A Novel Multidimensional Model for the OLAP on Documents: Modeling, Generation and Implementation

Maha Azabou[1], Kaïs Khrouf[1], Jamel Feki[1], Chantal Soulé-Dupuy[2], and Nathalie Vallès[2]

[1] University of Sfax, Faculty of Economics and Management, Computer Department,
MIR@CL Laboratory, University of Sfax,
Airport Road Km 4, P.O. Box. 1088, 3018 Sfax, Tunisia
Azabou.Maha@yahoo.fr, Khrouf.Kais@isecs.rnu.tn,
Jamel.Feki@fsegs.rnu.tn

[2] IRIT, University of Toulouse 1 Capitole,
2, Rue du Doyen Gabriel Marty, 31042 Toulouse Cedex 9, France
{Chantal.Soule-Dupuy, Nathalie.Valles-Parlangeau}@ut-capitole.fr

Abstract. As the amount of textual information grows explosively in various kinds of business systems, it becomes more and more essential to analyze both structured data and unstructured textual data simultaneously. However information contained in non structured data (documents and so on) is only partially used in business intelligence (BI). Indeed On-Line Analytical Processing (OLAP) cubes which are the main support of BI analysis in decision support systems have focused on structured data. This is the reason why OLAP is being extended to unstructured textual data. In this paper we introduce the innovative "Diamond" multidimensional model that will serve as a basis for semantic OLAP on XML documents and then we describe the meta modeling, generation and implementation of a the *Diamond* multidimensional model.

Keywords: XML Documents, OLAP, *Diamond* multidimensional model.

1 Introduction

With the boom of Internet and the ever increasing business intelligence applications, search and analysis of textual data have attracted broad attention. Several studies have proposed the reuse of On-line Analytical Processing (OLAP) technology on documentary information in order to analyze a set of textual data with their underlying semantic information. However, these studies have shown that most of standard multidimensional models (such as star and snowflake models) have not been designed to manage complex textual data (multi-structural and multi sources data requiring the resolution of semantic ambiguities). In order to overcome this lack and to satisfy the need of dealing with the semantic content of documents, we have introduced a novel multidimensional model dedicated to the OLAP on documents which we called "*diamond* model". This model is able to represent the semantic content of a collection of documents whose structures are homogeneous (same or similar XML schemas or logical structure for example) but heterogeneous in semantics (semantic structure of each document). This innovative model will guide the design of data marts dedicated to data analysis on collections of documents according to their content and structure. In order to generate automatically this *diamond* multidimensional model, we propose a set of heuristic rules aiming at determining the various components of the model (i.e., fact and dimensions).

This paper is structured as follows. Section 2 presents the related work dealing with the multidimensional modeling of documents. Then, we present in Section 3 the conceptual modeling of *diamond* models. Section 4 describes the Meta model to keep this model in the same referential. Section 5 details the rules and process we propose to generate *diamond* multidimensional models. Section 7 concludes this paper by a review of our proposals and some future work.

2 Related Work

The conventional multidimensional modeling has been proved robust in the case of highly-structured digital data, but inappropriate for the OLAP on documents. In fact, several works have dealt with this issue in order to provide star or snowflake schemas, integrating a content description of textual fragments by term hierarchies, documents hierarchies, term clusters, AP-Dimension (AP stands for A Priori), and contextual dimension.

- Term hierarchies [9] [12] and [11]

[9] proposed a new data cube called Text cube based on the star schema in which a textual dimension is represented by terms hierarchy. This hierarchy specifies the semantic relationships between textual terms extracted from documents, which allows semantic navigation in textual data thanks to two associated operators: pull-up (which generates a term level L_0 from a lower term level L) and push-down (which generates a term level L_0 from a higher term level L). These two operators can define or change a term level, hence helping users in the multi-semantic-level analysis of textual data.

The Topic Cube proposed by [12] is a good example of Text OLAP using a topic dimension based on the star schema which extends the traditional data cube by integrating a hierarchy of topics as an analytical dimension. It is a new cube model using a topic dimension and a text content measure which carries parameters of a probabilistic model that serves to indicate how well the text content matches with the topic. Two kinds of measures are stored in a topic cube cell: word distribution of a topic p(w_i|*topic*) and topic coverage by documents p(*topic*|d_j). The topic coverage means the probability that document d_j covers the topic.

Based on Topic Cube and information network analysis, [11] proposed iNextCube (Information Network-Enhanced Text Cube) which integrated the capability of automatic formation of topic hierarchy through information network analysis. It is tedious and error-prone to rely on human experts to specify topic hierarchy, so it is desirable to automatically construct topic hierarchy by information network analysis.

