an XML part $(XPath)$, and a relational part (SQL). Algebraic query rewriting rules are applied to push as much of the query evaluation toward the OLAP and XML components as possible. The cost-based optimization strategies are based on the cost model described in [60] and a set of the techniques that include in-lining literal XML data values into OLAP predicates, caching, and prefetching [61].

In a more recent paper [20], Pedersen et al. show an implementation of their XML-OLAP federation for the commercial OLAP tool TARGIT Analysis and extend their approach to allow the evaluation of federated OLAP queries with XML data as measures.

5.4 Summary and Open Research Lines

Organizations can now find highly valuable information about their business environment on the Web. Most of these data is available in XML format. In this section, we have introduced two of the most important projects on Web data warehousing [14], [15]. The work presented in [14] is focused on gathering, storing, and querying XML data, whereas [15] considers both XML and HTML documents. The first project is based on the typical tree-like representation of the XML documents. The data model of the second one relies on a special construct, called *Web table*, that represents sets of interlinked documents. In [14], a tree pattern algebra is provided for querying. In this case, the query result is a set of document pieces, whereas the algebraic operators in [15] return *Web tables* (i.e., sets of interlinked documents). That is, [14] operates at the document level, and the queries in [15] involve sets of documents related by means of hyperlinks. Since Web data is highly dynamic, an important issue addressed in [14] and [15] is data change control. The two approaches act on a subscription basis, i.e., the users specify which are the documents to "watch," and provide mechanisms to compute the differences between two versions of a document (document sets in [15]). In [15], the difference is calculated by performing a join operation on the relevant *Web tables*. Neither [14] nor [15] addresses the analysis of the data.

The research on analyzing XML data mainly follows two different directions: the physical integration of the XML sources in a multidimensional database [16], [17], [57] versus the integration of the XML sources and a multidimensional database at a logical level [19], [20]. The physical integration provides better query performance, whereas the logical integration is more suitable for frequently changing XML data. Logical integration also implies extending the existing OLAP techniques to allow the execution of queries that involve online XML data [19], [20]. The papers on the physical integration address the design of the multidimensional database schema starting from the DTDs that describe the structure of the XML documents. Once the XML data is loaded into the database, the traditional OLAP techniques can be used for querying. A comparative study of this group of papers [16], [17], [57] was done at the end of Section 5.2.

Nowadays, the major database companies provide XML extensions to index and query XML data and to support XML as a built-in datatype [62], [63], [64]. We foresee that these database companies will integrate XML extensions into their OLAP tools. The work in [57] on multidimensional schema design for relational and XML data will have a direct application here.

As already mentioned in Section 2, sometimes the XML sources provide documents with a very irregular and dynamic structure. In this case, the DTDs are not always available, and when available, they consist of sequences of structural alternatives so large that the direct application of the multidimensional schema design techniques proposed in [16], [17], [57] is difficult. Even when the DTD of an XML document collection is not provided, the logical structure of

the documents is still implicitly given in their contents by the tag's attributes and nesting relationships. Some papers (see, for example, [65]) present algorithms to cluster XML documents with similar structure and infer the DTD that depict the structure of each XML document cluster. These algorithms can be run as a preprocessing stage, before addressing the design of the analysis schema.

A recent area of research is the extension of native XML databases and their query languages to perform OLAP-like analysis [66], [67]. The work presented in [66] proposes an extension of the XQuery language with constructs for the grouping and numbering of results. The new constructs simplify the construction and evaluation of queries requiring grouping and ranking, and at the same time, they enable complex analytical queries. The authors of [67] propose a distinct grouping operator for XML, where the grouping dimensions are specified by means of tree patterns [68]. This work [67] studies the summarizability problems that arise when aggregating XML data (e.g., optional and repeatable elements may result in missing or double-counting some data) and presents different algorithms for computing the data cubes efficiently. It is still too early to talk about native XML-OLAP (XOLAP), since the performance of the native XML approach is not comparable to the efficiency of the ROLAP/MOLAP approaches. Nevertheless, directly using a native XML database for analyzing XML data supposes an attractive alternative to the physical or logical integration of these data in a multidimensional database. The results obtained in [67] are promising. They encourage to continue the research in this line and study specific indexing and optimization strategies for OLAP in XML databases.

In the future, with the Semantic Web widely adopted, companies will be able to gather huge amounts of valuable semantically related data concerning their subjects of interest. Then, an interesting topic of research is the extension of the work on physically or logically integrating XML data in a multidimensional database to deal with semantically annotated data. That is, the design of the multidimensional database schema starting from ontology representations (e.g., OWL [34] document collections) or the extension of the traditional OLAP operations, in order to use external online semantically annotated data as dimensions or measures of the analysis cubes. As far as we know, currently, the only work in this line is [69].

