Integrating Data Warehouses with Web Data: A Survey

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

November 8, 2006

TR-18

A DB Technical Report

Title	Integrating Data Warehouses with Web Data: A Survey Copyright © 2006 Juan Manuel Pérez, Rafael Berlanga, María José Aramburu and Torben Bach Pedersen . All rights reserved.
Author(s)	Juan Manuel Pérez, Rafael Berlanga, María José Aramburu and Torben Bach Pedersen
Publication History	November 2006. A DB Technical Report

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 data warehouses. The paper addresses the problem of integrating heterogeneous DWs and explains how to deal with both semi-structured and unstructured data in DWs and On-Line Analytical Processing.

1 Introduction

The Web is nowadays the World's largest source of information. The Web has brought interoperability to a wide range of different applications (e.g., web services). This success has been possible thanks to XML-based technology [24], which constitutes a means of information interchange between applications, as well as a semi-structured 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 shown very useful for effectively retrieving text-rich documents. Recently, Information Extraction techniques are also being applied to detect and query the factual data contained in the documents (e.g., Question & Answering Systems). Finally and more recently, the Web is being enriched with semantic annotations (e.g., RDF and OWL formats), allowing the retrieval and analysis of the Web contents in a more effective way in the near future.

During recent years, there has also been a large interest in DW [17] and On-Line Analytical Processing (OLAP) [9] technologies. A DW system stores historical data integrated and prepared for being analyzed by OLAP and other tools. Many companies satisfy their needs of strategic information by applying these technologies to their structured databases.

The two main goals of this paper are to review how DW and Web technolgies are being combined, and to identify the main limitations and opportunities of these approaches. Section 2 summarizes the large range of XML technologies available today. Section 3 describes how these technologies can be applied to integrate distributed heterogeneous DW systems. Section 4 introduces the problem of dealing with semi-structured data in DW and OLAP systems. Section 5 addresses unstructured data, IR and DW technologies. Finally, Section 6 provides conclusions and points to future work.

2 XML-Based Web Technology

According to the authors of the Xyleme project [63] "The Web is huge and keeps growing at a healthy peace. 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, semi-structured data models, and in particular the standardization of XML [24] for Web data exchange plays an important role and opens a wide new range of possibilities. Two main features of its semi-structured 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. Specifically, the structure of a 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. Figure 1 shows an example XML document. 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.

Other advantages of XML as a semi-structured data format are its simplicity and flexibility. Moreover, XML is free, extensible, modular, platform independent and well-supported.

A number of technologies are evolving around XML. These technologies include among others: XML Schemas [13], an alternative to DTDs that improves data typing and constraining capabilities; the XPath language [7], which is used to refer to parts of XML documents; XQuery [55], the standard query language for XML documents, which provides powerful constructs for navigating, searching and restructuring XML data; XPointer and XLink [12], which define linking mechanisms between XML documents; and XSL [11], 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 [10]. A number of languages have been proposed to describe the semantics of resources, namely: Topic Maps (XTM) [46], Resource Description Framework (RDF, RDF/S) [29] and Ontology Web Language (OWL) [62].

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

3 XML-Based DW Integration

The Internet has opened an attractive range of new possibilities for DW applications. First, 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 web site are discussed in [52]. [17] and [20] 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 web site according to his/her preferences. On the other hand, 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 the definition on XML languages to represent the data and metadata of warehouses. Afterwards, it discusses some XML-based DW integration architectures proposed in the literature.

3.1 XML Languages for 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 [16] 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); *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). Figure 1 shows an example *XCubeFact* document depicting two cells with sales made on August 3, 2005 for the products LA-123 and RS-133, respectively.

```
<business_newspaper date=‘‘Dec.1,1998’’>
<cubeFacts version=‘‘0.4’’
xmlns=‘‘http://www.xcube-open.org/V0_4/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>
```

Figure 1: Example *XCubeFact* document [16]

The work presented in [28] also includes its own XML language 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 [16] and [28] resides in their underlying multidimensional model, which in the second case is tightly related to MDX [58]. Apart from these, the authors of [60] propose a UML-based multidimensional model along with its representation in XML. In this case the XML language is only focused on metadata interchange.

