CAME-BDO: A TOOL FOR DESIGNING DATA MARTS FROM OBJECT DATABASES

Salma Ben Mefteh — Jamel Feki—Yasser Hachaichi

Multimedia, InfoRmAtion systems and Computing Laboratory (Mir@cl)
Faculté des Sciences Economiques et de Gestion de Sfax, University of sfax
3018 Sfax, P.O.Box 1088 – Tunisia
E-Mail: salma_ben_mefteh@yahoo.com : [Jamel.feki , Yasser.hachaichi]@fsegs.rnu.tn

ABSTRACT

This work presents a software tool called CAME-BDO that helps DSS (Decision Support System) designers to construct data mart schemas from an object database. This tool relies on an approach that first gets the object database schema from the object DBMS repository. Secondly, it implements a set of heuristics those extract multidimensional concepts (i.e., facts, measures, dimensions and their attributes), classify them by relevance level and then, generates and returns a set of star schemas. In order to assist the decision makers, CAME-BDO allows them to adjust the generated star schemas according to their analytical requirements. In addition, being identified from objects of the enterprise transactional system, these schemas carry information useful for the implementation of the data mart (e.g., data type, length...), and for the generation of the ETL (Extract, Transform and Load) procedures to feed the data mart with data directly from the enterprise transactional system.

Keywords: Object database, Matisse Object DBMS, multidimensional modeling, star schema, data mart.

1. INTRODUCTION AND RELATED WORK

The data warehouse (DW) technology is a decision support system component based on the federation of information issued from various departments of an organization. The DW design steps cover the conceptual, logical and physical modeling. In conceptual steps, the DW design and implementation tasks are generally defined for DW partial views, called data marts (DM) [16]. Currently proposed DM design approaches are classified into three categories namely Top-down [13], bottom-up [8], [9], [10], [11], [14], [18], [19] and mixed [15].

Bottom-up DM design approaches are also said data-driven approaches because they start directly from the data model of the operational information system of the organization. In fact, these approaches enjoy a twofold advantage: First, they reduce the task of decision makers since they build potential analytical schemas from the data source model and, consequently, they guarantee that the organization’s information system can feed the user-selected schemas with pertinent data. In these approaches, user-requirements collection and study are voluntary neglected. In fact, Bill Inmon argued that requirements are the last thing to be considered in a decision support system development [20] since they are well understood after the DW is populated with data and query results are analyzed. Considering the advantages of data-driven approaches from one hand, and the absence of commercial software tools to help the DW designer designing the DW from the other hand, we elected to propose a data mart design method within this category.

In addition, we noticed that in the literature works, almost all data-driven approaches start either from E/R [5], [14], [9], relational [8] or XML models [11], [12], [18]. In particular, in [15] the authors propose a conceptual design phase that starts from a UML class diagram representing the decision makers’ initial requirements. This UML model is then enriched/transformed in order to facilitate subsequent mapping of the UML conceptual model into a logical multidimensional schema. To achieve this enrichment, the authors define four transformations they apply on the UML model: 1) determination of identifying attributes necessary to determine attributes of the participating classes; 2) manual determination of attributes representing measures; 3) migration of 1-1 and N-1 associations into association attributes; and finally 4) transformation of generalizations into aggregations that represent hierarchies. The resulting class diagram is represented in an extended version of UML that highlights the multidimensional concepts.

Also, the authors of [19] propose the construction of Data Mart (DM) schemas from a UML class diagram in five semi-automatic steps: 1) identification of transaction entities representing facts; 2) construction of decisional UML-packages each containing a transaction entity and its associated classes; 3) graphical annotation of packages to mark multidimensional concepts resulting from the preceding stage; 4) validation of the annotation by the decisional designer; and 5) automatic generation of a DM star schema from each annotated package. This method is supported by a software prototype.

To our knowledge, almost all data-driven approaches relying on an object data-source start from a UML class diagram. However, in practice, such diagram either does not exist within a company, or it is obsolete and does not reflect the latest modifications
due to the evolution of the operational information system.

To overcome this problem, we propose in the remainder of this paper a software tool that implements a data-driven approach to assist the design of DM from an object database. Accurately, it relies on the recent version of the object data source. To reach our goal, we extract this recent version from the repository of the MATISSE Object DataBase Management System (ODBMS). Our tool implements a set of heuristics detailed in [10]; they are for the identification of multidimensional concepts from an object DB and, the construction of DM schemas.