Finally, notice that the proposals surveyed in this section deal with highly structured XML data (e.g., online XML product pricing lists), from where the measures and dimensions can be directly selected using XPath expressions or tree patterns. These approaches are not suitable for analyzing document-centric XML collections, which require some kind of document processing to extract measures and dimension values from the documents textual contents [70]. Section 6 deals with the combination of DW and IR technologies to exploit text-rich XML documents.

6 The Combination of DWs and Document-Centric XML Collections

Many new Web applications store unstructured data with large text portions requiring IR techniques [2] to be indexed, queried, and retrieved.

In an IR system, the users describe their information needs by supplying a sequence of keywords. The query result is a set of documents ranked by relevance. The relevance is a numerical value that measures how well the document fits the user information needs. Traditional IR models (e.g., the vector space model [71]) calculate this relevance value by considering the local and global frequency (tf-idf) of the query keywords in the document and the collection, respectively. Intuitively, a document will be relevant to the query if the specified keywords appear frequently in its textual contents and they are not frequent in the collection. Newer proposals in the field of IR include language modeling [72] and relevance modeling [73] techniques. The papers on language modeling consider each document as a language model. Thus, documents are ranked according to the probability of obtaining the query keywords when randomly sampling from the respective language model. An extension of the language modeling approach is relevance modeling [73], which estimates the probability of observing a query keyword in the set of documents relevant to a query. The language and relevance modeling approaches still internally apply the keyword frequency to estimate probabilities, and they have been shown to outperform baseline tf-idf models in many cases [72], [73].

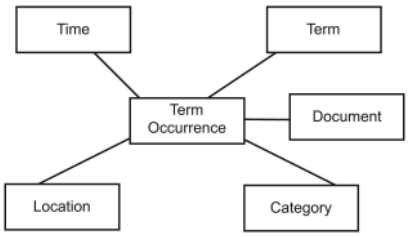

Fig. 12. Multidimensional implementation of an IR system proposed in [21].

In this section, we study how the OLAP and IR approaches have been combined. Current research follows two main lines: the application of multidimensional databases to implement an IR system and the extension of OLAP techniques to support the analysis of text-rich documents.

6.1 Cubes for Document Analysis and Retrieval

OLAP cube dimensions provide an intuitive general-to-specific (or vice-versa) method for the analysis of document contents. Moreover, the optimized evaluation of aggregation functions in multidimensional databases can be applied to efficiently compute the relevance formulas of IR systems. This section studies how multidimensional databases and OLAP can help IR.

The work presented in [21] implements an IR system based on a multidimensional database. As Fig. 12 shows, the fact table measures the weights (i.e., frequency) of each term at each document. Thus, the relevance of a document to a query is computed by grouping its term weights, which are obtained by slicing the cube on the term dimension. The final relevance value is calculated by applying the so-called pivoted cosine formula [74] to the weights of the query terms. Furthermore, if the document collection is categorized by location and time, more complex queries can be formulated, like retrieving the documents with the terms "financial crisis" published during the first quarter of 1998

in New York and then drilling down to obtain those documents published in February 1998. Following this line of research, in [75], the authors study different indexing strategies to improve the performance of their system and, in [76], propose a method for incorporating a hierarchical category dimension to classify the documents by theme.

The benefits of implementing an IR system on a multidimensional database are also discussed in [22] together with a novel user interface for exploring document collections. This approach defines a dimension for each subject of analysis relevant to the application domain (e.g., in a financial application, subjects such as economic indicators, industrial sectors, and regions are relevant dimensions). Each dimension is modeled as a concept hierarchy. They choose a star schema too, but instead of keeping term weights, the fact table links documents to categories of concepts.

Finally, a recent paper [77] provides a mechanism to perform special text aggregations on the contents of XML documents, e.g., getting the most frequent words of a document section, their most frequent keywords, a summary, etc. Although these text-mining operations are very useful to explore a document-centric XML collection, they cannot be applied to evaluate OLAP operations over the facts described by document textual contents. This is the focus of Section 6.2.

6.2 IR Techniques Applied to OLAP

Most information is published on the Web as unstructured documents. These documents typically have large text sections and may contain highly valuable information about a company's business environment. The current trend is to find these documents available in XML-like formats [14]. This situation opens a novel and interesting range of possibilities for DW and OLAP technology: trying to include the information described by these text-rich XML documents in the OLAP analysis. We can thus imagine a DW system able to obtain strategic information by combining all the company sources of structured data and documents.

The approaches discussed in Section 6.1 to implement an IR system by using a multidimensional database are very useful to explore a document-centric XML collection. However, these techniques cannot be applied to evaluate OLAP operations over the facts described by document textual contents. The extension of OLAP techniques for XML data studied in Section 5.3 are not suitable for analyzing text-rich documents either. They only deal with highly structured XML data (e.g., online XML product pricing lists), from where the measures and dimensions can be directly selected using XPath expressions.