- Term clusters

[13] have proposed a new model called MicroTextClusterCube. The main idea is to speed up on-line analysis of textual cells by doing as much preprocessing as possible. Specifically, they preprocess the documents so that to generate a good number of micro-clusters to compress similar documents. They have suggested to introduce a new analytical measure ($mean_i$, $size_i$) representing respectively the vector of weighted terms and the size of a micro-cluster.

- Document hierarchy

[7] proposed a new model called Cube Index based on a hierarchical description of each document. This hierarchy specifies relationships between words with respect to one document. It is used for the analysis of words in various levels of abstraction (L_i) in a document such as Document (L_5), Paragraph (L_4), Sentence (L_3), Word Pair (L_2) and Word (L_1). Two new operations scroll up (given a level L_i and term v belonging to L_i, the result is a higher level L_{i+1} of document hierarchy) and scroll down (inverse operator of scroll up.) are discussed exclusively for the cube index. It supports term frequency and inverted document frequency to facilitate information retrieval techniques.

- An AP-Dimension

In [3], the authors have proposed a multidimensional model that supports textual information by introducing a textual dimension (AP-dimension) obtained from a semantic structure called AP-Structure. This one is based on the frequent items named AP-sets (a priori sets) obtained by applying the a priori algorithm (a basic algorithm for frequent item set mining and Boolean association rule learning [2]) on textual attributes of a transactional database. This structure represents the hidden meaning behind the text instead of a simple word play.

- A contextual dimension

In [10], the authors have proposed a contextual text cube model called CXT-Cube, associated with contextual dimensions. The fact table includes a new measure for textual data analysis, based on an adapted vector space model to represent textual data. In order to calculate the document terms' weights, [10] have proposed a relevance propagation technique on a concept hierarchy. Also, the authors have provided a new aggregation operator denoted ORank, to aggregate the documents during the analysis process.

More recently, considerable works was directed towards online analytical processing on informational networks and mostly focusing on the Semantic Web data.

- Open Cubes

[5] introduced Open Cubes which focus on the publication of multidimensional cubes on the Semantic Web and they found the limitation of the RDF Data Cube (DC) which can only address statistical data. Their work revolves around informational OLAP aggregations. Furthermore they revisit RDF Data Cube by extending DC's capabilities to support multidimensional levels to build hierarchies and to implement other OLAP operators beside the sole Slice operator offered by DC.

- Tag cloud

In the context of Web 2.0 and OLAP applications, [1] was particularly interested in tag clouds. The principle of a tag cloud provides a visual representation of keywords on a Web site. Generally, words appear in a font size that is greater than that of the words used on the site. The keyword cloud can be viewed as a semantic abstract of the web site. Applied to OLAP, this principle can use a tag cloud to represent a cube where each keyword is a cell and where keyword size depends on the measured value of the fact (cell).This new cube representation makes it possible to use specific tag cloud operations such as sorting keywords according to their size and removing keywords with small sizes. To assist the user in his exploration of the cube, [1] built classes of similar keywords and rearranged the attributes of the dimensions.

These works have been able to manage the textual data semantics. However, after a large analysis of related work, we concluded that the structural aspect has not been taken into account in multidimensional modeling of document collections. As a result, it is impossible to achieve analyses on various structural levels.

Hence, in order to satisfy the need to represent the textual data semantics and to organize it hierarchically, we introduce a new multidimensional model dedicated to the OLAP on documents, called "diamond", allowing us to manage both document structures and the semantics of the textual content. This model includes standard dimensions with factual data (such as date, author, publisher), as well as a dimension for the semantics of the textual fragment contents (such as summary, content, paragraph). Given a set of XML text-centric documents, the main objective of this paper is the automatic generation of a *diamond* Model in order to facilitate the task of the designer. To do so, we suggest a set of heuristic rules which determine the various components of the model.