3.2 XML-Based Integration Architectures

This section surveys relevant research on integrating distributed data warehouses. These proposals 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.

A framework that combines the federation and mediation architectures is presented in [25]. As Figure 2 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. These systems execute queries coming from the mediation layer and return the results to the corresponding mediator. In order to participate in the federation, each DW should provide its local schema to the corresponding mediator. At the mediation layer, each mediator module receives subqueries from the federated layer, translates them into the local DW query language, restructures the results and returns them to the federated layer. Mediators also provide the federated layer with export schemata, which are the translation of local schemata into a common canonical data model. The federated schema imports the export schemata of the local DW systems and integrates them into a single DW schema. In the federated layer, the queries of the client applications are first divided into subqueries that are issued to the corresponding mediators, and afterwards, the results are merged and returned to the client application. The applications of the client layer will access the federated warehouse using a single homogeneous interface.

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 [16]. 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.

A similar architecture was proposed in [32] but with a different underlying canonical multidimensional model called *MetaCube* [33]. 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

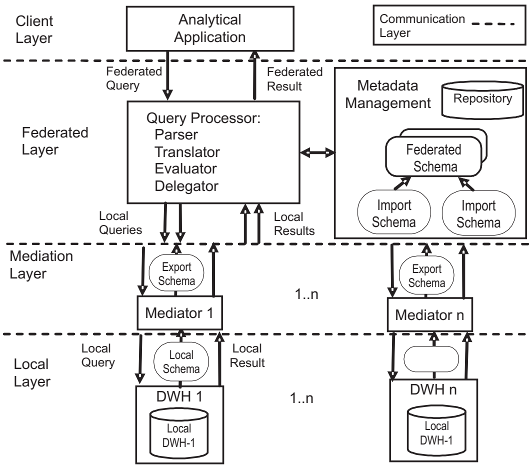

Figure 2: Federated DW architecture [25]

schemata. None of the approaches [25, 32] address query processing 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 [25], in order to completely overcome semantic heterogeneity in DW integration (e.g., different hierarchies for the same dimension) a deeper study of the mapping strategies is required.

The work made in [61] classifies the main issues that arise from the semantic integration of heterogeneous warehouses, and studies how they should be addressed. This work also proposes a federated architecture in which mediator components are replaced by native XML databases (see Figure 3). Each native XML database stores an XML version of 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 semi-automatically by studying the *site metadata*, and by designing and evaluating XQuery statements 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.

A different architecture, based on Grid technology [14], is proposed in [34, 35]. Figure 4 shows the system architecture. Analysis takes place as follows. (1) A virtual "universal" data warehouse schema representing all the data available in the warehouses is presented to the user. (2) The user establishes an analysis query. (3) The *Collection Server* analyses the query, and according to a distribution schema (i.e. how the data is distributed between the different warehouses) sends request to the relevant warehouses. (4) The involved warehouses compute the selection and aggregation calculations in parallel. A Grid-based distributed computing platform is used to perform this distributed data processing. (5) The *Collection Server* receives the data and performs a final aggregation, if needed. (6) The *Collection Server* sends the resulting cube data to the OLAP Server. (7) The user analyses the cube in the OLAP Server.

In this approach XML is used to represent the "universal" cube schema, the initial user query, the distribution schema, the data returned by each warehouse, and the final analysis cube data. The authors of [34, 35] 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. The main contribution of [34, 35] is the use of Grid technology to distribute the computation needed in the cube construction process. However, they do not show how the heterogeneity conflicts are solved.

