MP-BUILDER: A TOOL FOR MULTIDIMENSIONAL PATTERN CONSTRUCTION

Mounira Ben Abdallah — Nadia Ben Saïd — Jamel Feki— Hanene Ben-Abdallah

Multimedia InfoRmAtion Computing Laboratory (MIRACL)
Faculté des Sciences Economiques et de Gestion de Sfax
B.P. 1088 Sfax, Tunisie
E-Mail: {Mounira.Benabdallah, Jamel.Feki, Hanene.Benabdallah}@fsegs.rnu.tn

ABSTRACT

A multidimensional pattern (MP) is a generic solution of typical analytical requirements in a given domain. It can be instantiated to specify particular OLAP (On-Line Analytical Processing) requirements, to build the corresponding data mart (DM) schema and eventually to load the schema from a data source. This paper presents a toolset, called MP-builder, that implements our MP construction method. It illustrates the functionalities of MP-builder through an MP analyzing the "sales" fact in the commercial domain.

Keywords: *Decisional support system, multidimensional pattern, data mart design, commercial domain fact.*

1. INTRODUCTION

Within today's competitive economic context, information plays a crucial role in the day-to-day activities of every enterprise. In fact, information acquisition, analysis and exploitation became strategic and unavoidable choices for an enterprise. Moreover, in order to guarantee their persistence and growth, enterprises are forced, henceforth, to capitalize an expertise in this domain.

Data warehouses (DW) emerged as a potential solution answering the needs of storage and analysis of large data volumes. A DW is a database system specialized in the storage of data used for decisional ends. This type of systems was proposed to overcome the incapacities of OLTP (On-Line Transactional Processing) systems in offering analysis functionalities. It offers integrated, consolidated and temporized data to carry out decisional analyses.

On the other, the different objectives and functionalities between OLTP and DW systems created a need for a development method appropriate for DW. In practice, the DW design and implementation steps are generally defined for DW partial views, called *data marts* (DM) [16]; the content of a DM is extracted from the DW and adapted for a particular analytical requirement.

Currently proposed DM design approaches are classified into three categories. The first category of methods starts from the decisional requirements to build DM schemes [13] [18]. Being solely based on the requirements, the methods in this category may produce models unloadable from a target OLTP system. On the other hand, the second category of methods overcomes this problem by basing the DM design on the enterprise data model [10] [11] [4] [14]; however, these approaches limit the decision maker's participation in the design by ignoring their analytical requirements. Finally, the third category of methods, called *mixed* approaches [3] [5] [17], combines the first two categories in order to resolve their drawbacks and profit from their advantages: It involves the decision maker by taking their analytical requirements into account, and it guarantees the loadability of the resulting DM schema by confronting the requirements to the data model of the OLTP system.

Despite their tangible advantages, the mixed approaches require a double expertise: requirement specification and data model comprehension. However, such a double expertise is, often, not at hand for either decision makers or OLTP system designers.

To tackle this limit of mixed approaches, we have proposed a DM design approach [1] [8] inspired from development through reuse of *design patterns* [9]. More specifically, our DM design approach overcomes the limit of the mixed approaches in two ways. First, it bases the design of a DM on *typical* DM schemes in a fixed domain; that is, it assists the decision maker in specifying their analytical requirements. Secondly, it defines the typical DM schemes in a way that facilitates the schema projection on the data model of the OLTP system; that is, it guarantees the loadability of the specified requirements.

The hypothesis behind our DM design approach relies on the fact that, independently of any design method, all analytical requirements within a fixed domain share several elements. Thus, re-using generic and proven solutions in the specification of one's analytical requirements should accelerate this first step in the development of a DW. Being generic, the proven solution is independent of any particular OLTP system's data model and can be expressed in terms of analytical elements (facts, measures, dimensions …).

In our previous works, we have introduced this concept of generic solutions as *multidimensional patterns* (MP). An MP is a generic, typical star-schema defined in one entrepreneurial activity domain. It describes an analysis subject (fact) according to a collection of significant perspectives. An MP can be re-used to design a particular DM [6] [7].

In this paper, we present a toolset, called *MP-builder,* that implements our MP construction method [1] [8]. We start by presenting the concept of MP and over viewing the MP construction steps. Then, in Section 3, we detail out the MP-builder functionalities. Finally, in Section 4, we outline functional extensions we are currently examining for MP-builder.

2. MULTIDIMENSIONAL PATTERN CONSTRUCTION: AN OVERVIEW

Our MP construction method [1] [8] is a document-based approach: it extracts analyzable data from any information artifact that circulates within the enterprises. These information artifacts, we call *real-world entities* (RWE), represent and/or are generated from data stored in the enterprise OLTP system. As examples of RWE, we find invoice, delivery order, customer file, a screen form of a transactional application...

Before outlining our MP construction method, we next review the definition of an MP.

2.1 MP DEFINITION

In our approach, an MP is a star schema describing one analysis subject (fact) according to a collection of significant axes (dimensions). Being a star schema, an MP is composed of one fact representing the star's center and containing at least one measure; around the center, other data describe the dimensions along which data analyses can be performed through various parameters.

Each MP element (measures, dimensions and parameters) is identified with a unique, standard name and is documented with one or several RWE from which it was extracted.