3 Diamond Multidimensional Model: Modeling

The *diamond* multidimensional model we propose is dedicated to the OLAP on documents. This new *diamond* Model consists of:

- A *fact* which corresponds to a given observation on textual documents (any kind of observation);

- A set of *dimensions* distributed as follows:

- *Standard* dimensions are the axes of analysis; these axes are constituted by the elements of the first level of the documents' structure. For each element, its descendants represent the parameters (organized into hierarchies) or the weak attributes (attached to a parameter to complement it). For example in Figure 1, Id-Movie parameter of D-Movie standard dimension completed by two weak attributes, Title and Summary;
- *Version* dimension of documents concerns the various versions of documents, as well as associated metadata, such as *Author, Date of creation,* and *Description*;
- *Semantic* dimension which is a central dimension; it consists of the following hierarchy: *Concept* → *Semantic resource*. *Concept* parameter is connected to textual elements (like *Section*, *Paragraph*) of documents in order to clarify the semantics of the element in an OLAP analysis. A *semantic resource* may be ontology, taxonomy, thesaurus or any other kind of resource which can be filled and validated owing to semantic web tools and sources[1].

As an example, Figure 1 presents the *Diamond* Model corresponding to the multidimensional modeling of a collection of film Festival descriptions (derived from the description of each film festival structure shown in Figure 2).

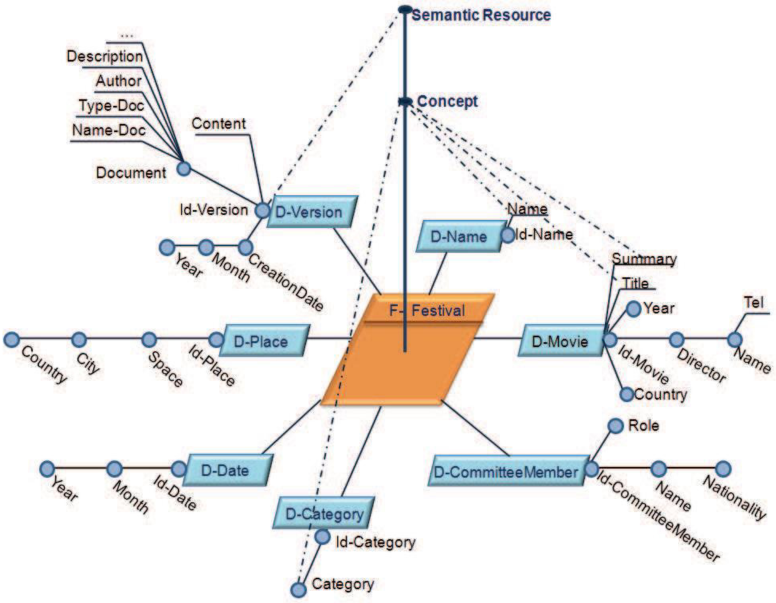

Fig. 1. Diamond multidimensional model for the Festival collection

[1] The study and the generation of a semantic dimension has been the subject of previous work.