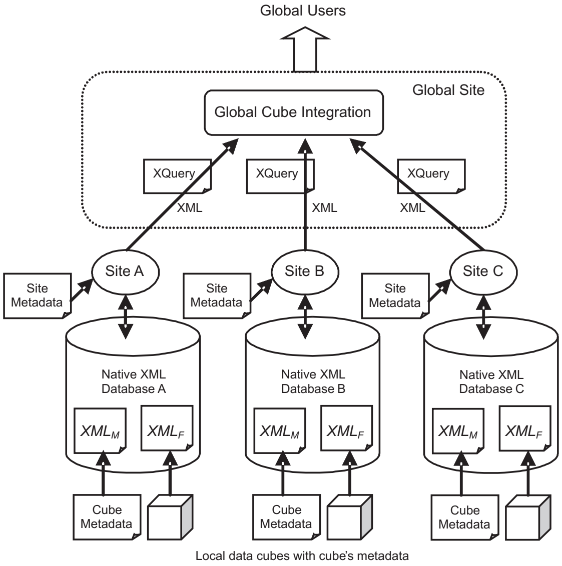

Figure 3: Federated DW architecture [61]

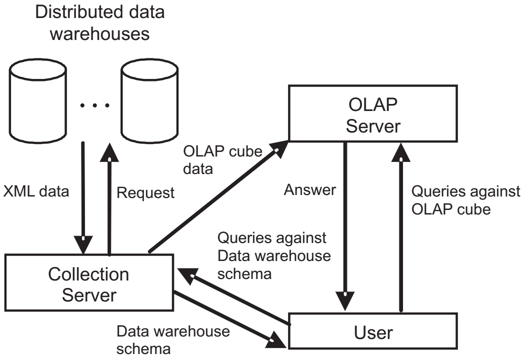

Figure 4: Distributed DW architecture proposed in [34, 35]

Although the application of XML technology has supposedly been a great advance for DW integration, so far this integration has been mostly syntactic, as it simply consists of translating DW schemata into DTD or XML files. Semantic heterogeneity discrepancies between DW schemata are still handled manually [25] or semi-automatically [61]. Trying to address these conflicts, some work has applied Semantic Web languages to describe the DW conceptual schemata. For instance, [6] follows a federation approach too, and applies Topic Maps to describe the local multidimensional schemas. Thus, the measures, dimensions and hierarchy dimension levels are represented by topics in the local topic maps. Association relations are used for modeling the facts structure (i.e., the dimensions and measures that constitute the fact) and the roll-up relationships between dimension levels. Afterwards, at the federated layer, a global topic map provides the unifying view of the local schemas. Thus, the global topic map deals with the semantic conflicts between the local schemas. 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 a topic *day* with two scopes, *English* and *German*. Then, each scope will be linked to the corresponding dimension.

4 DWs for Semi-Structured Data

With the emergence of XML as the lingua franca of the Web, semi-structured information is now widely available, and several solutions have been proposed to build warehouses for XML data. This section first introduces work oriented towards 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.

4.1 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 [63] or web documents [59]. The main issues of this research area include the efficient storage, indexing, query processing, data acquisition, 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 [63] and Whoweda [59].

Xyleme [63] 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. 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 documents 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 which satisfy a given tree pattern. Xyleme partitions the XML documents into clusters corresponding to different domains of interest (e.g. tourism, finance, etc.) which allow 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, its estimated rate change, or under the request of the owner of the document (i.e. in a notification/subscription basis).

The Whoweda (Warehouse of Web Data) project is also aimed at warehousing relevant data extracted from the Web [59]. Their efforts have been mainly focused on the definition of a formal data model and an algebra to represent and manage web documents [4], their physical storage [64] and change detection [5]. In their data model, called WHOM (Warehouse Object Model) [4], a web warehouse is conceived as a collection of web tables. The tuples of these tables are directed graphs where each node represents a document, and the edges depict hyperlinks between documents. In order to manage the data stored in the web tables, a set of algebraic operators is provided (i.e, global web coupling, web join, web select, etc.). For example, the global web coupling operator retrieves a set of inter-linked 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 is an important issue that has spawned a lot of research. Xyleme [63] allow users to subscribe to changes in an XML document [31]. When such a change occurs, subscribers receive only the changes made, called *deltas* [8, 26], and then incrementally update the old document. This approach is based on a versioning mechanism [26] and an algorithm to compute the difference between two consecutive versions of an XML document [8]. 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 inter-linked documents to ”watch”. Given two versions of this set of inter-linked 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 [65] considered a more general problem by studying how to update materialized views of graph-structured data when the sources change. In [1] an adaptative query processing technique for federated database environments was proposed. Finally, [39, 40] considers adaptivity in a federation of XML and OLAP data sources (see Section 4.3).