The analysis of the factual information described in the textual contents of the documents is a hard issue. It is difficult to find work in the current literature that tries to address this problem. For this purpose, some kind of document processing to extract measures and dimension values from their textual contents [70] is needed.

The authors of [23] propose a setting where this analysis is possible, called a contextualized warehouse. In particular, they propose to *contextualize* the facts of a traditional corporate DW with the documents that describe their circumstances. The dimension values found in the documents will be used to relate documents and facts. Thus, a contextualized warehouse is a new type of decision support system that allows users to combine all their sources of structured and unstructured data and to analyze the integrated data under different contexts.

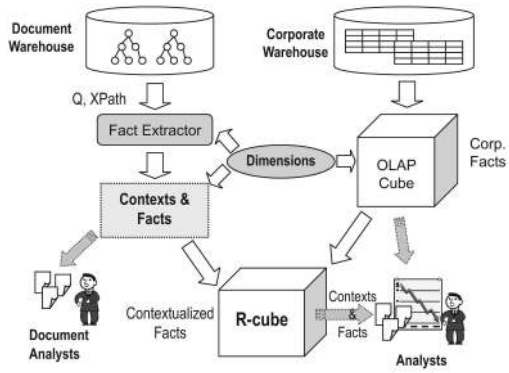

Fig. 13. Contextualized warehouse architecture [23].

Fig. 13 shows the architecture proposed for the contextualized warehouse. Its main components are a corporate warehouse, an XML document warehouse, and the fact extractor module. The corporate warehouse is a traditional DW that integrates the company's structured data sources (e.g., the different department databases). The unstructured data coming from external and internal sources are stored in the document warehouse as XML documents. These documents describe the context (i.e., circumstances) of the corporate facts. The document warehouse allows the user to evaluate queries that involve IR conditions. The fact extractor module relates the facts of the corporate warehouse with the documents that describe their contexts. This module identifies dimension values in the textual contents of the documents and relates each document with the facts that are characterized by these dimension values.

In a contextualized warehouse, the user specifies an analysis context by supplying a sequence of keywords (i.e., an IR condition like "financial crisis"). The analysis is performed on a new type of OLAP cube, called *R-cube*, which is materialized by retrieving the documents and facts related to the selected context.

R-cubes have two special dimensions, the *relevance* and the *context* dimensions. Thus, each fact in the *R-cube* will have a numerical value representing its relevance with respect to the specified context (e.g., how important the fact is for a "financial crisis"), thereby the name *R-cube* (Relevance cube). Moreover, each fact will be linked to the set of documents that describe its context.

The relevance and context dimensions provide information about facts that can be very useful for analysis tasks. The relevance dimension can be used to explore the most relevant portions of an *R-cube*. For example, it can be used to identify the period of a political crisis or the regions under economical development. The usefulness of the context dimension is twofold. First, it can be used to restrict the analysis to the facts described in a given subset of documents (e.g., the most relevant documents). Second, the user will be able to gain insight into the circumstances of a fact by retrieving its related documents.

The IR model for retrieving the documents that describe the analysis context and estimating the relevance of the facts

described by these documents to the analysis context (IR query) was presented in [78]. The data model and algebra for the R-cubes are described in [23] and extend the multidimensional model of [79]. Finally, a system implementation based on multidimensional databases is proposed in [80].

From a different point of view, the work presented in [81] proposes to annotate external information sources (e.g., documents, images, etc.) by means of an ontology in RDF format that comprises all the values of the DW's dimensions. In this way, the results of OLAP queries can be associated with the external sources annotated with the same dimension values. However, unlike [23], it does not provide a formal framework for calculating fact relevance with respect to user queries.

The research on contextualized warehouses can be continued by following several directions. One of these research lines is the direct analysis of the factual data described by the documents. Notice that the document warehouse may provide highly valuable strategic information about some facts that are not available in the corporate warehouse nor in external databases. Sometimes, it is relatively easy to obtain these facts, for example, when they are presented as tables in the documents. However, many times documents contain already aggregated measure values. The main problem here is to automatically infer the implicit aggregation function that was applied (i.e., average, sum, etc.). Different IR and information-extraction-based methods for integrating documents and databases are discussed in [82]. Specifically, [82] proposes a strategy to extract from documents information related to (but not present in) the facts of the warehouse. The work on contextualized warehouses introduced in this section [23] shows how the dimension values found in documents can be applied to the process of relating them with the corporate facts that have the same dimension values. Trying to directly analyze the facts extracted from the documents without considering the corresponding corporate facts is an even more challenging task. In this case, the analysis may involve facts that are incomplete (not all the dimensions may be quoted in the documents contents) and/or imprecise (if the dimension values found belong to nonbase granularity levels). The *R-cubes* base model supports incompleteness and imprecision [79]. In the future, these features can be exploited to analyze the facts described in the documents that are not available in the corporate warehouse.