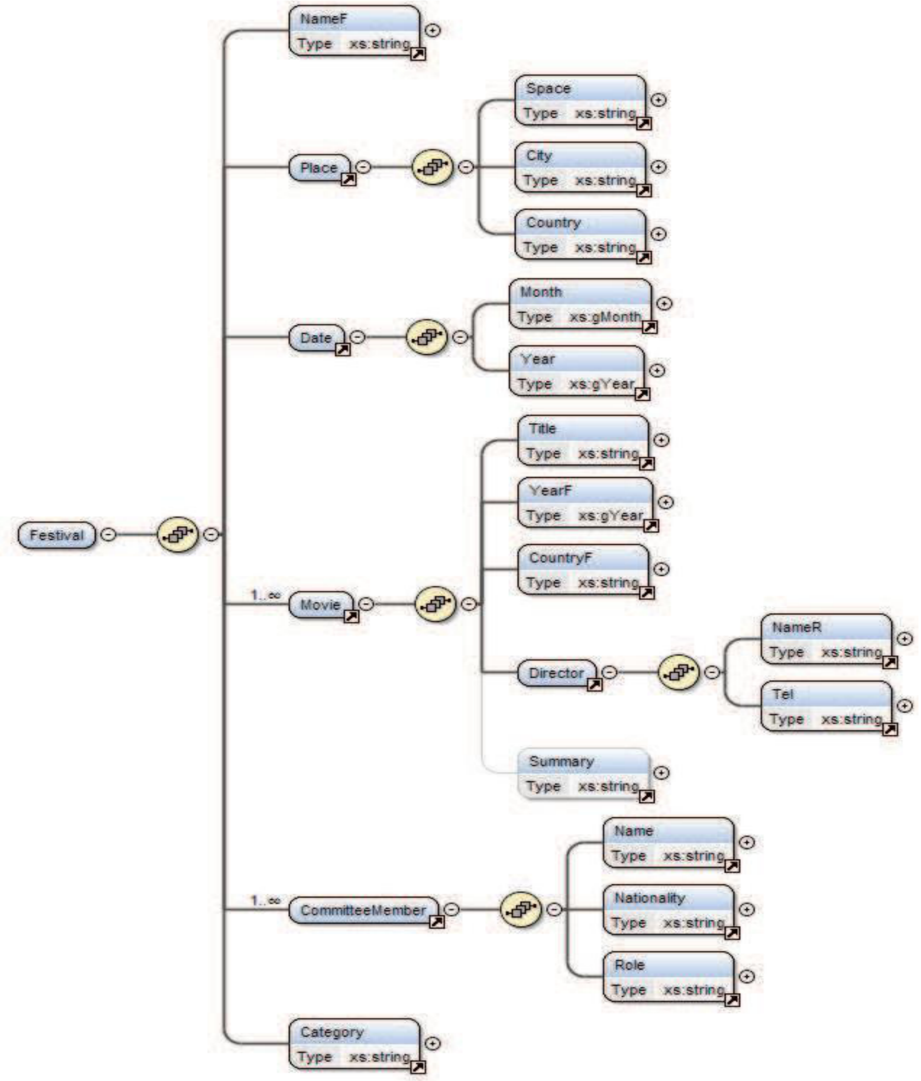

Fig. 2. Festival XSchema (generated with Oxygen tool)

4 Diamond Multidimensional Model: Meta-modeling

In our work, we propose the *Diamond* model to represent any data mart dedicated to a collection of documents.

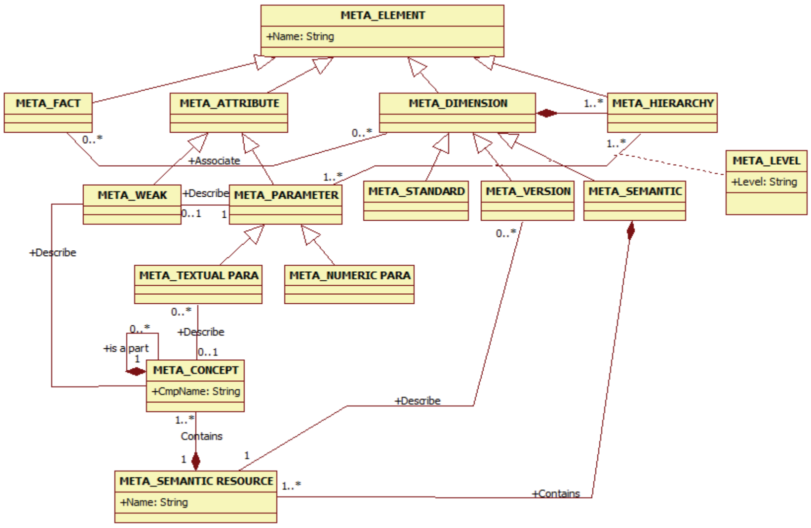

Fig. 3. The Meta model of Diamond models

In our case a collection of documents gathers documents relying to a same topic and having similar structures. To keep together these different models in the same referential, we propose the meta-model depicted in Figure 3.

This Meta model describes a set of elements and their characteristics. These elements are either facts, attributes, dimensions or hierarchies. We define three dimensions (Standard, Version and Semantic) as a specialization of dimension. Each dimension has one or many hierarchies. Each hierarchy is described by a set of parameters. Some parameters can be enriched with weak attributes that give more meaning to a parameter. We distinguish two subclasses of parameters: textual and numeric parameters. This distinction is needed and facilitates the use of textual parameters in semantic analysis. A semantic dimension is described by a set of semantic resources that contain a set of concepts affected to textual parameters and weak attributes.

In the following section, we describe the rules and the process we propose to generate a *diamond* multidimensional model from a collection of XML documents conform to a logical structure (i.e., Document Type Definition DTD or XSchema).

5 Diamond Multidimensional Model: Generation

Our objective is to generate the *diamond* multidimensional model in quasi-automatically, i.e., a way where the user's intervention is limited to the verification, rectification and approval of the generated model. Moreover, the model must include all elements of the DTD or XSchema of the considered collection of documents.

5.1 Identification of the Fact and the Dimensions

The heuristic identification rules of fact and dimensions are described in this section as follows:

Rule 1: The root *A* of a logical structure (DTD or XSchema) containing at least one leaf becomes a fact named *F-A* [6].

Rule 2: Each node *d* immediate descendant of the root node *A* of the logical structure becomes a dimension named *D-d*.

Rule 3: At each dimension *D* will be affected an artificial identifier (surrogate key) named *Id-D*.