An MP plays a double role:

1. Requirement engineering: an MP is a typical solution of analytical requirements and can be reused by a decision maker to specify their particular analytical requirements [6] [7].
2. Decision support system (DSS) development: the RWE documenting an MP can be used to project the DM schema derived from the MP on the data model of a particular OLTP system, and later to define the ETL (Extract, Transformation, Loading) procedures needed to load the DM from the data source [1] [8].

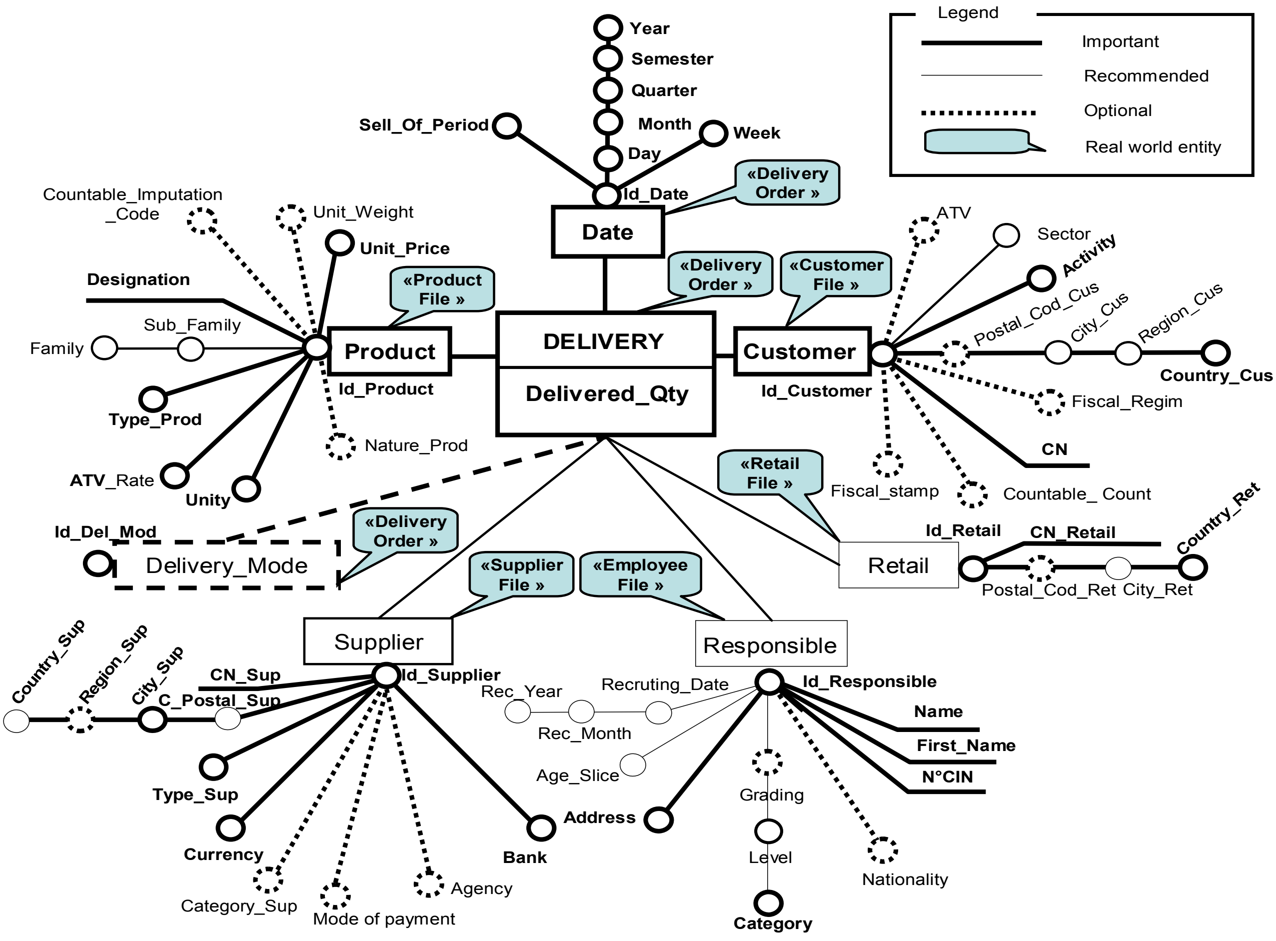

Figure 1: P: MP analyzing delivery in the commercial domain.

Example

Figure 1 illustrates an example of MP analyzing the "*DELIVERY"* fact in the commercial domain. This fact measures the "*Delivered_Qty"* according to seven analysis axes: "*Customer*", "*Date"*, "*Product*", "*Retail*", "*Responsible*", "*Supplier*", and "*Delivery_Mode*". These analysis axes are performed through various parameters. These parameters are ordered in hierarchies such as the hierarchy "*Id_Customer < Postal_Cod_Cus < City_Cus < Region_Cus < Country_Cus*" which represents the address of a "*Customer*".

The MP presented in the Figure 1 is constructed and documented by the RWE "*Delivery Order"*, "*Customer File*", "*Employee File"*, "*Retail File"*, "*Product File"*, and "*Supplier File*".

2.2 MP CONSTRUCTION

As mentioned above, our MP construction approach is based on RWE commonly circulating within enterprises active in a given sector. This document-based approach has two main advantages. On the one hand, it ensures that the constructed MP covers the majority of typical analytical requirements for one activity domain of the enterprise. On the other hand, since the RWE are generated by the enterprise operational system, the analytical requirements derived from an MP are guaranteed to be loadable. In fact, the MP-RWE association facilitates the mapping of the MP elements onto data in the target IS through this latter's RWE. Furthermore, it facilitates the generation of the loading procedures by limiting the necessary data source elements.

