Designing and Implementing OLAP Systems from XML Documents

Franck Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh

Abstract There has been a lot of research on OLAP (On-Line Analytical Processing) systems during the past decade. These systems allow decision makers to improve their decisions. Despite numerous multidimensional conceptual models, none tackle the problem of analysing data extracted from text-rich XML documents. These documents represent a lot of unavailable information for actual OLAP systems. Moreover, the implementation of such a system requires an adapted design process. In this paper, we present an adapted "galaxy" model for the analysis of text-rich XML documents. This model is associated to an adapted design process and a tool that takes in charge all automated tasks of the process.

1 Introduction

OLAP (On-Line Analytical Processing) systems allow decision makers to gain insight within enterprise performance by consulting and analysing aggregated historical business data [8]. These systems are based on well mastered techniques that allow numeric centric information analysis [35] within specially designed multidimensional databases (MDB). Nevertheless, according to a recent study [36], only 20% of corporate information system data is transactional, i.e. numeric data. The remaining 80% (mainly text-rich documents) stays out of reach of OLAP technology. This is due to the lack of adapted OLAP systems and methods in order to be able to take into account non numerical indicators such as textual indicators. Not taking into account these data sources may lead to erroneous decisions [36]. Recently XML [39] technology has provided a large framework for sharing documents within corporate networks of over the Web. Textual data in XML format is now a conceivable data source for OLAP systems.

Franck Ravat, Olivier Teste, Ronan Tournier, Gilles Zurfluh
IRIT (UMR5505), Université de Toulouse, 118 rte de Narbonne, F-31062 Toulouse Cedex 9, FRANCE. e-mail: {ravat, teste, tournier, zurfluh}@irit.fr

Our goal is to integrate these text-rich data sources within an OLAP system. More precisely, we intend to offer a complete design method for OLAP systems built on documents. This method must rest on: a multidimensional model sufficiently rich for integrating textual data; a design approach taking into account both user requirements and available data sources (documents); and a CASE[1] tool. From this goal, several issues arise: 1) to offer a global solution that includes a model, an approach and a tool; 2) to ensure the integration of all the information that is in documents (content, structure and metadata); and 3) the necessity not to restrain the user by the analysis of document data with only numeric indicators (document contents being mainly composed of textual data).

Definition 1. We define *XML Document Warehousing* as a storage stage (e.g. a data warehouse) adapted to handle text-rich XML document data and associated to an analysis stage allowing OLAP style processing on non-numeric analysis indicators.

1.1 Related works

According to the authors of [10] two types of XML documents may be encountered: 1) *Data-centric XML documents* that are raw data documents close to database content, where XML tags are used to separated data as lines and columns would in a database (e.g. logs, invoices, dumps of databases ...); and 2) *Document-centric XML documents* also known as text-rich XML documents which are the digital equivalent of our traditional paperwork (e.g. scientific articles, e-books...).

We divide related works in two categories: multidimensional modelling and design processes.

1.1.1 Multidimensional Modelling

We consider four subcategories for modelling. The first one is related to *traditional multidimensional modelling* [18]. A recent survey may be found in [33] and current issues are highlighted in [34]. Multidimensional modelling is based on the concepts of facts (analysis subjects) and dimensions (analysis axes). All these models, conceptually or logically oriented, have been conceived for traditional numeric data analysis and do not deal with documents.

The second subcategory concerns *multidimensional modelling for the analysis of data-centric XML documents*. The analysis of documents like logs or outputs of Web services has been introduced in several articles such as [16]. See [24, 38, 40] for a more complete list of works. Although these articles consider textual data through the use of XML documents, these propositions limit themselves to numeric indicators do not take into consideration more complex data.

[1] CASE : Computer Aided Software Engineering.

The third subcategory concerns *multidimensional modelling allowing complex data analysis*. In [24] the authors define an xFACT, a complex hierarchical structure containing structured and unstructured data (such as documents). Measures, called contexts, may be seen as complex objects. In [3], a complete XML approach for modelling complex data analysis is presented. These works do not take into account more complex text-rich XML documents and are limited to numeric indicators.

Nevertheless some works focussed on text-rich documents and the fourth subcategory concerns *multidimensional modelling for the analysis of document-centric documents*. In [27] the authors combine traditional numeric analysis and information retrieval techniques to assist analyses by providing documents relevant to the ongoing analysis context. In [17, 22, 23, 36], the authors present applications of document-centric document analysis using a star schema [18] but still with numeric analysis indicators. Recently, in [26], the concept of multidimensional document analysis was introduced using an xFACT [24] and specific aggregation functions inspired by text mining techniques but these are not detailed. These articles do not take into account complex document-centric document properties (e.g. structure and complete content). Moreover, they are no detailed adapted formal conceptual model.

So far, and to our knowledge, there is no proposal for designing and implementing an OLAP system that allows document-centric document content analysis. Up to now, apart from [26], research has been based on quantitative analyses, e.g. the number of publications that contain a specific keyword. Textual data is provided for the analysis through dimensions which model analysis axes and not subjects of analysis, moreover analysis indicators (measures) are always numeric.

1.1.2 Design Processes

To our knowledge, design processes have only be specified for traditional multidimensional models. Their goal is to implement a multidimensional database and they may be classified in three approaches:

Firstly, *bottom-up approaches* are data driven approaches. Multidimensional schemas are derived from the analysis of available data sources without considering user needs. In [11] and [5], a multidimensional schema is built from E/R schemas of the data sources, taking full advantage of the data sources' semantics. But, as the domain represented may be broad, this may require a great deal of resources and time. To get round this problem, in [14], the authors provide a method for the user to designate relevant sources; but, users have to consult the data source schemas and they might not master them. In [21], Architecture Driven Modernization (ADM) is used to derive semi-automatically a multidimensional conceptual schema from sources. However, in this approach, user requirements are not much considered.

Secondly, *top-down approaches* are demand driven approaches. Multidimensional schemas are derived from the analysis of user requirements. In this approach data sources are not taken into account. In [18], the author presents a general methodology for managing decisional projects but with no detailed process for the specification of multidimensional schemas. In [29], the authors present a design

process that rests on UML notations. With this design process, the designer translates user requirements in a UML class diagram. This diagram is then enriched and transformed into a multidimensional conceptual schema. But, without taking into account data sources, it is possible to design inconsistent schemas that will lack data in the sources.