Example
Let us consider the *Festival XSchema* (cf. Figure 2).
According to *Rule 1,* the fact is *F-Festival*.
Rules 2 and *3*: The first descendants become the following dimensions; *D-Name*, *D-Place*, *D-Date*, *D-Category*, *D-CommitteeMember*, *D-Movie*, with the corresponding identifiers *Id-Name*, *Id-Place*, *Id-Date*, *Id-Category*, *Id-CommitteeMember*, *Id-Movie*.

Rule 4: For each node transformed into a dimension *D* that has no descendant, two cases may appear:

- If the *D* dimension contains distinct values[2] then the name of this dimension will be a weak attribute, directly connected to the identifier of *D* (*Id-D*).
- Otherwise, we add a parameter of rank 2 whose name is the one of the *D* dimension.

The intuition behind this rule is that an element having distinct values cannot be used as an aggregation criterion and, therefore could not be elected as a parameter.

Example: Let us take a part the *Festival XSchema* (cf. Figure 2). According to *Rule 4,* the *Name* element contains distinct values (two different festivals cannot have the same name); therefore the *D-Name* dimension will have *Name* as weak attribute directly connected to the *Id-Name* identifier. Conversely, the *Category* element does not contain distinct values (two different festivals may belong to the same category – long film, short film or horror film for example); then we add to the *D-Category* dimension a parameter of rank 2 named *Category*.

Rule 5: For each dimension *D* having descendants, the first descendant *d* is processed as follows*:*

- If *d* contains distinct values then *d* will be a weak attribute, directly connected to the identifier of *D* (*Id-D*) because we have already assigned an identifier to the *D* dimension owing to *Rule 3.*
- Otherwise, we add a parameter of rank 2 having for name *d.*

[2] Two different instances of this element cannot contain the same value.

The objective of this rule is to check if the first descendant d of a dimension D constitutes an identifier (containing only unique values).

Note: We consider that the first descendant element of the dimension *D* is the most appropriate one to become the identifier. However, the user can intervene to change this choice depending on his research interest.

Example: Let us take a part of the *XSchema* of Figure 2. We notice that the *Space* element in the *D-Place* dimension does not have distinct values (several different festivals may be located at the same place); hence *Space* will be a parameter of rank 2 of the *D-Place* dimension. The same logic applies for *Name* of the sub-tree *CommitteeMember*. Conversely, the *Title* element of the *Movi*e sub-tree contains distinct values (we consider here that a film cannot be presented in different festivals); it will represent a weak attribute for *Id-Movie* identifier of the *D-Movie* dimension.

5.2 Identification of the Hierarchies

In order to identify hierarchies, we have to determine the Functional Dependencies (*FD*) between the elements of the document structure. In database, a *FD* from the attribute *A* to the attribute *B*, noted *A*→*B*, expresses that each value of *A* is associated to one, and only one, value of *B*. This choice can be explained by the fact that the element *A* can play the role of parameter for a specific level and the element *B* can be a parameter of a more generic level. (Example: City→ Country).

- Hierarchical organization of parameters

Let *A*, *B* and *C*, be the immediate descendants of a node dimension *D*. Our objective now is to extract the parameters of rank greater than 1.

Rule 6: Determination of FD
If there is a non-symmetrical FD *A*→*B* (i.e., without *B*→*A*) then *A* and *B* constitute two consecutive parameters for *D*, i.e., of rank *i* and *i +1* respectively.

Generally, the multidimensional acyclic constraint checks that the graph of parameters is a directed acyclic graph (DAG), i.e., with no cycles.

Rule 7: Avoiding cycles
Eliminate transitive FDs in order to respect the multidimensional cyclic constraint. Let us have the following three FDs: *A*→*B, B*→*C* and *A*→*C*, then the FD *A*→*C* is transitive and has to be eliminated. Then, we get the following hierarchy: *A*→*B*→*C*.

Rule 8: Determination of i th hierarchy level (i>1)
If there is no FD between *A* and all descendants of the *D* dimension-node, then *A* constitutes a parameter of rank 2 for the *D* dimension (i.e., *A* is directly connected to the identifier *Id-D* of *D*).

Example: Let us take the *Place* sub-tree in the *Festival XSchema* (cf. Figure 2).

Rule 6 determines the existence of three non-symmetrical FDs *Space*→*City*; *Space*→*Country* and *City*→*Country*. Applying *Rule 7* allows us to determine the following hierarchy: *Space*→*City*→*Country* (cf. Figure 4.a).