Another interesting research topic is to integrate the architecture of the contextualized warehouse with existing Web search engines, thus providing better scalability, as well as the possibility to contextualize the data cubes with online Web documents.

7 Conclusions

The advent of XML and related technologies is playing an important role in the future development of the Web. DW and OLAP tools also take part in the Web revolution. This paper has summarized the most relevant research on combining DWs with Web/XML data and technologies. We classify the work in this field into three research lines: 1) the use of XML technology as an integration tool in distributed DW systems, 2) the development of DWs for semistructured XML Web data, and 3) the combination of OLAP and IR to manage unstructured data in a DW.

The first research line studies the work on XML formats tailored to express multidimensional metadata [7] and both data and metadata [5], [6]. The format proposed in [6] is based on the data model of MDX and is the most widely used in the OLAP industry. Several architectures apply these XML formats for solving the syntactic heterogeneity conflicts that appear in the integration of DW systems [8], [9], [10], [13], [11], [12]. The architectures in [8], [9], [10], and [13] adapt the federated architecture to DWs, whereas [11] and [12] are based on Grid technology. The semantic discrepancies between the local DWs are handled manually in [8] and semiautomatically in [10]. The semantic conflicts are addressed in [13] by using topic maps for describing and relating the local schemas.

In the future, the Semantic Web will provide us with domain ontologies, i.e., rich logical descriptions that will allow users and applications to manage a large variety of resources from different domains. Here, the main issue is how domain ontologies can help DWs to interoperate in a large-scale scenario, not only between them, but also with other information-provider applications.

Within the second research line, we have introduced the work aimed at storing, querying, and controlling changes in XML documents [14] or sets of interlinked HTML/XML documents [15]. Neither [14] nor [15] addresses the analysis of the XML data. In order to analyze XML data in a DW, two different approaches are followed: the physical integration of the XML sources in a multidimensional database versus the integration of the XML sources and a multidimensional database at a logical level. The papers on physical integration propose different techniques for designing the multidimensional analysis schema starting from a single DTD [16], a set of DTDs [17], or both DTDs and a relational schema [57]. The papers on logical integration address the extension of OLAP techniques for analyzing online XML data [19], [20].

An emerging area of research is the extension of native XML databases and their query languages to perform OLAP-like analysis [66], [67], i.e., XOLAP. This approach supposes an attractive alternative to the physical or the logical integration of the XML sources in a traditional multidimensional database. Although still not comparable to the performance of the ROLAP and MOLAP systems, the results obtained in [67] encourage continuing the research by studying indexing and optimization strategies for OLAP in XML databases. Another promising topic of research is physically or logically integrating semantically annotated data extracted from the Semantic Web in a multidimensional database, that is, the design of the multidimensional database schema starting from ontology representations like OWL or the extension of OLAP with operators able to use online semantically annotated data as dimensions or measures, respectively.

Most information is nowadays published on the Web as unstructured documents. The proposals surveyed within the second research line only deal with data-centric XML and are not suitable for analyzing document-centric XML collections. In the third research line, we have showed how IR and OLAP technologies can be combined to explore

document collections, (i.e., the use of multidimensional databases for implementing IR systems [21], [22]) and to analyze facts and documents together in the contextualized warehouses [23].

The research on contextualized warehouses is still novel and can be continued in several ways. A challenging one is the direct analysis of the facts extracted from the documents, without contextualizing a traditional data cube. Another interesting topic of research is to integrate Web search engines within the architecture proposed in [23] to contextualize the data cubes with online Web documents.

ACKNOWLEDGMENTS

This work was partially supported by the Spanish National Research Project TIN2005-09098-C05-04, the Fundació Bancaixa Castelló, and the Danish Research Council for Technology and Production under Grant 26-02-0277.

REFERENCES

[1] E. Maler, T. Bray, J. Paoli, F. Yergeau, and C.M. Sperberg-McQueen, *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, World Wide Web Consortium (W3C) recommendation, http://www.w3.org/TR/2006/REC-xml-20060816, Aug. 2006.

[2] R.A. Baeza-Yates and B.A. Ribeiro-Neto, *Modern Information Retrieval.* ACM Press/Addison-Wesley, 1999.

[3] W.H. Inmon, *Building the Data Warehouse.* John Wiley & Sons, 2005.

[4] E.F. Codd, S.B. Codd, and C.T. Salley, *Providing OLAP to User-Analysts: An IT Mandate.* Codd & Date, Inc., 1993.

[5] W. Hümmer, A. Bauer, and G. Harde, "XCube—XML for Data Warehouses," *Proc. Sixth ACM Int'l Workshop Data Warehousing and OLAP (DOLAP '03),* pp. 33-40, 2003.

[6] Microsoft Corp. and Hyperion Solutions Corp., *XML for Analysis Specification,* http://xmla.org, 2001.