Thirdly, *mixed approaches* combine both previous design processes, taking into account both user requirements and data sources. Within this approach, user requirements are translated into one (possibly more) "ideal" multidimensional schema and the analysis of the data sources also produces "candidate" multidimensional schemas. A confrontation phase allows the comparison between the different multidimensional schemas and allows the designer to come up with a final schema. In [6], the authors present a method that designs several ideal schemas. In [2, 7, 28], the authors present a method that generates several candidate schemas. But within these works, the confrontation phase may be tedious. In [20], one ideal schema is confronted to one candidate schema. All these approaches are not formally described.

Some approaches, such as [37], settle for ETL processes (Extract, Transform, Load) and focus on data integration but with XML documents, the handled data structures are different. All these approaches have been conceived for models based on the duality of the fact and dimension concepts and these models are not adapted to represent analyses based on text-rich XML documents [31]. As a consequence, to our knowledge, there is a need for adapting an approach and a complete design process with different levels of abstraction (conceptual, logical and physical) is still an open issue.

1.2 Objectives and contributions

The objectives of this paper are: 1) to allow the analysis of the contents of text-rich documents and their metadata with the use of numeric and textual indicators; 2) to allow multidimensional analysis of data extracted from text-rich XML documents; and 3) to offer a complete solution for the design and the implementation of such a multidimensional database.

The contributions are: 1) a model that allows an easy representation of the multidimensional elements available for expressing analyses; 2) an approach that enables the integration of XML documents within the decisional system; and 3) a tool to ease the design and implementation process.

This paper is organised as follows: Section 2 defines a multidimensional model adapted for expressing analysis on document-centric documents; section 3 specifies an approach for designing and implementing a multidimensional database from XML documents; and section 4 presents an adapted design tool.

2 Multidimensional Model for XML Document Data Analysis

In order to allow the analysis of document-centric XML documents, several characteristics have to be taken into account: 1) the hierarchical structure of the textual data of the documents; 2) the content mainly composed of textual data; and 3) the metadata associated to the document. Moreover, the model should not restrain the decision maker by offering limited or pre-specified analysis solutions based on numerical analyses. In the past decade, numerous multidimensional models have been proposed, but none handle these issues (see section 1.1).

2.1 Conceptual Model

To answer to these issues, we defined a multidimensional model nicknamed "galaxy" model [31]. The galaxy allows flexibility in the specification of multidimensional analyses by not restraining the decision maker with predefined analysis subjects.

A dimension is composed of hierarchically organised *attributes*. These attributes allow the representation of the multidimensional aspect of the data, each attribute being a graduation of the analysis axis, i.e. detail levels or granularity levels.

Definition 2. A *dimension* $D = (A^D, H^D, I^D, I^{StarD})$ where:

- $A^D = \{a^{D_1}, \ldots, a^{D_r}\}$ is a set of *attributes*,
- $H^D = \{H^{D_1}, \ldots, H^{D_s}\}$ is a set of *hierarchies*,
- $I^D = \{i^{D_1}, \ldots, i^{D_t}\}$ is a set of *dimension instances*. Each attribute has a value for each instance $a_u^D(i_x^D)$, called an *attribute instance*.
- $IStar^D = \{IStar_1^D, IStar_2^D \ldots\}$ is set of functions $IStar_i^D : I^D \rightarrow (I_1^D)^* \times \ldots \times (I_n^D)^*$ each associating the instances of the D dimension to the instances of other linked dimensions through $Star^G$ ($\forall k \in [1..n]$, $D_k \in D^G$, $D_k \neq D$ and $D_k \in Star^G(D)$, i.e. D_k is associated/linked to D).[2]

Within the hierarchies that organise the dimension attributes, there exist two types of attributes: *parameters* which represent a particular level of detail and *weak attributes* which represent complementary data of a parameter.

Definition 3. A *hierarchy* noted H_i^D or $H = (Param^H, Weak^H)$ where:

- $Param^H = \langle p_1^H, \ldots, p_{n_p}^H \rangle$ is an ordered set of attributes, called *parameters*, which represent the levels of granularity of the dimension, $k \in [1..n_p]$, $p_k^H \in A^D$ and $p_1^H = a_1^D$;
- $Weak^H : Param^H \rightarrow 2^{A^D - Param^H}$ is an application possibly associating *weak attributes* to parameters, completing the parameter semantic.

[2] The notation $(I)^*$ represents a finite set of elements of I.

The galaxy is based on: 1) a unique concept of dimension that represents not only an analysis axis but also a possible subject of analysis (namely a fact); 2) a gathering of dimensions into specific groups indicating compatibility for the specification of analyses with the use of these dimensions.

Definition 4. A *Galaxy* $G = (D^G, Star^G, Lk^G)$ where:

- $D^G = \{D_1, \ldots, D_n\}$ is a set of *dimensions*,
- $Star^G : D_i \rightarrow 2^D_j$ is a function that associates each dimension D_i to its linked dimensions $D_j \in D^G$ $(D_j \neq D_i)$. This expression models nodes c_z that may be expressed through: $\{D_{c_1}, \ldots, D_{c_n}\} \subseteq D^G | \forall i, j \in [c_1..c_n], i \neq j, \exists D_i \rightarrow 2^{D_j} \in Star^G$. This represents dimensions compatible within a same analysis.[3]
- $Lk^G = \{g_1, g_2, \ldots\}$ is a set of functions that associate attributes together. Due to space restriction, these will not be detailed (consult [31] for more details).

Note that each attribute of a dimension is a possible analysis indicator.

2.2 Case Study

This application aims at analysing the performance of research institutes. More precisely, we wish to analyse scientific articles and research project reports produced by researchers. There is a need for two analyses: firstly, the analysis of research articles published at a certain date, in conferences and by authors; and secondly, the analysis of reports published by authors at a certain date. The result is a galaxy composed of 5 dimensions grouped into 2 groups (nodes).

Let $G_1 = (D^{G_1}, Star^{G_1}, Lk^{G_1})$, where:

$D^{G_1} = \{CONFERENCES, ARTICLES, TIME, AUTHORS, REPORTS\}$;

$Star^{G_1} = \{CONFERENCES \rightarrow (ARTICLES, TIME, AUTHORS), ARTICLES \rightarrow (CONFERENCES, TIME, AUTHORS), TIME \rightarrow (CONFERENCES, ARTICLES, AUTHORS, REPORTS), AUTHORS \rightarrow (CONFERENCES, ARTICLES, TIME, REPORTS), REPORTS \rightarrow (TIME, AUTHORS)\}$;

In these research institutes, authors are described by their name, status (professor,...), research team and institute (or company). Let the *AUTHORS* dimension (noted D_A for short) with two hierarchies be: $D_A = (A^{D_A}, H^{D_A}, I^{D_A}, IStar^{D_A})$, where:

$A^{D_A} = \{Author, Name, Status, Team, Institute\}$,

$H^{D_A} = \{HA, HSt\}$,

$I^{D_A} = \{author_1, author_2, \ldots\}$,

$IStar_1^{D_A} = \{author_1 \rightarrow (conference_1, paragraph_{125}, date_{10}), \ldots\}$.