Now, let us continue with the *CommitteeMember* sub-tree of the *Festival XSchema* (cf. Figure 2).

Rule 6 proves the existence of a non-symmetrical FD: *Name*→*Nationality*.

Rule 8: No FD between *Role* and all descendants of *CommitteeMember* (*Name* and *Nationality*) because a member of a committee can be simultaneously president in a given festival and simple member in another festival, Role is then a parameter of rank 2 connected to *Id-CommitteeMember* (cf. Figure 4.b).

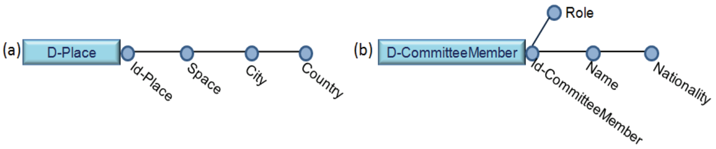

Fig. 4. Dimension D-Place (a) and D-CommitteeMember (b)

- Determination of weak attributes

Some dimension parameters can be characterized by so-called "weak attributes". A weak attribute is a descriptive attribute that gives more meaning to a parameter; it is recommended especially when the parameter values are artificial data (as a Client identifier), and facilitates the understanding of OLAP results.

Rule 9: Case of null- value
Given two attributes *A* and *B*, if there are two symmetrical FDs *A*→*B* and *B*→*A* and, if one of these two attributes can sometimes be null (has the *null* value) then we consider this eventually null attribute as a weak attribute associated to the other element (not containing null values) which becomes a parameter.

Note: This rule considers that an attribute with not null values is more important for OLAP analyses than an attribute containing null values.

Rule 10: Case of not null-valued
Given two attributes *A* and *B*, if there are two symmetrical FDs *A*→*B* and *B*→*A* and, if both attributes do not contain null values then we consider the attribute situated just to the left of the other attribute in the structure (i.e., the previous one in the structure) as the parameter and the right one becomes a weak attribute.

Arbitrary, if *A* and *B* contain both null values then the attribute situated at the left hand side of the structure is the parameter and the right one is its weak attribute.

Naturally, the designer will check in order to approve or reject this arbitrary choice with respect to the semantics of the two attributes in the application domain.

Example: Let us consider the *Movie* sub-tree in the *Festival XSchema* (cf. Figure 2).

*Rule 6 i*dentifies the existence of three non-symmetrical FDs*: Title* → *Year, Title* → *Country* and *Title* → *Director.*

*Rule 9 denotes the e*xistence of two symmetrical FDs *Title* → *Summary* and *Summary* → *Title*; the *Summary* attribute contains a null value and therefore it is elected as a weak attribute for the *Title* parameter.

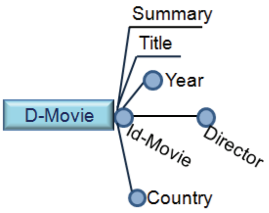

Fig. 5. Dimension D-Movie partially modeled

- **Determination of the temporal dimension**

Generally, the time dimension is included in any multidimensional model [8]. The following rule deals with this dimension.

Rule 11: Temporal dimension
The set of nodes of a same level that describe *temporal* components (e.g., month and day that compose a date) constitutes a temporal dimension (e.g. named *D-Date*) where each node component becomes a parameter in the hierarchy.

Example: The application of *Rule 11* on the *Date* sub-tree *Date* in the *Festival XSchema* (Figure 2) generates the following hierarchy.

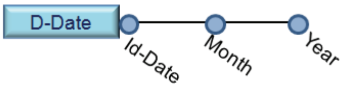

Fig. 6. Dimension D-Date

Rule 12: Iterate rules from 5 to 11 for each element having descendants in order to determine its parameters and weak attributes.

Example: let us continue with the sub-tree *Movie* of the *Festival XSchema*.

According to Rule 10, there exist two symmetrical FDs *Name* → *Tel* and *Tel* → *Name*. *Name* and *Tel* do not contain null values; then we consider *Name* as a parameter and *Tel* is its weak attribute.

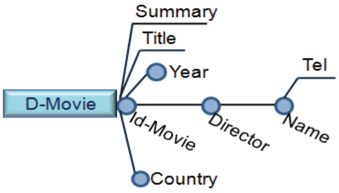

Fig. 7. Dimension D-Movie

Note that the rules we have presented here deal only with the automatic generation of the model (*standard dimensions*) from the logical structure. After that we need to add the *Version dimension* which gathers information about the different versions of each document and the *Semantic dimension* which reflects the semantics of the unstructured textual elements. Those dimensions are introduced in the next section.