[7] J. Trujillo, S. Luján-Mora, and I. Song, "Applying UML and XML for Designing and Interchanging Information for Data Warehouses and OLAP Applications," *J. Database Management,* vol. 14, no. 1, pp. 41-72, 2004.

[8] O. Mangisengi, J. Huber, C. Hawel, and W. Essmayr, "A Framework for Supporting Interoperability of Data Warehouse Islands Using XML," *Proc. Third Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK '01),* pp. 328-338, 2001.

[9] T.B. Nguyen, A.M. Tjoa, and O. Mangisengi, "MetaCube-X: An XML Metadata Foundation of Interoperability Search among Web Data Warehouses," *Proc. Third Int'l Workshop Design and Management of Data Warehouses (DMDW '01),* pp. 8.1-8.8, 2001.

[10] F. Tseng and C. Chen, "Integrating Heterogeneous Data Warehouses Using XML Technologies," *J. Information Science,* vol. 31, no. 3, pp. 209-229, 2005.

[11] T. Niemi, M. Niinimäki, J. Nummenmaa, and P. Thanisch, "Constructing an OLAP Cube from Distributed XML Data," *Proc. Fifth ACM Int'l Workshop Data Warehousing and OLAP (DOLAP '02),* pp. 22-37, 2002.

[12] T. Niemi, M. Niinimäki, J. Nummenmaa, and P. Thanisch, "Applying Grid Technologies to XML Based OLAP Cube Construction," *Proc. Fifth Int'l Workshop Design and Management of Data Warehouses (DMDW '03),* pp. 4.1-4.13, 2003.

[13] R.M. Bruckner, T.M. Ling, O. Mangisengi, and A.M. Tjoa, "A Framework for a Multidimensional OLAP Model Using Topic Maps," *Proc. Second Int'l Conf. Web Information Systems Eng. (WISE '01),* pp. 109-118, 2001.

[14] L. Xyleme, "A Dynamic Warehouse for XML Data of the Web," *IEEE Data Eng. Bull.,* vol. 24, no. 2, pp. 40-47, 2001.

[15] The Web Warehousing & Mining Group, "Whoweda,"http://www.cais.ntu.edu.sg:8000/~whoweda, 2007.

[16] M. Golfarelli, S. Rizzi, and B. Vrdoljak, "Data Warehouse Design from XML Sources," *Proc. Fourth ACM Int'l Conf. Data Warehousing and OLAP (DOLAP '01),* pp. 40-47, 2001.

[17] J. Pokorný, "Modelling Stars Using XML," *Proc. Fourth ACM Int'l Conf. Data Warehousing and OLAP (DOLAP '01),* pp. 24-31, 2001.

[18] M.R. Jensen, T.H. Møller, and T.B. Pedersen, "Specifying OLAP Cubes on XML Data," *J. Intelligent Information Systems,* vol. 17, nos. 2-3, pp. 255-280, 2001.

[19] D. Pedersen, K. Riis, and T.B. Pedersen, "XML-Extended OLAP Querying," *Proc. 14th Int'l Conf. Scientific and Statistical Database Management (SSDBM '02),* pp. 195-206, 2002.

[20] D. Pedersen, J. Pedersen, and T.B. Pedersen, "Integrating XML Data in the TARGIT OLAP System," *Proc. 20th Int'l Conf. Data Eng. (ICDE '04),* pp. 778-781, 2004.

[21] M.C. McCabe, J. Lee, A. Chowdhury, D. Grossman, and O. Frieder, "On the Design and Evaluation of a Multi-Dimensional Approach to Information Retrieval," *Proc. ACM SIGIR '00,* pp. 363-365, 2000.

[22] J. Mothe, C. Chrisment, B. Dousset, and J. Alaux, "Doccube: Multi-Dimensional Visualisation and Exploration of Large Document Sets," *J. Am. Soc. for Information Science and Technology,* vol. 54, no. 7, pp. 650-659, 2003.

[23] J.M. Pérez, R. Berlanga, M.J. Aramburu, and T.B. Pedersen, "A Relevance-Extended Multi-Dimensional Model for a Data Warehouse Contextualized with Documents," *Proc. Eighth ACM Int'l Workshop Data Warehousing and OLAP (DOLAP '05),* pp. 19-28, 2005.

[24] S. Lu, Y. Sun, M. Atay, and F. Fotouhi, "On the Consistency of XML DTDs," *Data & Knowledge Eng.,* vol. 52, no. 2, pp. 231-247, 2005.

[25] J. Kamps, M. Marx, M. de Rijke, and B. Sigurbjörnsson, "Best-Match Querying from Document-Centric XML," *Proc. Seventh Int'l Workshop the Web and Databases (WebDB '04),* pp. 55-60, 2004.

[26] D.C. Fallside and P. Walmsley, *XML Schema Part 0: Primer Second Edition,* World Wide Web Consortium (W3C) recommendation, http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/, Oct. 2004.