The remainder of this paper is organized as follows. Section 2 briefly presents the standard object data model of the ODMG (Object Database Management Group) and then introduces MATISSE as an ODBMS compliant with this standard. Section 3 presents our software tool called CAME-BDO for the design of DM schemas from an object DB. In section 4, we evaluate CAME-BDO. Finally, the paper terminates with a summary of the presented work and some perspectives.

2. THE ODMG STANDARD AND THE MATISSE ODBMS

In recent years, ODB have been increasingly present within companies [3]. Indeed, there is several Object DBMS on the market [3]. In order to define portable interfaces between these DBMS, the standard of Object Database Management Group (ODMG) has been constituted. In this section, first we describe the main concepts of the ODMG standard and then, we present MATISSE as an ODBMS compliant with this standard. Finally, we give an example of object database for illustrations.

2.1 ODMG: THE STANDARD OBJECT DATA MODEL

In order to define a generic design method, we rely on the standard object data model as defined by the ODMG, a consortium of the leading object DB vendors. This section overviews the object model supporting the last release of the ODMG 3.0 [6].

The object model specifies the semantics that can be defined explicitly by an ODBMS. This semantics determine the characteristics of objects, how objects can be named, identified and related. As shown in Figure 1, the ODMG standard is based on a common object model and uses several aspects of the OMG object model. It supports types (interface), classes (implementation), encapsulation, inheritance and polymorphism. However, few database specific features have been added, among which we find the following (cf., Figure 1):

- A Relationship is a property of an object. The ODMG Object Model supports only binary relationships, i.e., relationships between two objects. A relationship is defined explicitly by declaring the traversal path which implements a logical connection between the objects participating in the relationship. Traversal paths are declared in pairs, one for each traversal direction of the relationship.
- A Key uniquely identifies an instance of a type; simple and compound keys are supported.
- Object IDentifiers (OID) are unique within a storage domain. The value of an OID never changes over the lifetime of an object and is never re-used by other objects.

In terms of typing (cf. Figure 2), a number of predefined Collection Types are available to build complex objects. The Structured Literals Date, Time, Timestamp and Interval are also supported. In addition, the standard model defines inheritance between types (i.e., sub typing): If $S$ is a subtype of a type $T$, then $S$ inherits all the operations and properties of $T$, and $S$ can define new specific operations and properties applicable to its instances.

![Figure 1: ODMG Meta model: Main concepts [6].](image1)

![Figure 2: The ODMG full set of types [6].](image2)

2.2 THE MATISSE ODBMS

MATISSE is an Object DBMS. Its strength lies mainly in its meta-schema (cf. Figure 3). Note that everything is object in MATISSE (i.e., supports inheritance, polymorphism, forums, classes, methods...): Class (instance of Mt-Class class), relations (instance of Mt-Relationship class), indexes (instances of Mt-Index class), attributes (instances of Mt-Attribute class) and methods (instance of Mt-Method class). The meta-class (i.e., the class of all classes) of this schema is the Mt-Class class. In fact, Mt-Class has:
- Two attributes Mt Name (for naming classes) and Mt Instance Check (to associate a constructor function to a class).
- Six relations which associate with a class a list of its attributes (Mt Attributes), a list of relationships (Mt Relationships), a list of super classes (Mt Super classes), a list of subclasses (Mt Successor Of), a list of methods (Mt Own Method), and a list of indexes (Mt Index).

Figure 3: MATISSE Meta-schema [7].

2.3 AN EXAMPLE OF OBJECT DB

To illustrate the application of our extraction heuristics [10] for multidimensional concepts, we adopt the object DB of Figure 4: it models the Media-planning [15] where objects are drawn as rectangles, operations (i.e., methods) are distinguished by the symbol \( \text{Mt} \), attributes and operations are linked to their object by horizontal lines, types of attributes and operations are enclosed between parentheses “(N)umeric, (S)tring, (B)oolean, (D)ate, (I)nterval”; key attributes are tagged with letter K; relationships are shown as lines between objects; and the cardinality of a relationship is indicated by arrows.