$IStar_2^{D_A} = \{author_1 \rightarrow (date_{23}, report_2), \ldots\}$.

Associated to this definition we propose a graphical formalism in order to easily represent a galaxy (see Fig. 1). Each dimension is represented by a rectangle whose hierarchies are represented by a graph. Dimensions are grouped into one or more

[3] The notation 2^E represents the *powerset* of E.

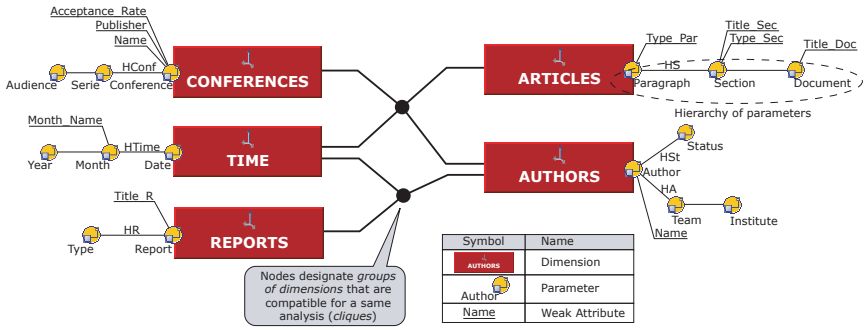

Fig. 1 Analysis of scientific publications and research reports.

sets (nodes on the graphic representation) in order to represent the different dimensions compatible for an analysis within a galaxy. The galaxy being represented by a graph, these groups or nodes are called cliques[4]. Note that in this example, some dimensions are similar to traditional OLAP dimensions (*CONFERENCES*, *TIME*, *REPORTS* and *AUTHORS*) except that their attributes may be designated as analysis subjects. The other dimension (*ARTICLES*) holds the complete content of the scientific article, i.e. text-rich documents. Its parameters are composed of large quantity of textual data.

The galaxy model has the advantage of being able to easily represent the textual content of XML text-rich documents as well as their structure in order to facilitate the specification of multidimensional analyses [31].

2.3 OLAP analysis with a galaxy

Within this subsection, we provide an analysis example using the galaxy. This is to show that how the galaxy supports analysis specifications. These are expressed with the use of adapted aggregation functions for textual data [26, 30], as well as adapted multidimensional operations. In [31] we have proposed a set of algebraic operations for specifying analyses by manipulating the concepts modelled by the galaxy. Through a FOCUS operation an analysis subject is designated and data is extracted from the galaxy into a multidimensional table (mTable), i.e. a bi-dimensional table [12]. mTables display a subject of analysis in its centre cells according to two analysis axes (one in columns and the other in lines). They act as inputs for a closed set of manipulation operations for modifying analyses (SELECT, DRILLDOWN, ROLLUP and ROTATE).

For example, in order to analyse the production of articles, a decision maker could make a simple analysis and count the number of articles by author and by con-

[4] In a graph, a *clique* is a complete subgraph, i.e. every vertex is connected every other vertex in the subgraph.

Table 1 Left, the number of published articles per author and per conference; right, the same analysis but with the top keywords of the corresponding publications.

COUNT(ARTICLES.Document)	AUTHORS			
	Institute	Inst1		
	Author	A1	A2	A3
CONFERENCES Name				
ER		3	2	1
SSDBM		2	-	-
DaWaK		1	1	2

TOP_KEYWORD(ARTICLES.Document)	AUTHORS			
	Institute	Inst1		
	Author	A1	A2	A3
CONFERENCES Name				
ER		XML, Docments	XML, Data Warehouse	Data Mining, Clustering
SSDBM		XML, Temporal DB	-	-
DaWaK		Data mining	Data mining	Data Mining, Clustering

ference as in any existing OLAP frameworks (table 1, left). But with the use of the galaxy and associated operations, the decision maker can go further and analyse the subjects that concern these publications (table 1, right). The subject of the publications is obtained with the use of an adapted aggregation function (TOP_KEYWORD) on the *Document* attribute of the *ARTICLES* dimension (i.e. the complete textual content of the articles) and that returns here the two major keywords for each set of article [26].

The following sections detail our approach that allows a user to create a galaxy schema and then to load it with data extracted from XML documents.

3 Design Process

The objective of the design process is to allow the design and the implementation of a multidimensional database modelled with a galaxy. But, several issues have to be solved: How are user requirements (analysis requirements) translated into a galaxy? How may available XML documents within the data sources be taken into account? Up to now and to our knowledge, there is no design process that considers text-rich (i.e. document-centric) XML documents as data sources for OLAP systems.

Due to the assets stated previously, our design process rests upon a mixed approach, taking into account user requirements as well as XML documents that are in the data sources. This mixed approach uses an iterative process in order to refine the galaxy schema. We defined three phases for our approach:

- Concurrent analysis of user requirements and document sources (steps 1 and 2);
- Confrontation of the outputs of the two previous analysis phases in order to detect and to solve incompatibilities (step 3, 4a and 4b);
- Implementation of a multidimensional database described with a galaxy schema and loaded with data extracted from XML documents (step 5).

All three phases are summarized in Fig. 2 (steps are numbers in brackets). The designer starts by analysing concurrently the user requirements, producing a galaxy schema (step 1), and the data sources, producing their description (step 2). This first phase allows the creation of a dictionary. With the use of this dictionary, within a confrontation phase (step 3), the designer then makes certain that compatibility is ensured between both results of the analysis phase by comparing the galaxy and the

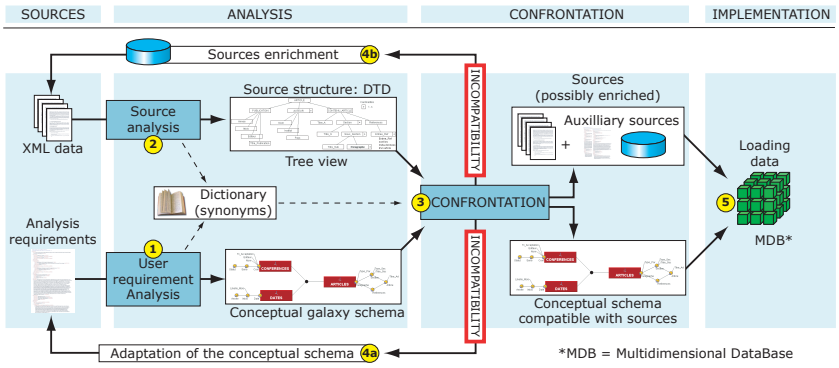

Fig. 2 Left, the number of published articles per author and per conference; right, the same analysis but with the top keywords of the corresponding publications.

sources. In case of incompatibility, the designer modifies either the galaxy (step 4a) or the sources (step 4b) or even both and the process is iterated until no more incompatibilities arise. After a successful confrontation, the process continues by loading data within the multidimensional database (step 5). These phases are described in the following subsections.