To guarantee the construction of generic MP, our construction method starts from a large sample of RWE belonging to different enterprises. It then operates over three steps: standardization of the RWE, classification of these entities, and identification of the multidimensional elements of an MP (fact, measures, dimensions, hierarchies, parameters and attributes). We next detail these three steps.

2.2.1 RWE standardization

As mentioned above, the construction of an MP relies on an empirical study carried out on a significant number of RWE used in a particular domain within a set of enterprises [1]. Such a collection of RWE may differ in their elements, element names and structures. To harmonize the RWE and to identify the presence rates and thus importance of their elements, our MP construction method starts with a standardizing step.

Over all, the RWE standardizing produces the most complete RWE from the diverse RWE. The produced RWE contains the "union" of the elements in their most detailed structures and their synonymous names. In addition, the RWE standardization calculates the presence rate of each RWE element as an indicator of its importance in the absence of its legal requirement. The importance of each RWE element is an indicator of its genericity level.

2.2.2 RWE classification

Once standardized, the RWE are then classified into two main classes depending on their persistence in the IS: ***fact entities*** (FE) and ***basic entities*** (BE). The fact entities (*e.g.*, invoice, purchase order ...) result from the enterprise transactional activities, whereas the basic entities (*e.g.*, customer, product ...) contain permanent/persistent data used to generate fact entities. Thus, a FE refers to a set of BE either directly or indirectly. Depending on the reference type, a BE can be classified into one of two classes:

1. When a BE is directly related to a FE, we call it a ***dimension entity***. It contains information answering the questions "who", "what", "when", "where", "how" and "why" of the FE
2. When a BE is related to a FE via one (or more) BE, it contains information pertinent to the intermediate BE. Hence, in general, it is not an analysis dimension for the FE; rather, it constructs hierarchies in the dimension obtained from the intermediate BE (the dimension entity). However, sometimes the BE indirectly related to the FE might be more relevant to the FE. In this case, our classification must be assisted by the MP designer's expertise. This latter may recommend to construct dimensions from this type of BE rather than to construct hierarchies.

2.2.3 MP element identification

Once classified, the RWE are examined according to a set of rules to identify the MP elements (measures, dimensions, attributes...). In our MP construction method, each FE generates an analysis fact (*e.g.*, *Invoicing*, *Delivery…*). The measures of a fact are determined among the FE attributes having values with multiple occurrences. These measures are either elementary (*e.g.*, *Delivered_Qty Unit_Price*,…) and necessary for the identified fact, or aggregate (*e.g.*, *Total,…*).

Once a fact and its measures are identified in a FE, the dimension attributes (*e.g., Name, City_Customer*,…) are extracted from the BE related to the FE. These attributes are all of the elements in the corresponding BE. In addition, some dimension attributes can be semantically ordered from the most specific to the most general to form candidate hierarchies; the attribute ordering requires domain knowledge.

Naturally, all the MP elements do not have the same importance/genericity in the domain. To assist a decision maker in selecting the MP elements they ought to keep, our method proposes to distinguish the *core* from the *variable part* of an MP. For this, it associates with each element its presence rate collected during the RWE standardization step. These rates can be used to define several genericity levels for the MP elements. We recommend three genericity levels: *important, recommended* and *optional.* However, the number of genericity levels depends on the domain of the RWE and the coverage rate of empirical study. Our method allows the MP developer to fine tune this number to meet their needs.

Example
In our example of Figure 1, we fixed three genericity levels for this MP's decisional elements: *important, recommended* and *optional.* We considered *important* elements all those elements with presence rates close to 100%. Being omni present, these elements should not be removed during the re-use phase of an MP; they represent the almost invariant core of the MP and have a high analysis potentiality.

On the other hand, to avoid constructing a complex MP, we decided to eliminate infrequent elements (those with presence rates close to zero).

For the remaining MP elements, we noticed presence rates concentrated around two values in our empirical study. The first rate values were around 75%; we tagged these elements as *recommended*. The second rate values were around 25%; we considered these elements as *optional.* The *recommended* and *optional* elements form the MP's *variable part.* That is, they are susceptible to be adapted to meet specific requirements in the re-use phase.

To represent visually the MP genericity levels, we introduced the graphical notation illustrated in Figure 1. According to this notation, the dimension "*Customer"*, the measure "*Delivered_Qty*" and the parameter "*Nature_Prod"* are important. As for the dimension "*Responsible"* and the parameter "*Sector*" are regarded as recommended. Among the optional multidimensional elements, we find the dimension "*Delivery_Mode*" and the parameter "*Unit_Weight*".

3. MP-BUILDER: A TOOL FOR MP CONSTRUCTION

MP-Builder is an assistance toolset for MP developers. As shown in Figure 2, the current *MP-Builder* prototype allows the capture of the two RWE classes, the standardization of the captured RWE, and finally the MP element identification and MP storage. Once constructed, an MP can be visualized and reused through our tool *MPI-Editor* [2].

In the following sections, we detail the *MP-Builder* functional architecture (Figure 2) and illustrate them through the construction of the MP shown in Figure 1.