This media-planning example (cf. Figure 4) describes the launch of announcements by a company advertising its products using several types of media. The goal is to select the media that will reach a maximum number of consumers.

A Target is identified by a "target_code", has a family situation "status" and a region. For each region, the manufacturers should know the "percentage_of_region" of different targets located in this region. A product is characterized by a code, a name, and has type that determines the product unit. Media are characterized by a "media_name" (i.e., "Channel n") and the "advertising_price". For all media, the decision makers must be able to know its main shareholder at a given date, as well as the "Percentage_of_shareholder" it holds.

Figure 4: A sample ODB modeling Media-planning system [15].
3. CAME-BDO

CAME-BDO is an assistance software tool for the construction of DM schemas from an object database. Its current prototype allows the extraction of the object database schema from the ODBMS repository. Also, it implements a set of heuristics for multidimensional concepts extraction, association relevance. Finally, it displays the generated DM schemas either graphically or in a tabular format. In the following sections, we describe the functional architecture of CAME-BDO tool and then, we illustrate its features (cf. Figure 5).

![Figure 5: CAME-BDO functional architecture.](image)

3.1 EXTRACTION OF THE OBJECT DATABASE SCHEMA

This extraction begins by selecting a MATISSE object DB. After that, we access the MATISSE meta-schema to extract the DB objects and display them in a tabular format as depicted in the interface of Figure 6 where the designer can:
- De/Select objects: this allows keeping all/some objects for the DM construction process.
- Visualize the objects of the MATISSE DB. So, every time an object is pointed out, its attributes, operations, relationships and its type (generalized, specialized or normal) is displayed. If the object is a specialized one then the attributes and methods inherited from their generalized objects are visible in light blue.
- Launch the extraction of the multidimensional concepts in order to obtain DM schemas.

3.2 EXTRACTION OF THE MULTIDIMENSIONAL CONCEPTS

This extraction starts with facts and then follows with measures, dimensions and their hierarchies. We classify all these extracted concepts by level of relevance; this helps the designer to focus on the most important components during the DM design process.

3.2.1 Fact identification

In a DW, the fact concept records observations (i.e., indicators/measures) describing information to be analyzed [10]. CAME-BDO implements two heuristics (cf. [2] [10]) for facts extractions: the first one relies on the relationship \( R \) among two objects \( o1 \) and \( o2 \). It builds a fact on \( R \) when the maximum cardinality from the two sides of \( R \) is greater than one.

Each fact generated by this heuristic is an empty fact, i.e., it has no measures and serves as a record of event (the fact) occurrences [9].

The second heuristic builds a fact among the transitive closure of an object \( o \) containing a numeric non-key attribute(s) or a method returning numeric value(s) [10]. Note that the transitive closure of an object \( o \) (noted \( Tc(o) \)) denotes the set of objects containing \( o \) and all objects \( o' \) directly or transitively linked to \( o \) by relationships with \( \text{Max} (o, o') = (1,1) \).

Applying these heuristics on example of Figure 4, we obtain the two empty facts (light blue colored) and the eight normal facts (dark blue) depicted in Figure 7.

3.2.2 Measure identification

A fact contains a set of measures. In most cases, measures serve to compute summarized results (e.g., total amount of sales by month and year, by product…) using aggregate functions; hence, measures have numeric values.

In our method, we exploit the presence of numerical methods in the object DB in order to identify measures; this is in addition to measures we can build on numerical attributes in a structure attribute. We classify identified measures in three levels of relevance: Each measure extracted from a method is Strong; a numerical non-key attribute is a measure of Medium relevance whereas a numerical attribute in a structure produces a measure of Low relevance [2].

Furthermore, as shown in Figure 2, a numeric attribute can be atomic, or a collection (e.g., set, bag) and therefore is multi-valued. However, in a DW, measures cannot be multi-valued. Thus, when a numeric collection of attributes is identified as a measure, the designer should define a numeric function that calculates a single numeric value. Such a function is obviously semantic/domain dependent [9].

For our running example, Figure 8 shows that the \( \text{percentage of shares} \) is a measure based on a numeric collection of attributes for the fact \( \text{F-Main Shareholder} \). Thus, define the numerical function of Figure 8 to derive a single-value measure.