3.1 Analysis Phase

The analysis phase is composed of two concurrent steps: on the one hand, the analysis of the user requirements (or analysis requirements) in order to specify a galaxy schema and on the other hand the analysis of the available data sources (text-rich XML documents).

3.1.1 User requirements analysis

The objective of this step is to produce a conceptual schema (a galaxy) as close as possible to the user requirements. This schema represents the multidimensional structures of the database that are used for expressing OLAP analyses. Questionnaires and typical analytical queries are used as input.

Typical queries represent the multidimensional analyses that a decision maker wants to specify. They are expressed with the use of a pseudo query language. In such queries, some elements (subjects) are analysed according to other elements (axes) with possible restrictions: "*Analyse what subject according to which analysis axes for what data*" (see example displayed in Fig. 3). The elementary information of each clause of these queries is converted into dimension attributes of the future

Q1: **Analyse** the number of references **according to** the name of the author of these references and their institute **and according to** the name of conferences where the articles contain these references for authors of the institute inst1.	Q3: **Analyse** the number of articles **according to** the name of the author **and according to** the years of publication **for** publications in a conference of interna-tional audience.
Q2: **Analyse** the content of articles **according to** the author's name **and according to** the year of publication of the article **for** article contents limited to section of the type introduction.	Q4: **Analyse** the number of project reports **according to** the authors (name, status and institute) **and according to** the year of publication of the report **for** reports of scientific type.

Fig. 3 Examples of typical queries.

galaxy schema. For example, in Q3, the following attributes are identified: *number of articles, authors' name, publication year, conference audience*.

Questionnaires are the result of user interviews that will provide valuable information on the domain of expertise. More precisely, this information is used to regroup attributes within dimensions and to organize them hierarchically. For example, in the four previous queries, it is possible to detect several attributes that concern authors and that may be regrouped together into an *AUTHORS* dimension: *authors, name, team, institute, status*. Complementary domain information, extracted from the questionnaires, allows the organisation of these attributes into 2 hierarchies (see Fig. 1). This information will also allow regrouping dimensions into one or more cliques (e.g. two cliques in the galaxy presented in Fig. 1).

The user requirement analysis step is summarised in Fig. 4. For the rest of the paper, we shall use a galaxy schema composed of a unique clique and three dimensions: *CONFERENCES*, *ARTICLES* and *TIME*.

Once the galaxy is created, all the attributes' (parameters and weak attributes) semantics are integrated within the dictionary for synonymy differentiation. For example, there are two attributes *Name*: one for a conference, the other of an author. The dictionary can be completed with words similar to those used to designate attributes in order to fulfil its task of distinguishing synonyms.

3.1.2 Document source analysis

Concurrently to the user requirement analysis, the available data sources are analysed. The goal is identifying the different available XML tags, thus the elements

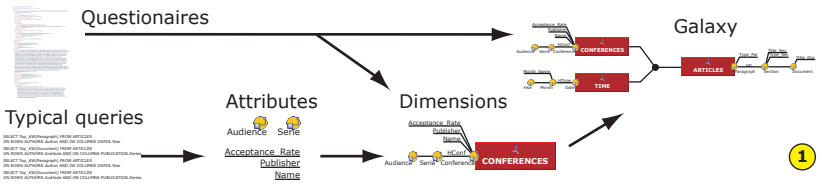

Fig. 4 Examples of typical queries.

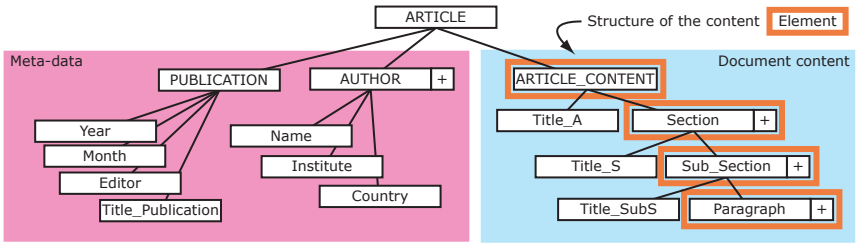

Fig. 5 Example of a document structure ("+" represents a cardinality of [1..*]).

of interest within the source XML documents (content, structure of the content and metadata) and updating the synonymy dictionary.

To do this, the documents structure is used as input: the DTD[5] , viewed as a tree. The DTD is either available within the data sources or it is built from well formed XML document sources. If only XSchema is available, it is converted into a DTD with some information loss (e.g. data types). From this tree view, with the use of the XML tags, the designer identifies: 1) the textual content of the documents (e.g. the textual data that constitutes the paragraphs of scientific papers); 2) the structure associated to this content (e.g. the elements that group paragraphs into subsections, themselves grouped into sections...); and 3) the metadata of the document (authors, date of publication...). The following rules are used to guide the analysis [32]:

- *Content* is held within elements usually far from the root of the document, i.e. leaves farthest from the root;
- The *structure of the content* is the hierarchy of elements allows one to reach the elements that hold the content data previously identified;
- Contrarily to content, *metadata* is held within elements that are close to the root.

This identification should be done in this order as once the elements that hold the content are identified; the associated structure may be easily deduced. As to metadata, it is in within some of the remaining elements. For example, in scientific articles, the DTD is extracted and displayed as a tree (see Fig. 5), notations are inspired by [4]. The content of the document (the textual data) may be found within the element farthest from the root: *Paragraph* (4 elements away from the root). The structure associated to this content is the hierarchy of elements below *ARTICLE_CONTENT*. It is composed of the elements: *ARTICLE_CONTENT*, *Section*, *Sub_Section* and *Paragraph*. The elements below *AUTHOR* and *PUBLICATION* (only 2 elements away from the root) represent the metadata associated to the document (here these elements correspond to information on the authors and the publication). The document source analysis phase is summarised in Fig. 6.