5.3 Version and Semantic Dimensions

- Version dimension

Once created, documents are rarely static in time. They can be marked by an evolution in content or structure constituting several versions of the same document. These versions can be considered as different views of the same document. At this level, we add a version dimension called *D-Version,* consisting of an identifier *Id-Version*, the document Contents and two hierarchies:

i) One temporal hierarchy related to the creation date of the document. This hierarchy is organized as follows: *CreationDate* → *Month* →*Year*;

ii) One hierarchy describing a set of metadata (physical document *Name*, extension *Type*, *Author*, summary *Description* of the document...).

It should be noted that each version of the document will be linked with a semantic resource [4] (cf. figure 1).

- Semantic dimension

Textual data convey a fairly rich semantics. This semantics is expressed in the *Diamond* Model through the Semantic dimension, referring to a set of semantics resources and concepts describing the content of documents.

The determination of the semantic dimension for a document is evaluated according to the approach described in [4] where authors uses a taxonomy[3] as semantic resource: i) Extraction of significant terms from leaf elements of the document (leaves of the tree structure of the DTD or XSchema), ii) Choice of a taxonomy describing the semantics of a document, iii) Associate concepts of the selected taxonomy to the leaf elements of the document (concepts that best reflect the semantics of the given terms that describe leaf element), and iv) Inference of concepts for non-leaf elements.

[3] Taxonomy allows a hierarchical representation between its different concepts.

The Semantic dimension in the *Diamond* Model is represented and integrated as follows: i) connection of the Version parameter with the Semantic resource parameter of the semantic dimension and ii) linking the parameters of the standard dimensions, rich in semantic text, with the Concept parameter of the semantic dimension. For example, the Title attribute of the D-Movie dimension will be connected to the Concept parameter. Thus, we can do the analysis by the title of movies or by related concepts (for example, “money”, “power”, “Wall Street” or “New York stock exchange” concepts for “The Wolf of Wall Street” Martin Scorcese’s film).

6 Contribution Review and Future Work

In this paper, we proposed a new multidimensional model, called *Diamond* Model, dedicated to the design of textual data marts and On-Line Analytical Processing (OLAP) on XML documents according to their content and structure. This model mainly consists of a *fact*, a set of *standard dimensions* (constructed from the structure of a set of documents), a *document Version* dimension and a *Semantic* dimension. The main objective of the Semantic dimension is to switch from the simple text to a semantic level.

We also proposed a specific method to semi-automatically generate a *Diamond* model starting from the XML structure (DTD or XSchema) of the collection of documents we want to analyze .We described in this paper the different steps and rules for the generation of the *Diamond* model. All the proposed rules are automated. When the generation of the *Diamond* model is achieved, the designer (assisted by the decision-maker) verifies and validates the obtained *diamond* multidimensional model. He can rename, delete the multidimensional elements and the links between dimensions, or reorganize the parameters and so on.

To illustrate our generation method, we have applied the steps and rules on a collection of Film Festival descriptions (festivals are described by the Festival Xschema of Figure 2) to obtain the *diamond* model shown in Figure 1.

Furthermore, in order to validate our proposals, we have developed a software prototype supporting our rules and allowing a 3D visualization of the generated *diamond* multidimensional model. The *diamond* model corresponding to the Festival collection is depicted in Figure 8. This prototype has been developed in Java using the Oracle RDBMS 10g extended for the management of XML.

Several perspectives for this work are possible. It would be interesting to define a process of instantiation of these diamond models from the contents of documents and visualization of analysis’ results in the cubes’ form or multidimensional tables. It is also important to propose new OLAP operators that take into consideration the specificities of this new multidimensional model. These operators will allow facilitating the interpretation of the results of the multidimensional analyses on the textual data, or at the level of documents or classes of documents (according to the aggregates of analysis). We also intend to exploit techniques of text mining to extract knowledge from documents so as to enhance the semantic dimension in the Diamond Model.