4.2 XML Multidimensional Database Design

This section surveys the most relevant research on multidimensional design for XML data. Specifically, the works by Golfarelli et al. [15], Pokorný [51], and Jensen et al. [18] are studied.

The authors of [15] 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 semi-automatic 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 or the XMLSchema of the XML documents. Such relationships are modeled by sub-elements nesting in DTDs and XMLSchemas, and key/keyRef in XMLSchemas. ID/IDREF(s) attributes of the DTDs are not considered, since IDREF(s) 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. Just focusing on DTDs, the authors provide an algorithm which represents the structure modeled by the DTD as a graph, and starting from a selected element (the analysis fact), semi-automatically builds the multidimensional schema by including the dimension and dimension levels depicted by the many-to-one relationships found between the elements and attributes of the graph. In order to understand the 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. They solve the problem by querying the document instances and asking the user.

In [15] it was assumed that the schema of the source XML data is provided by a single DTD or XMLSchema. In [51] 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, dimensions are modeled as sequences of logically related DTDs, and the XML-star schema is defined by considering the facts as XML elements (see Figure 5). In order to build the dimension hierarchies, this approach defines a subDTD as the portion of a source DTD that characterizes the structure of a dimension member. Then, XML view mechanisms are applied to select the members of each dimension. The concept of referential integrity for XML data is applied to establish hierarchical relationships between them.

The work in [18] deals with the conceptual design of multidimensional databases in a distributed environment of XML and relational data sources. This approach use UML diagrams [36] 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 [19] 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.

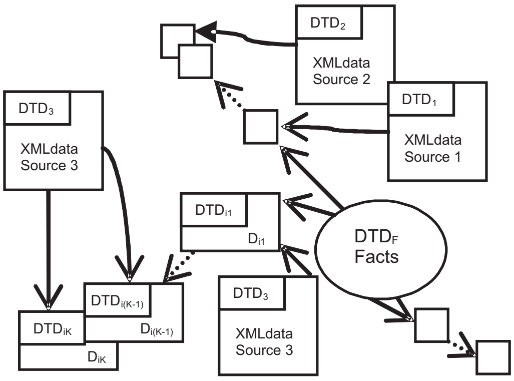

Figure 5: XML-star schema proposed in [51]

4.3 Extending OLAP Techniques to XML Data

This section mainly studies the works by Pedersen et al. on the extension of OLAP techniques to XML data [44, 38]. 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 [44] and/or measures [38] of the OLAP cubes.

In this work, OLAP-XML federations use links for relating dimension values in a cube to elements in 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* [41], which adds a new dimension to a cube based on the values of the linked XML elements. This work presents an extended multidimensional query language called SQL_{XM} that supports XPath expressions and allow linked XML data to be used for decorating, selecting and grouping fact data. For example, the query `SELECT SUM(Quantity), City/Population FROM Purchases GROUP BY City/Population` computes the total purchase quantities grouped by the city population which is found only in the XML document.

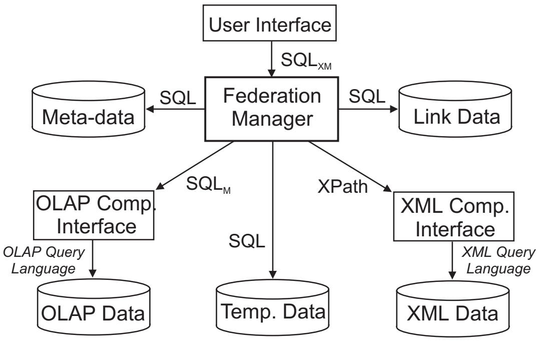

Figure 6: OLAP-XML federation architecture [44]

Figure 6 shows the architecture of the system proposed in [44]. 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). The Federation Manager receives SQL_{XM} queries and coordinates their execution in the two repositories. The metadata, link data and temporary data databases (e.g. traditional relational databases) are also managed by the Federation Manager component.