Tags from the elements of interest are inserted within the dictionary with synonyms that will ease understanding. For example, paragraphs of a document may be

[5] DTD : Document Type Definition, a grammar for expressing the structure of XML documents from http://www.w3.org/XML

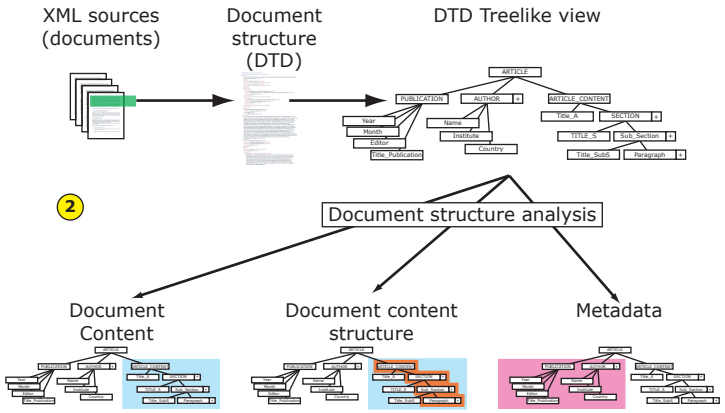

Fig. 6 Analysis of the data sources.

represented by a tag <*p*>, authors by <*au*>. As a consequence, these tags will be associated with the term *paragraph* for *p* and *author* for *au*.

The next step is to ensure the compatibility between the output of the user requirement analysis and the document source analysis phases.

3.2 Confrontation Phase

The goal of the confrontation phase is to ensure the compatibility between the outputs of the concurrent analysis phases: the galaxy schema (translation of the user requirements) and the document sources. Through a first step of comparison and association, the galaxy and sources are compared and incompatibilities detected. These later are dealt in afterwards. In the end, the user produces an association index that will allow data to be loaded within the multidimensional database structures.

3.2.1 Comparison and Association

The confrontation phase starts with a direct association process that links together elements of the data sources to elements (attributes) of the galaxy schema, see Fig. 7. This phase uses as input the galaxy schema previously produced and a tree-like view of the DTD structure of the XML data sources.

This is done by associating an element of the data sources' DTD to an attribute of the galaxy schema. To ease this process, the galaxy schema is converted into a DTD representation. In this representation, each clique is converted into an XML tree of elements and each element represents an attribute of the galaxy (due to lack of space, this conversion is not detailed here). Both structures are compared and attributes of the galaxy are associated with elements from the data sources. The association is

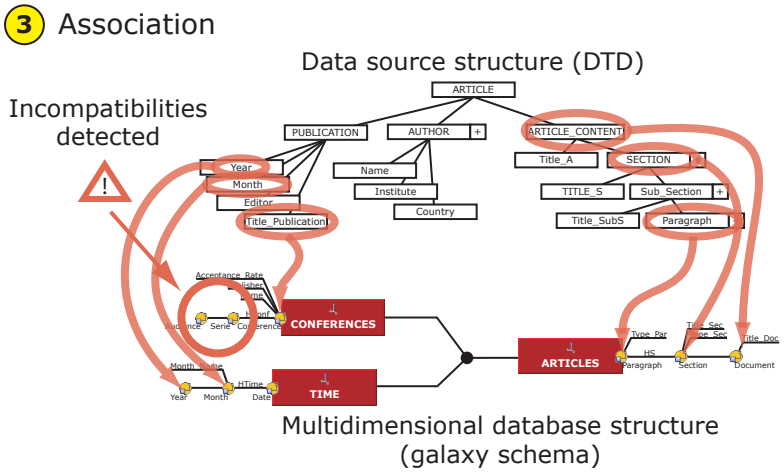

Fig. 7 Comparison: association (for simplicity weak attributes are not linked).

done in a semi-automatic way as associations are suggested with the help of the synonym dictionary. The user is requested to intervene if synonymy problem occur or if elements of the DTD do not correspond to attributes of the galaxy.

For each association between an element of the source DTD and an attribute of the galaxy schema, an index is updated with a line that summarises the association, i.e. an XPath [39] expression expressed from the root element to the linked element from the source and the corresponding attribute:

[*XPath expression to the element*]/[*Element*] : [*Dimension name*].[*Attribute name*]

For example, the elements *Year* (respectively *Month*) below the element *PUBLICATION* of the DTD will be associated to the attributes *Year* (resp. *Month*) of the galaxy. The corresponding association in the index is an XPath expression for the element from the source associated to the corresponding attribute in the galaxy:

/ARTICLE/PUBLICATION/Year : *TIME.Year*

The association process continues until all possible associations between the source elements and the attributes of the galaxy are done.

3.2.2 Dealing with Incompatibilities

Incompatibilities arise when an attribute of the galaxy may not be associated to an element in DTD of the data source. Note that the data sources have been through ETL processing and are already cleansed with no errors. As a consequence, the handled incompatibilities are here are not very complex. We consider only two incompatibility types in our environment (although other more detailed types exist [37]):

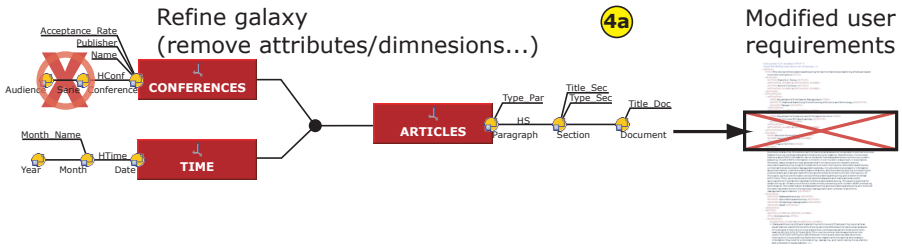

Fig. 8 Adapting the galaxy.

- *No data*: the data within the source does not exist or the data is only partially available (e.g. authors have their affiliations but not their);
- *Incompatible data*: the data exists in the source but it is incompatible. Incompatible cardinality, buried within other data, incompatible format. . . (e.g. first and last names of authors are together whereas only the last name would be required);

The last issue is less critical than the first as usually a conversion will help a lot. In our environment, as long as the conversion may be specified with an XQuery expression, the incompatible data issue may be solved. We thus concentrate on the more critical issue: lack of data. There are two possible solutions to this issue: to handle this incompatibility either within the galaxy or within the data sources. The first approach consists in adapting the galaxy in order to make it compatible with the data sources consists in removing incompatible elements, thus changing the analysis objectives (see Fig. 8).