3.1 RWE CAPTURE

This first step of an MP construction consists in capturing RWE of a particular domain according to the two classes: FE and BE. *MP-Builder* proposes for each class of entities a standard structure ensuring the capture, the modification and the deletion of an entity in these two classes. Figure 3 shows the meta-model of RWE adopted by *MP-Builder*. This meta-model was derived after the study of a large set of RWE in various domains and sectors.

While the RWE in each class may differ from one enterprise to another, at an abstract level, they share several elements. Indeed, any BE has an identification information (identifier, description) and other descriptive/numerical information. In addition, it may contain dates and refer to other BE. These latter elements can be either elementary or complex, *i.e.*, they can be composed of other elementary data. This structure is represented in Figure 3 by the aggregations between the BE meta-class and the "*Identification_Data"*, "*Descriptive_Data*" and "*Other_Data*" meta-classes; in addition, a BE can also aggregate other(s) BE.

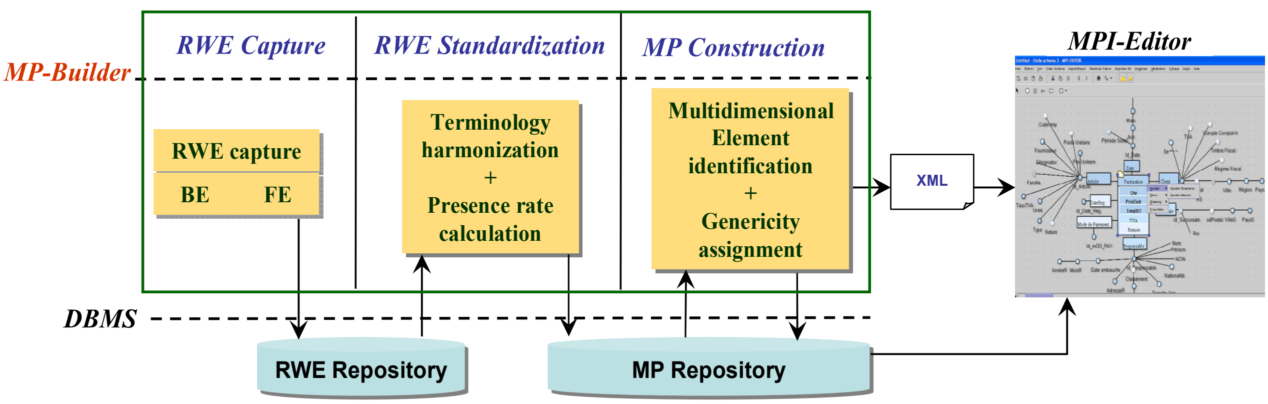

Figure 2: Functional architecture of MP-Builder

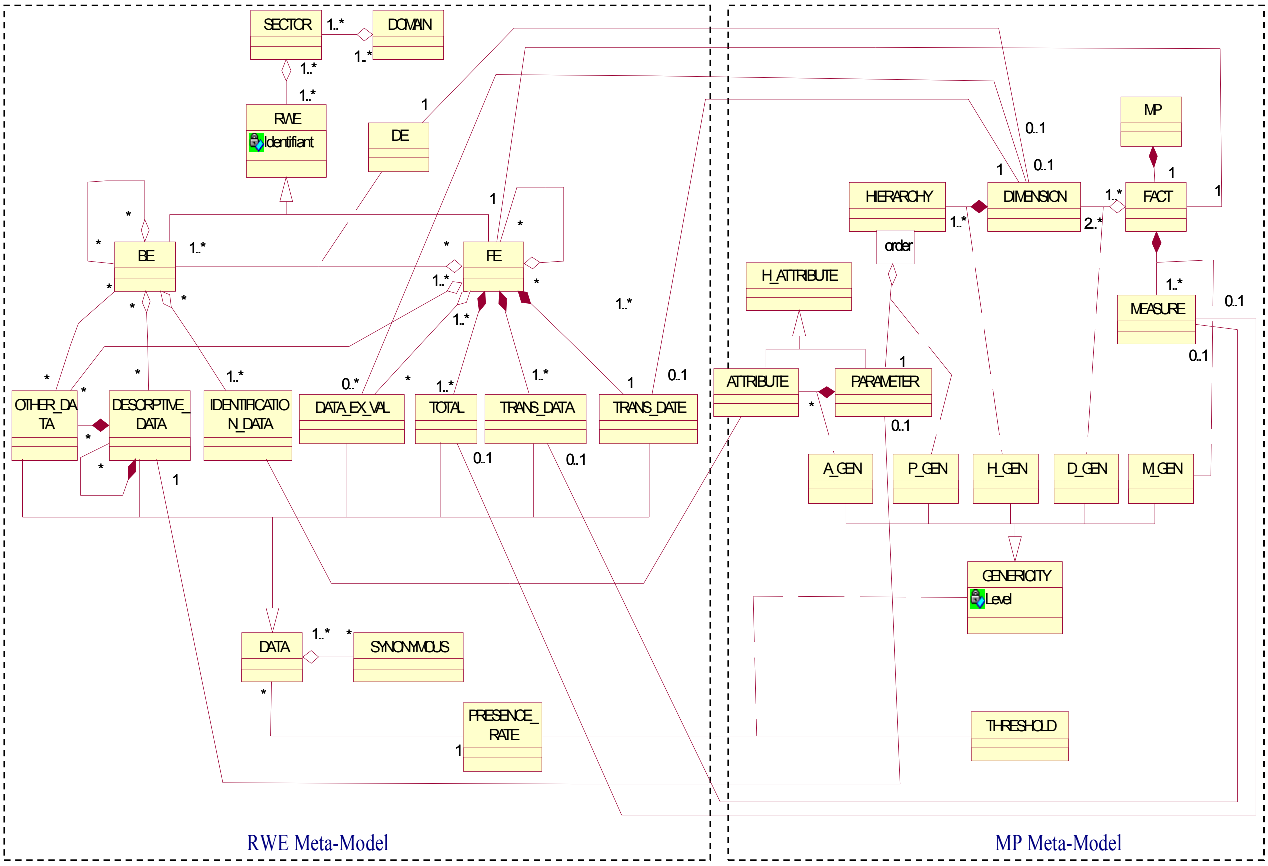

Figure 3: MP Meta-model of MP-Builder.