The un-optimized approach to process an SQL_{XM} query is 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 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, an XML part, and a relational part. Algebraic query rewriting rules are applied to push as much of the query evaluation towards the OLAP and XML components as possible. The cost-based optimization strategies are based on the cost model described in [42], and a set of the techniques that include in-lining literal XML data values into OLAP predicates, caching and pre-fetching [43].

In a more recent paper [38], 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.

A different approach to analyzing XML data with OLAP technology was presented in [3]. This paper proposes an extension to XQuery 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.

Notice that these proposals [3, 44, 38] deal with highly structured XML data (e.g. on-line XML product pricing lists), from where the measures and dimensions can be directly selected using XPath expressions. However, these approaches are not suitable for analyzing text-rich XML documents, which require some kind of document processing to extract measures and dimension values from their textual contents [48]. The next section deals with the combination of DW and IR technologies to exploit text-rich XML documents.

5 DWs & IR for Unstructured Data

Many new web applications store unstructured data with large text portions requiring Information Retrieval (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 result is a set of documents ranked by relevance. The relevance is a numerical value which measures how well the document fits the user information needs. Traditional IR models (e.g. the vector space model [56]) 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 [53] and relevance modeling [21] techniques. The works 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 [21] 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 [53, 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.

5.1 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 [27] implements an IR system based on a multidimensional database. As Figure 7 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 terms weights, which are obtained by slicing the cube on the terms dimension. The final relevance value is calculated by applying the so called pivoted cosine formula [57] 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 July 1998. Following this line of research, in [22] the authors study different indexing strategies to improve the performance of their system, and in [23] propose a method for incorporating a hierarchical category dimension to classify the documents by theme.

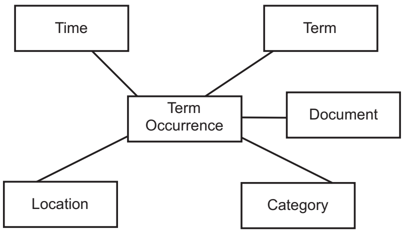

Figure 7: Multidimensional implementation of an IR system proposed in [27]

The benefits of implementing an IR system on a multidimensional database are also discussed in [30] 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 [37] 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 text-rich XML documents collection, they cannot be applied to evaluate OLAP operations over the facts described by document textual contents. This is the focus of the following section.

5.2 IR Techniques Applied to OLAP

Nowadays, 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 [63]. 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 5.1 to implement an IR system by using a multidimensional database are very useful to explore a text-rich XML documents 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 4.3 are not suitable for analyzing text-rich documents either. They only deal with highly structured XML data (e.g., on-line 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 [48] is needed.

The authors of [49] 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.

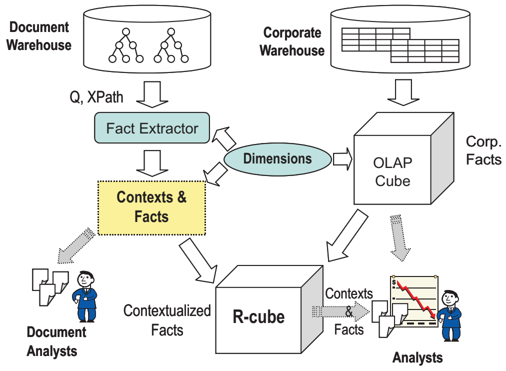

Figure 8: Contextualized warehouse architecture [49]

Figure 8 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 data warehouse 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 to retrieve the documents that describe the analysis context and to estimate the relevance of the facts described by these documents to the analysis context (IR query) was presented in [47]. The data model and algebra for the R-cubes is described in [49] and extends the multidimensional model of [45]. Finally, in [50] they prosed a system implementation based on multidimensional databases.

From a different point of view, the work presented in [54] 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 data warehouse’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 [49] it does not provide a formal framework for calculating fact relevance with respect to user queries.

6 Conclusion and Future Work

The advent of XML and related technologies is playing an important role in the future development of the Web. DW and OLAP tools take part in the Web revolution. This paper has summarized the most relevant research on combining DW and Web/XML data. As far as we know there not exists any similar survey.