The second approach consists in modifying the sources to render them compatible or, if possible, adding the missing data to the documents using auxiliary sources such as additional documents or domain documents. In order to do this, the new data must be compatible and there is a necessity for having a way of linking both data sources together (a compatible key or at least semantic join without any synonymy problems). The steps of the process are summarized in Fig. 9.

Two technical solutions may be considered for processing the data sources enrichment. If modifying the data sources is conceivable, the new data is added to the documents within the data sources. Otherwise, the new data is added separately and a linking index is used between the source data and the newly added data (using Ids for the links between two XPath expressions–one for the source data, one for the corresponding linked data). In both cases a unified DTD view is presented to the user, so that he will manipulate a unique DTD and not several.

For example, if the data sources do not have detailed information about conferences, such as the audience, the series, the publisher (. . .). It would be possible to use complementary information such as those from the DBLP XML database [9] and use this data as complementary sources.

The process is iterated until no more incompatibilities arise. The designer may then proceed to the loading phase.

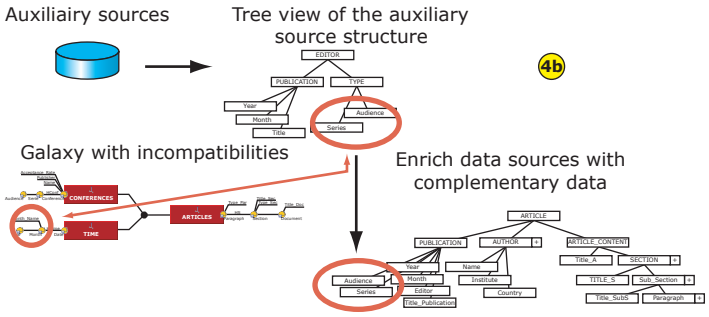

Fig. 9 Adding missing data to the XML documents sources.

3.3 Implementation: Creation of Multidimensional Structures and Loading

The objective of this phase is to create the multidimensional database that will be used by decision makers. First, the specification of the galaxy schema is used to generate the structures of the multidimensional database. Second, the association index generated at the end of a successful confrontation phase is used to load data extracted from sources into the empty structures of the multidimensional database. The phase is fully automatic and will be detailed in the section that describes our tool. The implementation phase is summarised in Fig. 10.

Once data is loaded within the MDB, the user may start analyses. Our design process allows a realistic implementation of the multidimensional schema, i.e. an OLAP systems that complies not only with the user requirements (the analysis requirements) but also with the XML document sources. In order to validate this approach we have implemented a data integration tool.

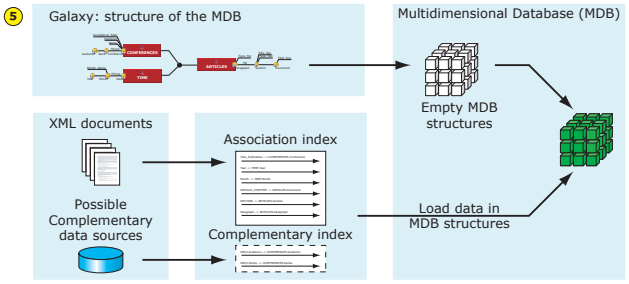

Fig. 10 Step 5: Loading source data within the multidimensional database.

4 Data Integration Tool

The objective of our tool is to help the user in the process of designing and implementing an OLAP system by automating some of the steps of the design process:

- The design of the galaxy schema (within step 1);
- The association of elements from the source document DTD to the attributes of the galaxy schema (within step 3);
- The modification / refining of the MDB (step 4a);
- The automatic generation of the structures of the MDB (within step 5);
- The loading of data extracted from the document sources within the structures of the multidimensional database (within step 5).

As a consequence, the tool focuses on assisting the user during certain steps 1, 3 (association only), 4a and 5 (see Fig. 2). The next part describes the architecture of the tool followed by the description of its interfaces.

4.1 Tool Architecture

The tool is used to create the structures of the multidimensional database through a galaxy schema and then loading data within these structures. The architecture of the tool rests on six components: two user interfaces each with an associated process, and two storage space (a document warehouse and a multidimensional database). The two interfaces will be described later (see section 4.2).

The *document warehouse* and the multidimensional database are stored within a ROLAP architecture in an Oracle 10*g*2 DBMS. The system uses part of Oracle XMLDB to handle the XML data within the ROLAP environment. The document warehouse is used to render the source XML documents available.

The *multidimensional database* (MDB) is composed of a meta-base and a database. The meta-base holds the description of the galaxy structures and the database holds these structures (relational tables) and their content data. Dimensions are denormalised tables and cliques are modelled by foreign keys. The multidimensional structures and the associated descriptive data (meta-base) are generated by the user with the use of a galaxy editor (see left part of Fig. 11).

Data is loaded within these structures from the document warehouse with the use a data integrator that allows the generation of the association index by the user. This index is used within scripts that will extract data from the documents to the empty relational tables that represent the structures of the galaxy (see right part of Fig. 11).

Two interfaces, described hereinafter, assist the designer in the design and implementation process.

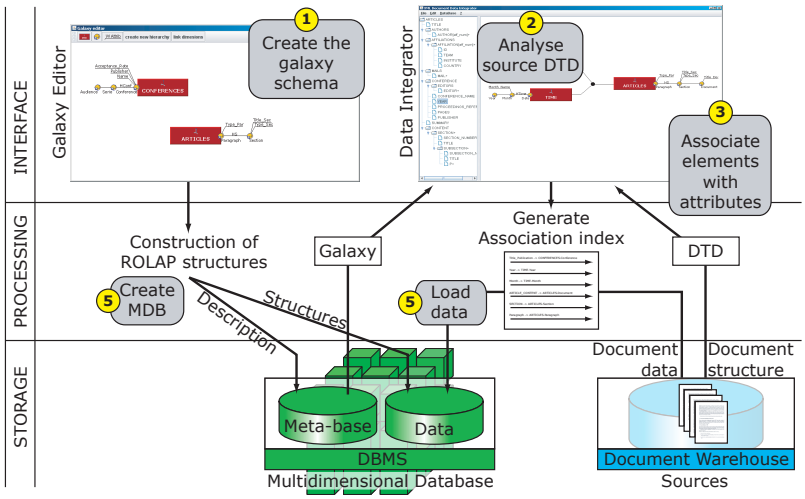

Fig. 11 Architecture of the tool (numbers correspond to the steps in section 3).

4.2 Tool Interfaces

The tool is manipulated with two interfaces: the first is a graphic editor that allows the designer to create a galaxy schema and the second is used to express associations between DTD elements of the sources and galaxy attributes (see Fig. 12).