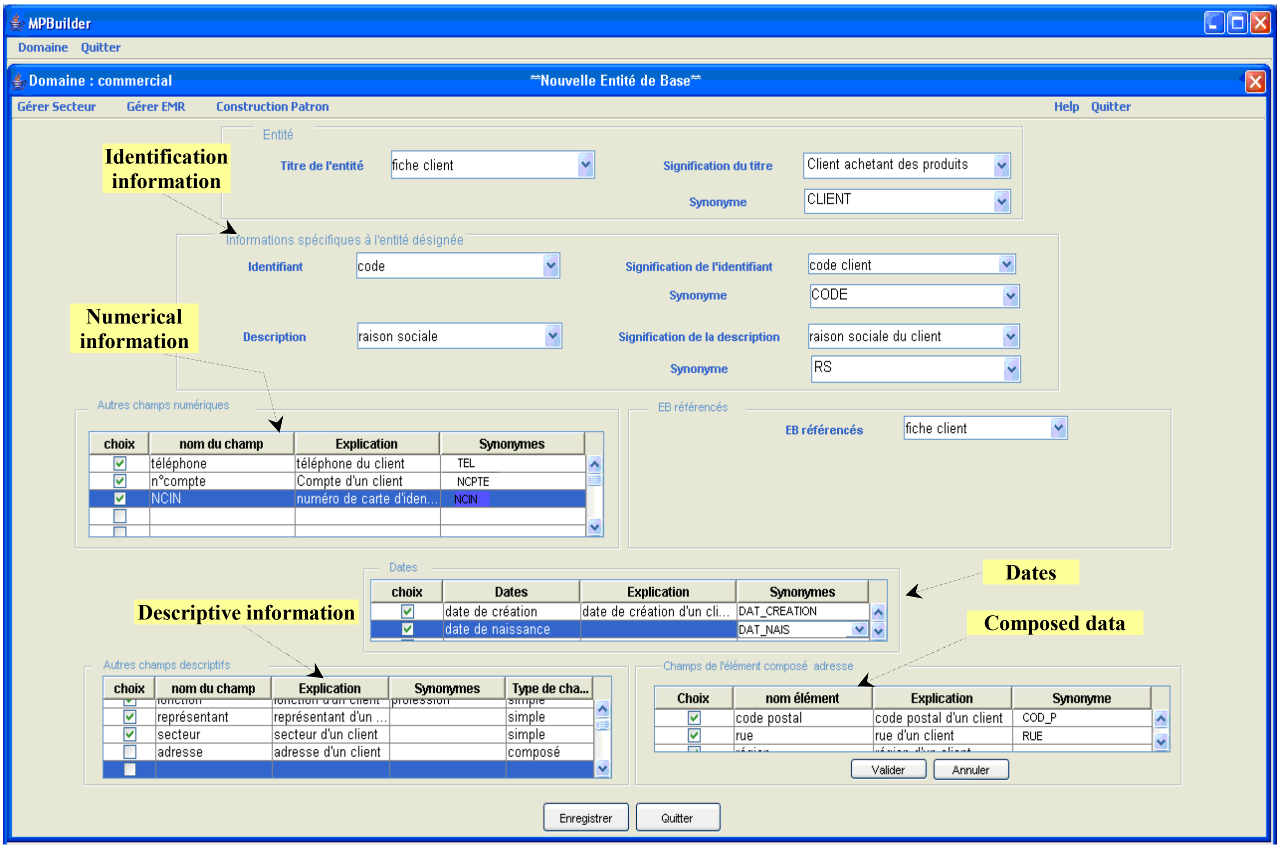

Figure 4: A "Customer file" captured according to MP-Builder standard interface.