The paper has studied the advantages of XML as an integration tool for heterogeneous and distributed DW systems. In this sense, it has first described work focused on the definition of XML languages to represent warehouses data and metadata, and then discussed different XML-based data warehouse integration architectures. It has also addressed the construction of warehouses for semi-structured XML web data. Specifically, it has introduced some work oriented towards the construction of XML web data repositories, the research done on the design of multidimensional databases for XML data, and the extension of OLAP techniques for analyzing external XML data. As most information is nowadays published on the Web as unstructured (in the near future text-rich XML) documents, the paper finally showed how IR and OLAP technologies can be combined to explore text-rich documents collections, (i.e., the use of multidimensional databases for implementing IR systems), and to analyze facts and documents together in the so-called contextualized warehouses.

In the future, with the Semantic Web widely adopted, companies will be able to gather huge amounts of valuable semantically-related metadata concerning their subjects of interest. All this information will be used to create metadata warehouses for global decision-making. As far as we know, currently there does not exist any approach to build data warehouses for the metadata generated by the Semantic Web.

7 Acknowledgements

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 no. 26-02-0277.

References

[1] R. Avnur and J. M. Hellerstein. Eddies: Continuously Adaptive Query Processing. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 261–272. ACM Press, New York, NY, 2000.

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

[3] K. Beyer, D. Chambérlin, L. S. Colby, F. Özcan, H. Pirahesh, and Y. Xu. Extending XQuery for analytics. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 503–514. ACM Press, New York, NY, 2005.

[4] S. S. Bhowmick. *WHOM: A Data Model and Algebra for a Web Warehouse*. PhD thesis, School of Computer Engineering, Nanyang Technological University (Singapore), 2001.

[5] S. S. Bhowmick, S. Mandria, and W. K. Ng. Detecting and Representing Relevant Web Deltas in Whoweda. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):423 – 441, 2003.

[6] R. M. Bruckner, T. M. Ling, O. Mangisengi, and A. M. Tjoa. A Framework for a Multidimensional OLAP Model using Topic Maps. In *Proceedings of the 2nd International Conference on Web Information Systems Engineering*, pages 109–118. IEEE Computer Society, Washington, DC, 2001.

[7] J. Clark and S. DeRose. XML path language (XPath) version 1.0. W3C recommendation, W3C, Nov. 1999. http://www.w3.org/TR/1999/REC-xpath-19991116.

[8] G. Cóbena, S. Abiteboul, and A. Marian. Detecting changes in XML documents. In *Proceedings of the 18th International Conference on Data Engineering*, pages 41–52. IEEE Computer Society, Washington, DC, 2002.

[9] E. F. Codd. Providing OLAP to user-analysts: An IT mandate, 1993.

[10] 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*. Wiley Publishing Inc., 2003.

[11] 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. W3C recommendation, W3C, Oct. 2001. http://www.w3.org/TR/2001/REC-xsl-20011015/.

[12] S. DeRose, E. Maler, and D. Orchard. XML linking language (XLink) version 1.0. W3C recommendation, W3C, June 2001. http://www.w3.org/TR/2001/REC-xlink-20010627/.

[13] D. C. Fallside and P. Walmsley. XML schema part 0: Primer second edition. W3C recommendation, W3C, Oct. 2004. http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/.

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

[15] M. Golfarelli, S. Rizzi, and B. Vrdoljak. Data warehouse design from XML sources. In *Proceedings of the 4th ACM international conference on Data warehousing and OLAP*, pages 40–47. ACM Press, New York, NY, 2001.

[16] W. Hümmer, A. Bauer, and G. Harde. XCube - XML For Data Warehouses. In *Proceedings of the 6th ACM international workshop on Data warehousing and OLAP*, pages 33–40. ACM Press, New York, NY, 2003.

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

[18] M. R. Jensen, T. H. Møller, and T. B. Pedersen. Specifying OLAP Cubes on XML Data. *Journal of Intelligent Information Systems*, 17(2/3):255 – 280, 2001.

[19] M. R. Jensen, T. H. Møller, and T. B. Pedersen. Converting XML DTDs to UML diagrams for conceptual data integration. *Data & Knowledge Engineering*, 44(3):323 – 346, 2003.