[27] J. Clark and S. DeRose, *XML Path Language (XPath) Version 1.0,* World Wide Web Consortium (W3C) recommendation, W3C, http://www.w3.org/TR/1999/REC-xpath-19991116, Nov. 1999.

[28] J. Robie, M.F. Fernández, D. Chamberlin, S. Boag, D. Florescu, and J. Siméon, *XQuery 1.0: An XML Query Language,* World Wide Web Consortium (W3C) candidate recommendation, http://www.w3.org/TR/2006/CR-xquery-20060608/, June 2006.

[29] S. DeRose, E. Maler, and D. Orchard, *XML Linking Language (XLink) Version 1.0,* World Wide Web Consortium (W3C) recommendation, http://www.w3.org/TR/2001/REC-xlink-20010627/, June 2001.

[30] S. Deach, T. Graham, A. Berglund, P. Grosso, J. Caruso, J. Richman, S. Adler, R.A. Milowski, E. Gutentag, S. Zilles, and S. Parnell, *Extensible Stylesheet Language (XSL) Version 1.0,* World Wide Web Consortium (W3C) recommendation, http://www.w3.org/TR/2001/REC-xsl-20011015/, Oct. 2001.

[31] M.C. Daconta, L.J. Obrst, and K.T. Smith, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management.* John Wiley & Sons, 2003.

[32] S. Pepper and G. Moore, *XML Topic Maps (XTM) 1.0,* TopicMaps.Org specification, http://www.topicmaps.org/xtm/1.0/xtm1-20010806.html, Aug. 2001.

[33] E. Miller and F. Manola, *RDF Primer,* World Wide Web Consortium (W3C) recommendation, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/, Feb. 2004.

[34] F. van Harmelen and D.L. McGuinness, *OWL Web Ontology Language Overview,* World Wide Web Consortium (W3C) recommendation, http://www.w3.org/TR/2004/REC-owl-features-20040210/, Feb. 2004.

[35] R. Kimball and M. Ross, *The Data Warehouse Toolkit.* John Wiley & Sons, 2002.

[36] J. Widom, "Research Problems in Data Warehousing," *Proc. Fourth Int'l Conf. Information and Knowledge Management (CIKM '95),* pp. 25-30, 1995.

[37] G. Spofford, *MDX Solutions with Microsoft SQL Server Analysis Services.* John Wiley & Sons, 2001.

[38] T.B. Pedersen and C.S. Jensen, "Multidimensional Databases," *The Industrial Information Technology Handbook,* R. Zurawski, ed., pp. 1-13, CRC Press, 2005.

[39] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," *SIGMOD Record,* vol. 26, no. 1, pp. 65-74, 1997.

[40] P. Ponniah, *Data Warehousing Fundamentals: A Comprehensive Guide for IT Processionals.* John Wiley & Sons, 2001.

[41] Y. Lafon and N. Mitra, *SOAP Version 1.2 Part 0: Primer (Second Edition),* World Wide Web Consortium (W3C) recommendation, http://www.w3.org/TR/2007/REC-soap12-part0-20070427/, Apr. 2007.

[42] C. Lee, C.-J. Chen, and H. Lu, "An Aspect of Query Optimization in Multidatabase Systems," *SIGMOD Record,* vol. 24, no. 3, pp. 28-33, 1995.

[43] T.B. Nguyen, A.M. Tjoa, and R. Wagner, "Conceptual Multidimensional Data Model Based on MetaCube," *Proc. First Int'l Conf. Advances in Information Systems (ADVIS '00),* pp. 24-33, 2000.

[44] A.P. Sheth and J.A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys,* vol. 22, no. 3, pp. 183-236, 1990.

[45] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure.* Morgan Kaufmann, 1998.

[46] S.S. Bhowmick, "WHOM: A Data Model and Algebra for a Web Warehouse," PhD dissertation, School of Computer Eng., Nanyang Technological Univ., 2001.

[47] C. Yinyan, E.P. Lim, and W.K. Ng, "Storage Management of a Historical Web Warehousing System," *Proc. 11th Int'l Conf. Database and Expert Systems Applications (DEXA '00),* pp. 457-466, 2000.

[48] S.S. Bhowmick, S. Mandria, and W.K. Ng, "Detecting and Representing Relevant Web Deltas in Whoweda," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 2, pp. 423-441, Mar./Apr. 2003.

[49] B. Nguyen, S. Abiteboul, G. Cóbena, and M. Preda, "Monitoring XML Data on the Web," *Proc. ACM SIGMOD '01,* pp. 437-448, 2001.

[50] G. Cóbena, S. Abiteboul, and A. Marian, "Detecting Changes in XML Documents," *Proc. 18th Int'l Conf. Data Eng. (ICDE '02),* pp. 41-52, 2002.