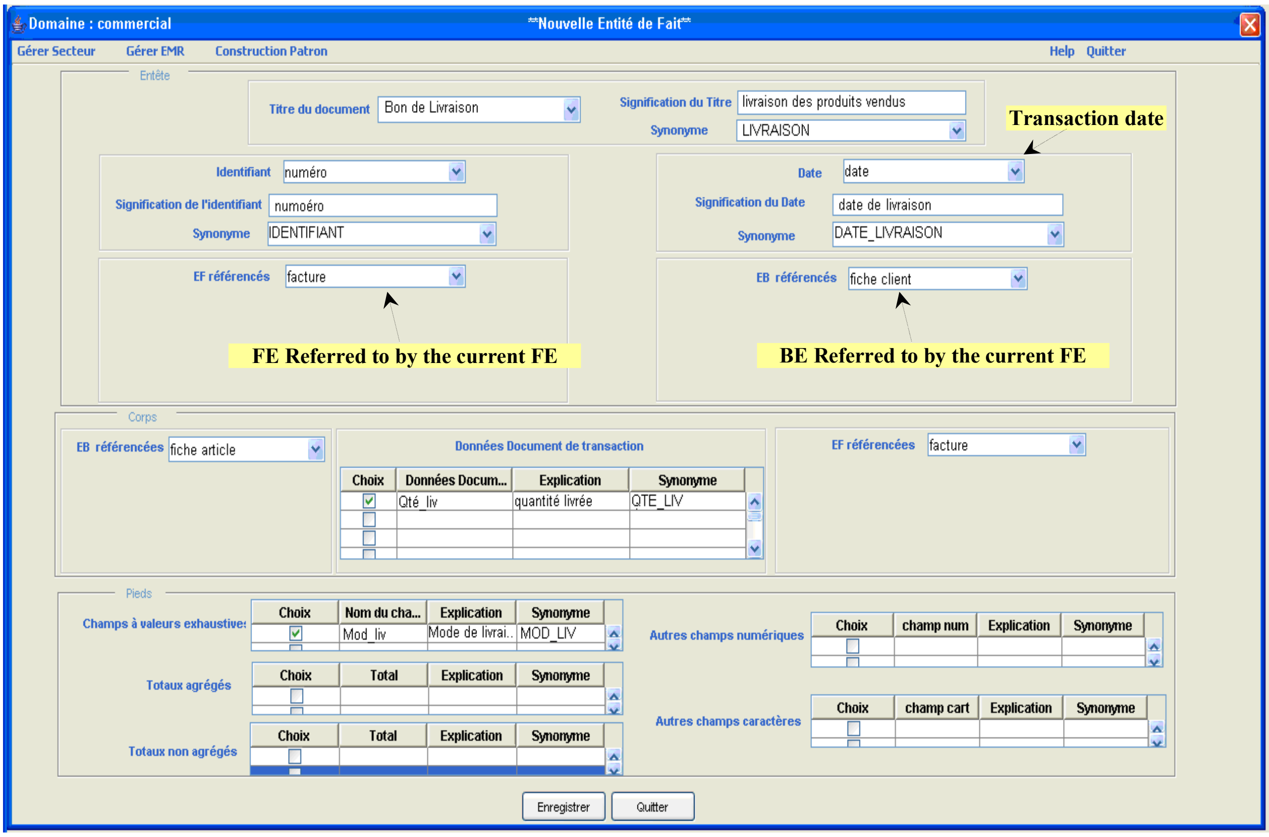

Figure 5: a "Delivery order" captured according to MP-Builder standard interface

On the other hand, any FE is organized in three parts: *heading, body* and *summary.* The heading contains identifying and descriptive information in addition to transactional dates. Moreover, a FE refers to one or more BE. The body contains transactional data pertinent to a BE and/or refers to other FE. Finally, the summary generally contains aggregate totals and/or data with exhaustive values.

In the *MP-Builder* meta-model (Figure 3), each FE is composed of a "*Trans_Date*", one or more "*Trans_Data*" and "*Total*" data. It may contain some "*Data_Ex_Val*" or refers to some other FE. Moreover, it aggregates some BE representing its DE.In terms of user interface, *MP-Builder* offer two GUI to allow the interactive capture of FE and BE. Figure 4 and Figure 5 illustrate the GUI to capture the BE “*Customer File*” and the FE “*Delivery Order*” through *MP-Builder.*
Note that, in certain domains, most of the FE elements are required either by the law, or by a standard (*e.g.,* EDI, EDIFACT). For the commercial domain, for example, the "*Delivered_Qty*" is required by the Tunisian law.

3.2 RWE STANDARDIZATION

Once the RWE are captured by domain and according to the two classes (BE and FE), the standardization step aims at harmonizing the terminology of elements used in a RWE type (*e.g., Delivery Order*) of a given class. This harmonization is manually carried out in the current *MP-Builder* prototype by choosing, from the set of the collected terms for an element, the one considered standard (*e.g.*, more used and/or legally required). Moreover, this step automatically calculates the presence rates of the found elements and orders increasingly. This element ordering helps fixing thresholds and identifying the number of genericity levels.

Example

Figure 6 illustrates the standardized “*Customer File”* RWE

3.3 MULTIDIMENSIONAL ELEMENT IDENTIFICATION

By applying the MP element identification rules, *MP-Builder* initially extracts all the facts from the standardized FE. By selecting the fact to analyze, our prototype shows the set of its measures and its dimensions per genericity level. Once a dimension is chosen, *MP-Builder* visualizes its parameters and its hierarchies, partitioned into levels (*e.g.,* important, recommended and optional). In addition, for each hierarchy or indicated parameter, *MP-Builder* distinguishes the hierarchy parameters ordered from the finest to the most general or the attributes of the indicated parameter.