3.2.3 Dimension identification

A dimension represents information, according to which measures are recorded and analyzed, i.e., it is an analysis axis. It is made up of a finite set of attributes. Some of them take part to define various levels of details (hierarchies), whereas others are less significant but used, for instance, to label results or to restrict data processed in queries. These latter are said weak (or non dimensional) attributes.

For a given fact, we build a dimension either on an object or on an atomic attribute (Boolean, temporal…) or on a collection. In a DW, a dimension reduced to an attribute is known as a degenerated dimension [13].

We classify dimensions into two level of relevance: Strong and weak [2].
**Figure 6:** Extracted schema for the object database "Media_planning".

**Figure 7:** Extracted facts.
CAME-BDO implements two heuristics (cf. [2] [10]) for extracting dimensions from objects: The first one builds a dimension among the transitive closure \( Tc(o) \) of an object \( o \) on which a fact is built [10]. The second heuristic relies on the relationship among two objects \( o1 \) and \( o2 \) such as the maximum cardinality between them is 1 from the side of \( o1 \) and is greater than 1 from the side of \( o2 \), and \( o2 \) is either identified as a fact \( F \). Thus, \( Tc(o1) \) is a candidate dimension for \( F \) [10].

Example:

Figure 9 depicts three dimensions \( \text{Media} \), \( \text{Shareholder} \) and \( \text{Dates} \), extracted using the second heuristic, for the fact \( \text{F-Main_shareholder} \).

3.3 DISPLAY OF THE DM SCHEMA

To display the result of the extraction process (i.e., DM schemas), we use two formats: Tabular and Graphical. The interface of Figure 10 displays the sets of multidimensional concepts extracted with CAME-BDO according to the tabular format. For the selected fact, this interface displays the measures and dimensions per relevance level. Once a dimension is selected, CAME-BDO visualizes its parameters and its hierarchies, partitioned by relevance level (e.g., strong or weak).

The interface of Figure 11 is a graphical display; it shows the DM schema modeled according to the DFM ( Dimensional Fact Model) [9]; this interface is obtained using MPI-Editor [1].

4. EVALUATION OF CAME-BDO

To evaluate CAME-BDO, we applied it on four object-DB; two of them are chosen from the works of the literature (i.e. two ODB, the first is Media-planning [15] and the second is Faculty members load [17]) and for which star schemas are built manually by the authors. The comparison between, on the one hand the results obtained with CAME-BDO and, on the other hand the schemas constructed by the authors of these cases leads to the following conclusions:

- CAME-BDO identifies all the facts that a bottom-up analysis can manually identify.
- The number of facts, measures, dimensions and parameters extracted by CAME-BDO is higher than the number those obtained by the authors of these cases. Thus, CAME-BDO offers a large panoply of analysis. This variation in numbers is explained by the following reasons:
  - CAME-BDO extracts empty facts.
  - In addition to measures extracted from simple attributes, it identifies measures from numerical methods or multi-valued numerical attributes.
  - It extracts dimensions and parameters from objects and atomic Boolean, temporal or structured collection of attributes, whereas the authors of the examined cases extract dimensions and parameters from objects or Date attributes.
  - It generates measures from complex attributes which remedies the problem raised by [4].
Figure 9: Extracted dimensions.

Figure 10: Identified facts, measures and dimensions in Media_planning database.

Figure 11: Example of a star schema built with CAME-BDO and displayed using MPI-Editor tool [1].
5. CONCLUSION

In this paper we have presented an assistance software tool, called CAME-BDO, for the design of DM schemas from the MATISSE object DB. This tool implements a set of heuristics for the identification of facts, dimensions, attributes and hierarchies from an object DB and classifies them by relevance. The output is a collection of star schemas.

Also, we have evaluated CAME-BDO by comparing its generated schemas with schemas manually constructed by the authors of a set of case studies selected from the literature of the DW domain. We can affirm that the obtained results are quite encouraging.

The advantage of this tool is that it keeps traceability between the elements of DM (fact, measure, dimension and hierarchy) and the elements of the source object-database (name of object, attribute, relation or method). This traceability assists the developer to model the logical star schemas issued from the tool. That is, it facilitates the generation of the logical schema and the generation of ETL procedures since each component is associated with a data element in the source data base.

REFERENCES