[51] A. Marian, S. Abiteboul, G. Cóbena, and L. Mignet, "Change-Centric Management of Versions in an XML Warehouse," *Proc. 27th Int'l Conf. Very Large Data Bases (VLDB '01),* pp. 581-590, 2001.

[52] Y. Zhuge and H. Garcia-Molina, "Graph Structured Views and Their Incremental Maintenance," *Proc. 14th Int'l Conf. Data Eng.,* pp. 116-125, 1998.

[53] R. Avnur and J.M. Hellerstein, "Eddies: Continuously Adaptive Query Processing," *Proc. ACM SIGMOD '00,* pp. 261-272, 2000.

[54] D. Pedersen and T.B. Pedersen, "Achieving Adaptivity for OLAP-XML Federations," *Proc. Sixth ACM Int'l Conf. Data Warehousing and OLAP (DOLAP '03),* pp. 25-32, 2003.

[55] D. Pedersen and T.B. Pedersen, "Synchronizing XPath Views," *Proc. Eighth Int'l Database Eng. and Application Symp. (IDEAS '04),* pp. 149-160, 2004.

[56] OMG—Object Management Group, *Unified Modeling Language (UML),* http://www.uml.org, 2004.

[57] M.R. Jensen, T.H. Møller, and T.B. Pedersen, "Converting XML DTDs to UML Diagrams for Conceptual Data Integration," *Data & Knowledge Eng.,* vol. 44, no. 3, pp. 323-346, 2003.

[58] B. Vrdoljak, M. Banek, and S. Rizzi, "Designing Web Warehouses from XML Schemas," *Proc. Fifth Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK '01),* pp. 89-98, 2003.

[59] D. Pedersen, T.B. Pedersen, and K. Riis, "The Decoration Operator: A Foundation for On-Line Dimensional Data Integration," *Proc. Eighth Int'l Database Eng. and Applications Symp. (IDEAS '04),* pp. 357-366, 2004.

[60] D. Pedersen, K. Riis, and T.B. Pedersen, "Cost Modeling and Estimation for OLAP-XML Federations," *Proc. Fourth Int'l Conf. Data Warehousing and Knowledge Discovery,* pp. 245-254, 2002.

[61] D. Pedersen, K. Riis, and T.B. Pedersen, "Query Optimization for OLAP-XML Federations," *Proc. Fifth ACM Int'l Workshop Data Warehousing and OLAP (DOLAP '02),* pp. 57-64, 2002.

[62] M. Krishnaprasad, Z.H. Liu, A. Manikutty, J. Warner, V. Arora, and S. Kotsovolos, "Query Rewrite for XML in Oracle XMLDB," *Proc. 30th Int'l Conf. Very Large Data Bases (VLDB),* 2004.

[63] S. Pal, I. Cseri, O. Seeliger, M. Rys, G. Schaller, W. Yu, D. Tomic, A. Baras, B. Berg, D. Churin, and E. Kogan, "XQuery Implementation in a Relational Database System," *Proc. 31st Int'l Conf. Very Large Data Bases (VLDB '05),* pp. 1175-1186, 2005.

[64] Z.H. Liu, M. Krishnaprasad, and V. Arora, "Native XQuery Processing in Oracle XMLDB," *Proc. ACM SIGMOD '05,* pp. 828-833, 2005.

[65] I. Sanz, J.M. Pérez, R. Berlanga, and M.J. Aramburu, "XML Schemata Inference and Evolution," *Proc. 14th Int'l Conf. Database and Expert Systems Applications (DEXA '00),* pp. 109-118, 2003.

[66] K. Beyer, D. Chambérlin, L.S. Colby, F. Özcan, H. Pirahesh, and Y. Xu, "Extending XQuery for Analytics," *Proc. ACM SIGMOD '05,* pp. 503-514, 2005.

[67] N. Wiwatwattana, H.V. Jagadish, L.V.S. Lakshmanan, and D. Srivastava, "X^3: A Cube Operator for XML OLAP," *Proc. 23rd Int'l Conf. Data Eng. (ICDE '07),* pp. 916-925, 2007.

[68] H.V. Jagadish, L.V.S. Lakshmanan, D. Srivastava, and K. Thompson, "Tax: A Tree Algebra for XML," *Revised Papers from the Eighth Int'l Workshop Database Programming Languages (DBPL '01),* pp. 149-164, 2002.

[69] O. Romero and A. Abelló, "Automating Multidimensional Design from Ontologies," *Proc. 10th ACM Int'l Workshop Data Warehousing and OLAP (DOLAP),* 2007.

[70] J.M. Pérez, R. Berlanga, and M.J. Aramburu, "Semi-Structured Information Warehouses: An Approach to a Document Model to Support their Construction," *Proc. Sixth Int'l Conf. Enterprise Information Systems (ICEIS '04),* pp. 579-582, 2004.