The galaxy editor is a graphic editor for creating a galaxy schema. The user selects the elements he wishes to create (dimension, parameter or weak attribute) and associates them together in order to first build hierarchically organised dimensions. The user then links dimensions together in order to create the galaxy. Once a galaxy schema has been specified, the corresponding multidimensional structures and descriptive metadata may be automatically created within the multidimensional database (see Fig. 12). Each dimension of the galaxy schema is translated into a denormalised relation as in a star schema [18] and associated to some descriptive data (stored in the meta-base). Note that this generation should only be done once all incompatibilities between the galaxy and the sources have been solved. These are solved by comparing the galaxy and the DTD of the source with an available graphic viewer (XML Spy, or XML Doctor)[6] , or directly with the use of the second interface. In case of incompatibilities, the user either enriches the data sources with XQuery scripts and/or modifies the galaxy schema using the galaxy editor interface.

When the source schema and the galaxy schema have no more incompatibilities, the XML *data integrator* interface is used. With this interface a designer associates elements from the DTD of the XML documents with attributes of the galaxy. (see Fig. 12). Note that for each attribute of the galaxy, the tool suggest elements whose

[6] ALTOVA XMLSpy from http://www.altova.com/products/xmlspy/xml_editor.html and XML Doctor from http://sourceforge.net/projects/xmldoctor

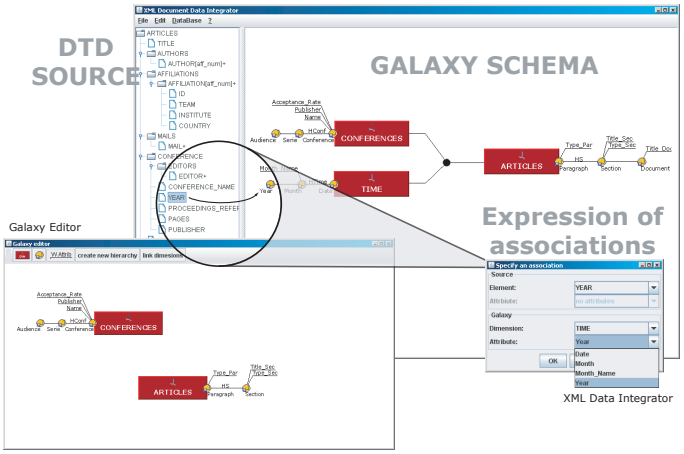

Fig. 12 Screenshots of galaxy editor (left) and the XML data integrator (right).

semantics seem close to the attribute (this is done with the dictionary). Associations specified are translated into links that will be used in order to extract data from XML documents and load it within the empty structures of the multidimensional database (see Fig. 11). We recall that the index is composed of XPath expressions associated to attribute names from the galaxy. A script combining SQL and XQuery queries is used to populate the dimension data. The XQuery part uses the XPath expressions to get the source data and returns the data that is then inserted, with the use of SQL, into the different relations of the multidimensional database (data types must be compatible).

We are currently using XML documents from the INEX IEEE XML document collection [15]. This collection of documents is composed of journal articles from the IEEE published between 1995 and 2002, transcribed into XML and represents over 12,000 documents.

5 Conclusion and Future Works

This paper aims at allowing OLAP analyses of documents. In order to do so, we have proposed a design and implementation method of OLAP schemas from text-rich (document-centric) XML documents. We have proposed a multidimensional model that has the advantage of resting on a unique concept (a galaxy) and that does not restrain the decision maker with limited predefined analyses with only numeric indicators [31]. The dimensions, which are multi hierarchical, allow the representation of document structures, content and metadata as well as their links. Associated to this model we have defined a design process that rests on a mixed approach, allowing the design and the implementation of a galaxy. The approach analyses jointly decision makers' requirements and the data sources (documents) in order to

provide valid OLAP schemas. A confrontation phase ensures the compatibility between the multidimensional schema and the available data sources. The schema is possibly refined through an iterative process. In order to validate our proposition, we have implemented a CASE tool allowing the design and implementation of galaxy schemas. The tool rests on graphic interfaces that provide an easy representation of the multidimensional schema and its associated loading process.

We consider three future directions. So far we have used a collection of XML documents that is uniform regarding a single DTD. We consider taking into account more heterogeneous sources such as XML documents with different structures (DTD or XSchema). As a solution, one could employ schema relaxation [1] as well as dictionaries for finding synonymy within the different xml elements. Throughout the paper, we based our approach on an existing XML document collection: the INEX IEEE XML document collection [15]. Had such a collection been unavailable, more complex ETL processes [37] would have been taken into consideration in order to build a uniform collection of XML documents. Another important direction would be to help the designer with semi-automatic generation of optimisations that are required by the slow processing of textual data. These optimisations, (e.g. materialised views [13]), would be generated as the galaxy is created and the data loaded.

References

1. Amer-Yahia, S., Cho, S., Srivastava, D.: Tree pattern relaxation. In: EDBT '02: Proceedings of the 8th International Conference on Extending Database Technology, pp. 496–513. Springer-Verlag, London, UK (2002)
2. Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., Paraboschi, S.: Designing data marts for data warehouses. ACM Trans. Softw. Eng. Methodol. **10**(4), 452–483 (2001)
3. Boussaïd, O., BenMessaoud, R., Choquet, R., Anthoard, S.: X-warehousing: an xml-based approach for warehousing complex data. In: 10th East-European Conference on Advances in Databases and Information Systems (ADBIS 06), Thessaloniki, Greece, *LNCS*, vol. 4152, pp. 39–54. Springer, Heidelberg, Germany (2006)
4. Braga, D., Campi, A., Ceri, S.: XQBE (XQuery By Example): A visual interface to the standard XML query language. ACM Trans. Database Syst. **30**(2), 398–443 (2005)
5. Cabibbo, L., Torlone, R.: A logical approach to multidimensional databases. In: EDBT '98: Proceedings of the 6th International Conference on Extending Database Technology, pp. 183–197. Springer-Verlag, London, UK (1998)
6. Carneiro, L., Brayner, A.: X-META: A methodology for data warehouse design with metadata management. In: Lakshmanan [19], pp. 13–22
7. Cavero, J.M., Piattini, M., Marcos, E.: MIDEA: A multidimensional data warehouse methodology. In: ICEIS (1), pp. 138–144 (2001)
8. Colliat, G.: OLAP, relational, and multidimensional database systems. SIGMOD Rec. **25**(3), 64–69 (1996)
9. DBLP: Computer science bibliography. XML available from: http://dblp.uni-trier.de/xml
10. Fuhr, N., Grosjohann, K.: XIRQL: A query language for information retrieval in XML documents. In: Research and Development in Information Retrieval, pp. 172–180 (2001)
11. Golfarelli, M., Rizzi, S.: Methodological framework for data warehouse design. In: DOLAP'98, ACM First International Workshop on Data Warehousing and OLAP, pp. 3–9 (1998)