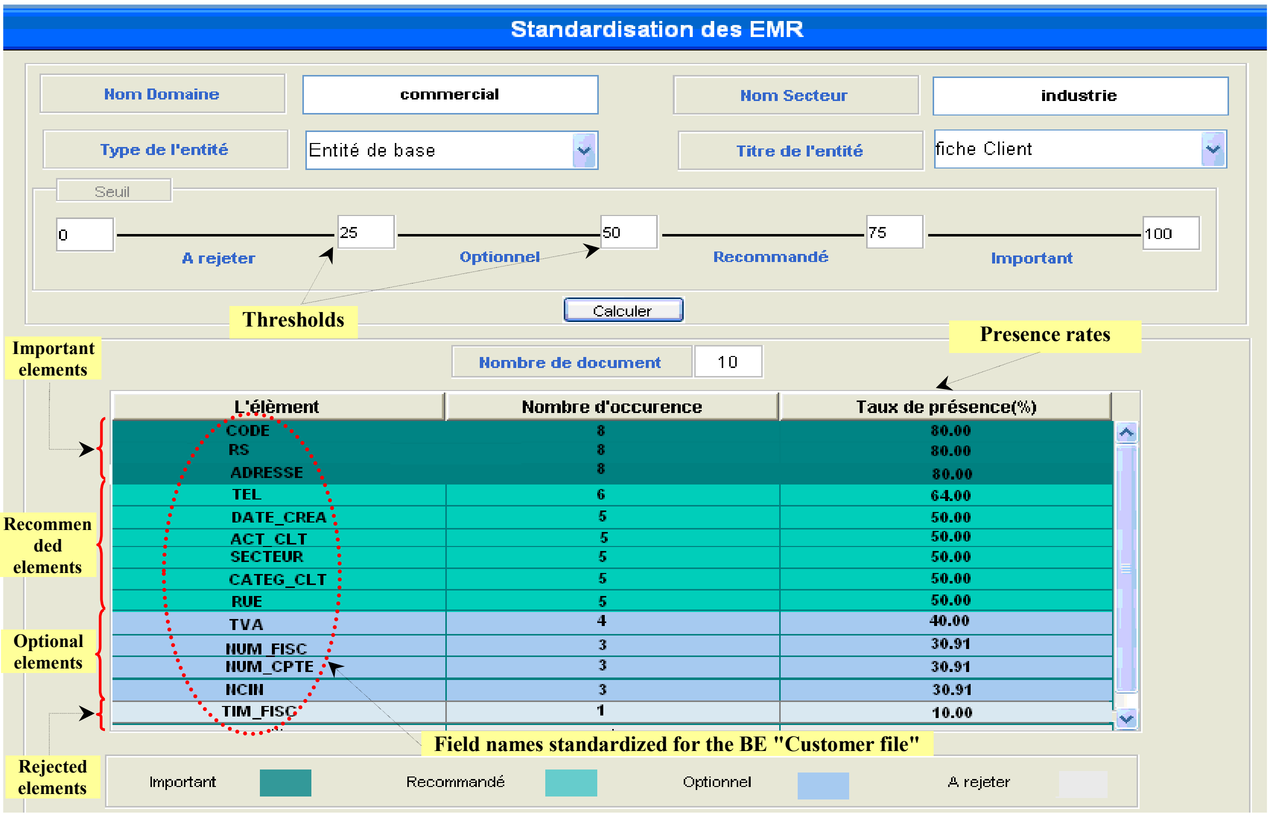

Figure 6: The basic entity "Customer file" standardized with MP-Builder.

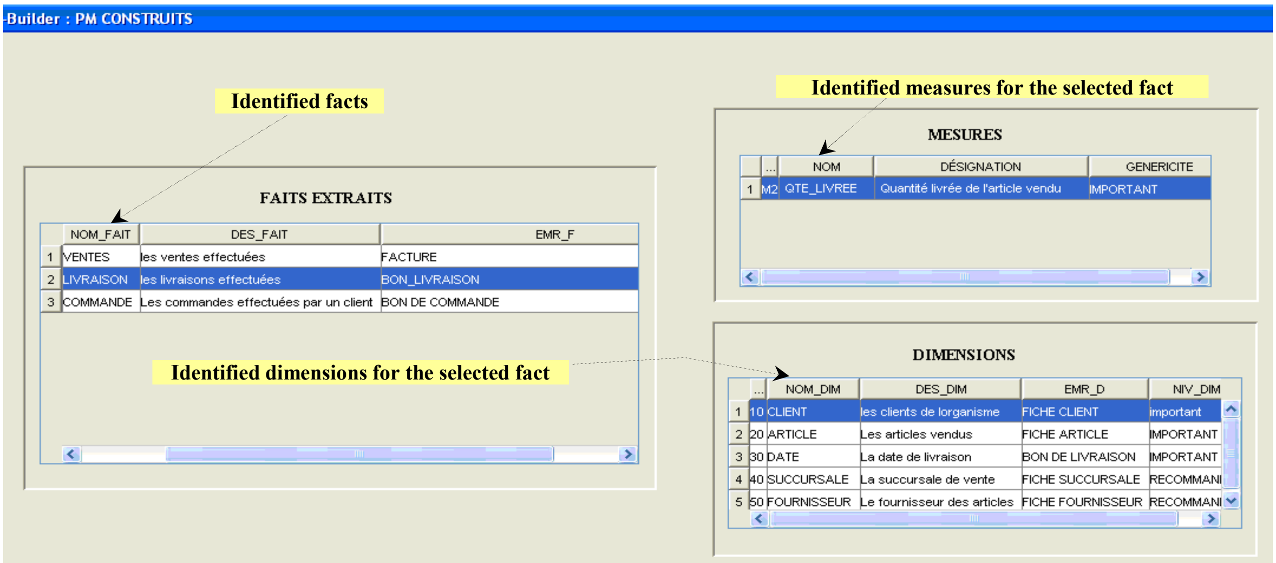

Figure 7: Identified facts, Measures and dimensions in the commercial domain.