[71] G. Salton, A. Wong, and C.S. Yang, "A Vector Space Model for Automatic Indexing," *Comm. ACM,* vol. 18, no. 11, pp. 613-620, 1975.

[72] J.M. Ponte and W.B. Croft, "A Language Modeling Approach to Information Retrieval," *Proc. ACM SIGIR '98,* pp. 275-281, 1998.

[73] V. Lavrenko and W.B. Croft, "Relevance-Based Language Models," *Proc. ACM SIGIR '01,* pp. 120-127, 2001.

[74] A. Singahl, C. Buckley, and M. Mitra, "Pivoted Document Length Normalization," *Proc. ACM SIGIR '96,* pp. 21-29, 1996.

[75] J. Lee, D. Grossman, and R. Orlandic, "MIRE: A Multidimensional Information Retrieval Engine for Structured Data and Text," *Proc. Int'l Conf. Information Technology: Coding and Computing,* pp. 224-229, 2002.

[76] J. Lee, D. Grossman, and R. Orlandic, "An Evaluation of the Incorporation of a Semantic Network into a Multidimensional Retrieval Engine," *Proc. 12th Int'l Conf. Information and Knowledge Management (CIKM '03),* pp. 572-575, 2003.

[77] B.-K. Park, H. Han, and I.-Y. Song, "XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses," *Proc. Sixth Int'l Conf. Data Warehousing and Knowledge Discovery (DaWaK '05),* pp. 32-42, 2005.

[78] J.M. Pérez, R. Berlanga, and M.J. Aramburu, "A Document Model Based on Relevance Modeling Techniques for Semi-Structured Information," *Proc. 15th Int'l Conf. Database and Expert Systems Applications,* pp. 318-327, 2004.

[79] T.B. Pedersen, C.S. Jensen, and C.E. Dyreson, "A Foundation for Capturing and Querying Complex Multidimensional Data," *Information Systems,* vol. 26, no. 5, pp. 383-423, 2001.

[80] J.M. Pérez, T.B. Pedersen, R. Berlanga, and M.J. Aramburu, "IR and OLAP in XML Document Warehouses," *Proc. 27th European Conf. Information Retrieval Research (ECIR '05),* pp. 536-539, 2005.

[81] T. Priebe and G. Pernul, "Towards Integrative Enterprise Knowledge Portals," *Proc. 12th Int'l Conf. Information and Knowledge Management (CIKM '03),* pp. 216-223, 2003.

[82] A. Badia, "Text Warehousing: Present and Future," *Processing and Managing Complex Data for Decision Support,* J. Darmont and O. Boussaïd, eds., pp. 96-121, Idea Group Publishing, 2006.

Juan Manuel Pérez received the BS degree in computer science and the PhD degree from Universitat Jaume I, Spain, in 2000 and 2007, respectively. Currently, he is an associate lecturer at Universitat Jaume I. He is the author of a number of communications in international conferences and workshops such as DEXA, ECIR, ICDE, DOLAP, etc. His research interests include information retrieval, multidimensional databases, and Web-based technologies.

Rafael Berlanga received the BS degree in physics and the PhD degree in computer science in 1996 from the Universidad de Valencia. He is an associate professor of computer science at Universitat Jaume I, Spain. He is the author of several articles in international journals such as *Information Processing & Management*, *Concurrency: Practice and Experience*, and *Applied Intelligence* and numerous communications in international conferences such as DEXA, ECIR, CIARP, etc. His current research interests include knowledge bases, information retrieval, and temporal reasoning.

María José Aramburu received the BS degree in computer science from the Universidad Politécnica de Valencia in 1991 and the PhD degree from the School of Computer Science, University of Birmingham, United Kingdom, in 1998. She is an associate professor of computer science at Universitat Jaume I, Spain. She is the author of several articles in international journals such as *Information Processing & Management*, *Concurrency: Practice and Experience*, and *Applied Intelligence* and numerous communications in international conferences such as DEXA, ECIR, etc. Her main research interests include document databases and their applications.

Torben Bach Pedersen received MS degree in computer science from Aalborg University, Denmark, and the PhD degree in computer science from Aarhus University. He is a full professor of computer science at Aalborg University. Before joining Aalborg University, he worked in the software industry for more than six years. His research interest includes multidimensional databases, OLAP, data warehousing, federated databases, data streams, and location-based services. He has published more than 60 scientific papers on these issues in journals such as *The VLDB Journal*, *Information Systems*, and *Computer* and in conferences such as VLDB, ICDE, SSDBM, SSTD, IDEAS, ACM-GIS, ECIR, Hypertext, DOLAP, and DaWaK. He is a member of the editorial board of the *International Journal on Data Warehousing and Mining* and has served on more than 30 program committees including VLDB, ICDE, EDBT, SSDBM, and DaWaK. He is a member of the IEEE, the IEEE Computer Society, and the ACM.