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

[21] V. Lavrenko and W. B. Croft. Relevance-Based Language Models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM Press, New York, NY, 2001.

[22] J. Lee, D. Grossman, and R. Orlandic. MIRE: A Multidimensional Information Retrieval Engine for Structured Data and Text. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, pages 224–229. IEEE Computer Society, Washington, DC, 2002.

[23] J. Lee, D. Grossman, and R. Orlandic. An Evaluation of the Incorporation of a Semantic Network into a Multidimensional Retrieval Engine. In *Proceedings of the 12th international conference on Information and knowledge management*, pages 572–575. ACM Press, New York, NY, 2003.

[24] E. Maler, T. Bray, J. Paoli, F. Yergeau, and C. M. Sperberg-McQueen. Extensible markup language (XML) 1.0 (fourth edition). W3C recommendation, W3C, Aug. 2006. http://www.w3.org/TR/2006/REC-xml-20060816.

[25] O. Mangisengi, J. Huber, C. Hawel, and W. Essmayr. A Framework for Supporting Interoperability of Data Warehouse Islands Using XML. In *In Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery*, pages 328–338. Springer, Berlin, 2001.

[26] A. Marian, S. Abiteboul, G. Cóbena, and L. Mignet. Change-centric management of versions in an XML warehouse. In *Proceedings of the 27th International Conference on Very Large Data Bases*, pages 581–590. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2001.

[27] 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. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 363–365. ACM Press, New York, NY, 2000.

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

[29] E. Miller and F. Manola. RDF primer. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[30] J. Mothe, C. Chrisment, B. Dousset, and J. Alaux. Doccube: Multi-dimensional visualisation and exploration of large document sets. *Journal of the American Society for Information Science and Technology*, 54(7):650–659, 2003.

[31] B. Nguyen, S. Abiteboul, G. Cóbena, and M. Preda. Monitoring XML data on the web. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 437–448. ACM Press, New York, NY, 2001.

[32] T. B. Nguyen, A. M. Tjoa, and O. Mangisengi. Meta Cube-X: An XML Metadata Foundation of Interoperability Search among Web Data Warehouses. In *Proceedings of the Third International Workshop on Design and Management of Data Warehouses*, pages 8.1–8.8. CEUR-WS.org., 2001.

[33] T. B. Nguyen, A. M. Tjoa, and R. Wagner. Conceptual Multidimensional Data Model Based on MetaCube. In *Proceedings of the First International Conference on Advances in Information Systems*, pages 24–33. Springer, Berlin, 2000.

[34] T. Niemi, M. Niinimäki, J. Nummenmaa, and P. Thanisch. Constructing an OLAP Cube from Distributed XML Data. In *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP*, pages 22–37. ACM Press, New York, NY, 2002.

[35] T. Niemi, M. Niinimäki, J. Nummenmaa, and P. Thanisch. Applying Grid Technologies to XML Based OLAP Cube Construction. In *Proceedings of the 5th International Workshop on Design and Management of Data Warehouses*, pages 4.1–4.13. CEUR-WS.org., 2003.

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

[37] B.-K. Park, H. Han, and I.-Y. Song. XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses. In *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery*, pages 32–42. Springer, Berlin, 2005.

[38] D. Pedersen, J. Pedersen, and T. B. Pedersen. Integrating XML Data in the TARGIT OLAP System. In *Proceedings of the 20th International Conference on Data Engineering*, pages 778–781. IEEE Computer Society, Washington, DC, 2004.

[39] D. Pedersen and T. B. Pedersen. Achieving Adaptivity for OLAP-XML Federations. In *Proceedings of the 6th ACM international conference on Data warehousing and OLAP*, pages 25–32. ACM Press, New York, NY, 2003.

[40] D. Pedersen and T. B. Pedersen. Synchronizing XPath Views. In *Proceedings of the 8th International Database Engineering and Application Symposium*, pages 149–160. IEEE Computer Society, Washington, DC, 2004.