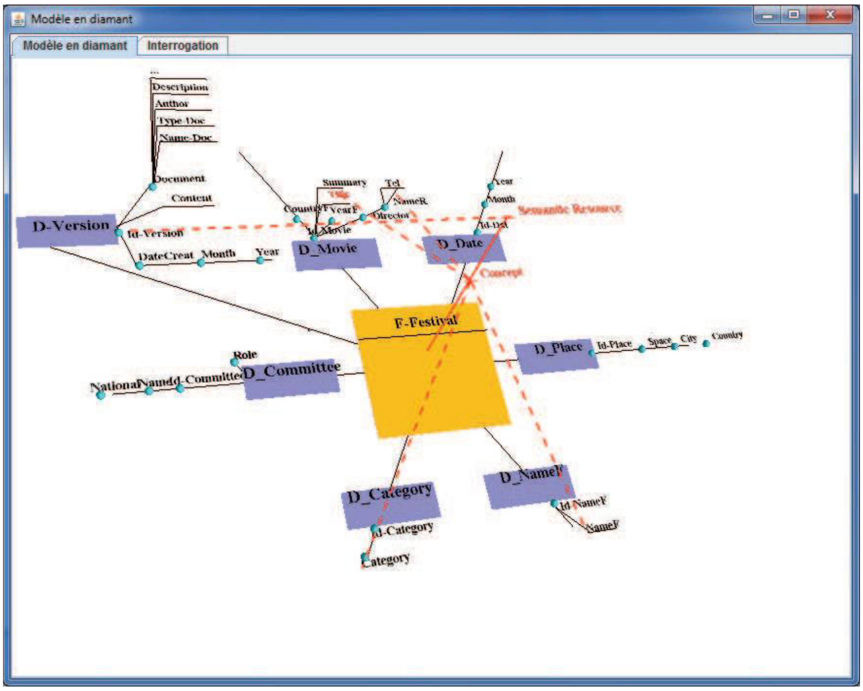

Fig. 8. A 3D visualization of the Diamond multidimensional model automatically generated for the Festival structure of Figure 2

References

1. Aouiche, K., Lemire, D., Godin, R.: Web 2.0 OLAP: From Data Cubes to Tag Clouds. In: Cordeiro, J., Hammoudi, S., Filipe, J. (eds.) WEBIST 2008. LNBIP, vol. 18, pp. 51–64. Springer, Heidelberg (2009)
2. Agrawal, R., Srikant, R.: Fast Algorithms for mining Association rules. In: Proceedings of VLDB, Santiago, Chile (September 1994)
3. Bautista, M., Molina, C., Tejeda, E., Vila, A.: A new multidimensional model with text dimensions: definition and implementation. In: International Conference, IPMU, Dortmund, Germany, pp. 158–167 (2013)
4. Ben Mefteh, S., Khrouf, K., Feki, J., Soulé-Dupuy, C.: Semantic Structure for XML Documents: Structuring and pruning. Journal of Information Organization 3(1), 36–46 (2013)
5. Etcheverry, L., Vaisman, A.A.: Enhancing OLAP Analysis with Web Cubes. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 469–483. Springer, Heidelberg (2012)
6. Hachaichi, Y., Feki, J.: An Automatic Method for the Design of Multidimensional Schemas from Object Oriented Databases. International Journal of Information Technology and Decision Making 12(6), 1223–1260 (2013)
7. Janet, B., Reddy, A.V.: Cube Index for Unstructured Text Analysis and Mining. In: Proceedings of the 2011 International Conference on Communication, Computing & Security, ICCC 2011, pp. 397–402 (2011)

8. Kimball, R., Ross, M.: The Data Warehouse Toolkit. Wiley, New York (2003)
9. Lin, C.X., Ding, B., Han, J., Zhu, F., Zhao, B.: Text cube: Computing in measures for multidimensional text database analysis. In: Eighth IEEE International Conference on Data Mining, vol. 54, pp. 905–910 (2008)
10. Oukid, L., Asfari, O., Bentayeb, F., Benblidia, N., Boussaid, O.: CXT-cube: contextual text cube model and aggregation operator for text OLAP (2013)
11. Yu, Y., Lin, C., Sun, Y., Chen, C., Han, J., Liao, B., Wu, T., Zhai, C., Zhang, D., Zhao, B.: iNextCube: Information network-enhanced text cube. In: VLDB 2009: Proceedings of the 35th International Conference on Very Large Data Bases, Lyon, France (2009)
12. Zhang, D., Zhai, C., Han, J.: Topic cube: Topic modeling for olap on multidimensional text databases. In: SDM 2009: Proceedings of the 2009 SIAM International Conference on Data Mining, Sparks, NV, USA, pp. 1124–1135 (2009)
13. Zhang, D., Zhai, C., Han, J.: Mitexcube: microtextcluster cube for online analysis of text cells. In: The NASA Conference on Intelligent Data Understanding (CIDU), pp. 204–218 (2011)