12. Gyssens, M., Lakshmanan, L.V.S.: A foundation for multi-dimensional databases. In: VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 106–115. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997)
13. Harinarayan, V., Rajaraman, A., Ullman, J.D.: Implementing data cubes efficiently. SIGMOD Rec. **25**(2), 205–216 (1996)
14. Hüsemann, B., Lechtenbörger, J., Vossen, G.: Conceptual data warehouse modeling. In: M.A. Jeusfeld, H. Shu, M. Staudt, G. Vossen (eds.) DMDW, *CEUR Workshop Proceedings*, vol. 28, p. 6. CEUR-WS.org (2000)
15. INEX: INitiative for the Evaluation of XML retrieval (INEX). XML document collection used until 2005, from http://inex.is.informatik.uni-duisburg.de/
16. Jensen, M.R., Møller, T.H., Pedersen, T.B.: Specifying OLAP cubes on XML data. J. Intell. Inf. Syst. **17**(2-3), 255–280 (2001)
17. Keith, S., Kaser, O., Lemire, D.: Analyzing large collections of electronic text using OLAP. In: APICS 2005, 29th Conf. in Mathematics, Statistics and Computer Science (2005)
18. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. John Wiley & Sons, Inc., New York, NY, USA (2002)
19. Lakshmanan, L.V.S. (ed.): Design and Management of Data Warehouses 2002, Proceedings of the 4th Intl. Workshop DMDW'2002, Toronto, Canada, May 27, 2002, *CEUR Workshop Proceedings*, vol. 58. CEUR-WS.org (2002)
20. Luján-Mora, S., Trujillo, J.: A comprehensive method for data warehouse design. In: 5th Intl. Workshop on Design and Management of Data Warehouses (DMDW'03), vol. 77. CEUR Workshop Proceedings, CEUR-WS.org (2003)
21. Mazón, J.N., Trujillo, J.: A model driven modernization approach for automatically deriving multidimensional models in data warehouses. In: Parent et al. [25], pp. 56–71
22. McCabe, M.C., Lee, J., Chowdhury, A., Grossman, D., Frieder, O.: On the design and evaluation of a multi-dimensional approach to information retrieval (poster session). In: SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 363–365. ACM, New York, NY, USA (2000)
23. Mothe, J., Chrisment, C., Dousset, B., Alaux, J.: DocCube: multi-dimensional visualisation and exploration of large document sets. J. Am. Soc. Inf. Sci. Technol. **54**(7), 650–659 (2003)
24. Nassis, V., Rajugan, R., Dillon, T.S., Rahayu, J.W.: Conceptual design of XML document warehouses. In: Y. Kambayashi, M.K. Mohania, W. Wöß (eds.) DaWaK, *Lecture Notes in Computer Science*, vol. 3181, pp. 1–14. Springer (2004)
25. Parent, C., Schewe, K.D., Storey, V.C., Thalheim, B. (eds.): Conceptual Modeling - ER 2007, 26th International Conference on Conceptual Modeling, Auckland, New Zealand, November 5-9, 2007, Proceedings, *Lecture Notes in Computer Science*, vol. 4801. Springer (2007)
26. Park, B.K., Han, H., Song, I.Y.: XML-OLAP: A multidimensional analysis framework for XML warehouses. In: A.M. Tjoa, J. Trujillo (eds.) DaWaK, *Lecture Notes in Computer Science*, vol. 3589, pp. 32–42. Springer (2005)
27. Pérez-Martínez, J.M., Berlanga-Llavori, R., Aramburu-Cabo, M.J., Pedersen, T.B.: Contextualizing data warehouses with documents. Decis. Support Syst. **45**(1), 77–94 (2008)
28. Phipps, C., Davis, K.C.: Automating data warehouse conceptual schema design and evaluation. In: Lakshmanan [19], pp. 23–32
29. Prat, N., Akoka, J., Comyn-Wattiau, I.: A UML-based data warehouse design method. Decis. Support Syst. **42**(3), 1449–1473 (2006)
30. Ravat, F., Teste, O., Tournier, R.: OLAP aggregation function for textual data warehouse. In: J. Cardoso, J. Cordeiro, J. Filipe (eds.) ICEIS (1), pp. 151–156 (2007)
31. Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: A conceptual model for multidimensional analysis of documents. In: Parent et al. [25], pp. 550–565
32. Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: Integrating complex data into a data warehouse. In: SEKE, pp. 483–. Knowledge Systems Institute Graduate School (2007)
33. Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: Algebraic and graphic languages for OLAP manipulations. International Journal of Data Warehousing and Mining **4**(1), 17–46 (2008)

34. Rizzi, S., Abelló, A., Lechtenbörger, J., Trujillo, J.: Research in data warehouse modeling and design: dead or alive? In: DOLAP '06: Proceedings of the 9th ACM international workshop on Data warehousing and OLAP, pp. 3–10. ACM, New York, NY, USA (2006)
35. Sullivan, D.: Document Warehousing and Text Mining: Techniques for Improving Business Operations, Marketing, and Sales. John Wiley & Sons, Inc., New York, NY, USA (2001)
36. Tseng, F.S.C., Chou, A.Y.H.: The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence. Decis. Support Syst. **42**(2), 727–744 (2006)
37. Vassiliadis, P., Simitsis, A., Georgantas, P., Terrovitis, M., Skiadopoulos, S.: A generic and customizable framework for the design of etl scenarios. Inf. Syst. **30**(7), 492–525 (2005)
38. Vrdoljak, B., Banek, M., Skocir, Z.: Integrating XML sources into a data warehouse. In: J. Lee, J. Shim, S. goo Lee, C. Bussler, S.S.Y. Shim (eds.) DEECS, *Lecture Notes in Computer Science*, vol. 4055, pp. 133–142. Springer (2006)
39. W3C: eXtensible Markup Language (XML) 1.0. W3C Recommendation (29/09/2006), from http://www.w3.org/TR/2006/REC-xml-20060816
40. Yin, X., Pedersen, T.B.: Evaluating XML-extended OLAP queries based on a physical algebra. In: DOLAP '04: Proceedings of the 7th ACM international workshop on Data warehousing and OLAP, pp. 73–82. ACM, New York, NY, USA (2004)