[41] D. Pedersen, T. B. Pedersen, and K. Riis. The Decoration Operator: A Foundation for On-Line Dimensional Data Integration. In *Proceedings of the International Database Engineering and Applications Symposium*, pages 357–366. IEEE Computer Society, Washington, DC, 2004.

[42] D. Pedersen, K. Riis, and T. B. Pedersen. Cost Modeling and Estimation for OLAP-XML Federations. In *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery*, pages 245–223. Springer, Berlin, 2002.

[43] D. Pedersen, K. Riis, and T. B. Pedersen. Query Optimization for OLAP-XML Federations. In *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP*, pages 57–64. ACM Press, New York, NY, 2002.

[44] D. Pedersen, K. Riis, and T. B. Pedersen. XML-Extended OLAP Querying. In *Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, pages 195–206. IEEE Computer Society, Washington, DC, 2002.

[45] T. B. Pedersen, C. S. Jensen, and C. E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Information Systems*, 26(5):383–423, 2001.

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

[47] J. M. Pérez, R. Berlanga, and M. J. Aramburu. A Document Model Based on Relevance Modeling Techniques for Semi-structured Information. In *Proceedings of the 15th International Conference on Database and Expert Systems Applications*, pages 318–327. Springer, Berlin, 2004.

[48] J. M. Pérez, R. Berlanga, and M. J. Aramburu. Semi-structured Information Warehouses: An approach to a document model to support their construction. In *Proceedings of the 6th International Conference on Enterprise Information Systems*, pages 579–582, 2004.

[49] 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. In *Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, pages 19–28. ACM Press, New York, NY, 2005.

[50] J. M. Pérez, T. B. Pedersen, R. Berlanga, and M. J. Aramburu. IR and OLAP in XML Document Warehouses. In *Proceedings of Advances in Information Retrieval: 27th European Conference on IR Research*, pages 536 – 539. Springer, Berlin, 2005.

[51] J. Pokorný. Modelling Stars Using XML. In *Proceedings of the 4th ACM international conference on Data warehousing and OLAP*, pages 24–31. ACM Press, New York, NY, 2001.

[52] P. Ponniah. *Data Warehousing Fundamentals: A Comprehensive Guide for IT Processionals*. Wiley, 2001.

[53] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM Press, New York, NY, 1998.

[54] T. Priebe and G. Pernul. Towards Integrative Enterprise Knowledge Portals. In *Proceedings of the 12th international Conference of Information and Knowledge Management*, pages 216–223. ACM Press, New York, NY, 2003.

[55] J. Robie, M. F. Fernández, D. Chamberlin, S. Boag, D. Florescu, and J. Siméon. XQuery 1.0: An XML query language. Candidate recommendation, W3C, June 2006. http://www.w3.org/TR/2006/CR-xquery-20060608/.

[56] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[57] A. Singahl, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM Press, New York, NY, 1996.

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

[59] The Web Warehousing & Mining Group. Whoweda. http://www.cais.ntu.edu.sg:8000/˜whoweda.

[60] J. Trujillo, S. Luján-Mora, and I. Song. Applying UML and XML for Designing and Interchanging Information for Data Warehouses and OLAP Applications. *Journal of Database Management*, 14(1):41 – 72, 2004.

[61] F. Tseng and C. Chen. Integrating Heterogeneous Data Warehouses Using XML Technologies. *Journal of Information Science*, 31(3):209 – 229, 2005.

[62] F. van Harmelen and D. L. McGuinness. OWL web ontology language overview. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[63] L. Xyleme. A dynamic warehouse for XML data of the Web. *IEEE Data Engineering Bulleting*, 24(2):40 – 47, 2001.

[64] C. Yinyan, E. P. Lim, and W. K. Ng. Storage Management of a Historical Web Warehousing System. In *Proceedings of 11th International Conference on Database and Expert Systems Applications*, pages 457–466. Springer, Berlin, 2000.

[65] Y. Zhuge and H. Garcia-Molina. Graph Structured Views and their Incremental Maintenance. In *Proceedings of the 14th International Conference on Data Engineering*, pages 116–125. IEEE Computer Society, Washington, DC, 1998.