Example

Figure 7 illustrates the GUI of MP-Builder showing three facts extracted from the FE in the commercial domain: "*VENTES*" (*SALES*), "*LIVRAISON*" (*DELIVERY*) and "*COMMANDE*". After selecting the fact "*LIVRAISON*", *MP-Builder* shows the list of corresponding measures *(e.g.*, "*QTE_LIVREE*" (*Delivered_Qty*) in Figure 7) and dimensions (*e.g.*, "*Client*" (*Customer*) in Figure 7).

Throughout this identification step, *MP-Builder* interacts with the MP designer in order to validate the set of identified multidimensional elements of each concept (fact, measure, dimension, hierarchy, parameter and attribute); he/she can rename, eliminate some multidimensional elements and/or reorder some hierarchy parameters. Once the identified multidimensional elements are validated, MP designer can store the built MP either in an XML format or in the reference model of Figure 3, and managed by a relational database. The stored MP can be later edited with our *MPI-Editor* [2] tool for OLAP requirements specification in order to derive a DM schema for a target IS in the reuse phase [6], [7].

4. CONCLUSION

This paper presented an assistance toolset, called *MP-Builder,* for multidimensional pattern (MP) construction. An MP is a star schema describing an analysis subject (fact) according to a collection of significant perspectives (dimension hierarchies...); it has a double role: 1) it is a typical solution of analytical requirements, generic and reusable for the specification of analytical requirements, and 2) it is a solution, documented by real world entities (RWE) such as invoices, orders…, that assist in building a DM schema and loading it from a data source.

The M*P-Builder* toolset offers a GUI to allow MP designers to capture different types of RWE. In addition, it standardizes the captured RWE by harmonizing the RWE elements and calculating their presence rates in a given activity domain. Finally, it automatically identifies the multidimensional elements building the MP. *MP-Builder* provides for the storage and retrieval of RWE and MP in an XML-based library of captured RWE and constructed MP, respectively.

Currently, we are completing the *MP-Builder* toolset with the constraints ensuring the well-formedness of the MP and every DM schema [12] derived through MP reuse.

REFERENCES

[1] Ben Abdallah M., Feki J., Ben-Abdallah H., Designing Multidimensional Patterns from Standardized Real World Entities, *Proc. Of the International Conference on Computer and Communication Engineering ICCCE'06*, Kuala Lumpur Malysia, 9-11 Mai 2006.

[2] Ben Abdallah M., Feki J., Ben-Abdallah H., MPI-Editor : un outil de spécification de besoins OLAP par réutilisation logique de patrons multidimensionnels, *Ateliers sur les Systèmes Décisionnels ASD 2006*, Agadir, Morocco, 6 December 2006.

[3] Bonifati A., Cattaneo F., Ceri S., Fuggetta A., Paraboschi S. Designing Data Marts for Data Warehouse, *ACM Transaction on Software Engineering and Methodology, ACM*, vol. 10, pp. 452-483, October 2001.

[4] Cabibbo L., Torlone R., A logical Approach to Multidimensional Databases, *EDBT'98*. Spain 1998.

[5] Carneiro, L., Brayner, A., X-META : A Methodology for Data Warehouse Design with Metadata Management, *4th International Workshop on Design and Management of Data Warehouses DMDW'02*, Toronto, Canada, pp. 13-22, May 2002.

[6] Feki, J., Ben-Abdallah H., "Multidimensional Pattern Construction and Logical Reuse for the Design of Data Marts", *International Review on Computers and Software I.RE.CO.S.* 2007.

[7] Feki J., Ben-Abdallah H., Star Patterns for Data Mart Design: Definition and Logical Reuse Operators, *International Conference on Control, Modeling and Diagnosis ICCMD'06*, Annaba Algeria, 22-24 May 2006.

[8] Feki, J., Ben-Abdallah H., Designing Data Mart Schema from Real World Entities, *International Arab Conference on Information Technology (ACIT'05)*, Al-Isra University Jordan, 6-8 December 2005.

[9] Gamma E., Helm R., Johnson J. and Vlissides J., *Design patterns: Elements of reusable Object Oriented Software*, Addisson-Wesley, Reading, MA,1995.

[10] Golfarelli M.., Maio D., Rizzi S., Conceptual design of data warehouses from E/R schemes, *31st Hawaii International Conference on System Sciences* 1998.

[11] Golfarelli M., Rizzi S., A Methodological framework for Data Warehouse Design, *DOLAP'98*, USA 1998.

[12] Hurtado C.A., Mendelzon A.O., OLAP Dimension Constraints. *21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems PODS'02*, Madison, USA, pp. 169-179 June 2002.

[13] Kimball R., Ross M., *The Data Warehouse Toolkit*, Wiley, New York, second edition 2002.

[14] Moody L. D., Kortink M. A. R., From Enterprise Models to Dimensional Models: A Methodology for Data Warehouses and Data Mart Design, *International Workshop on Design and Management of Data Warehouses*, Stockholm, pp. 5.1-5.12, Sweden 2000.

[15] Phipps, C., Davis, K., Automating data warehouse conceptual schema design and evaluation, *DMDW'02*, Canada 2000.

[16] Ravat, F., Teste, O., Zurfluh, G., "Modélisation multidimensionnelle des systèmes décisionnels", *Revue ECA*, Volume 1 - n°1-2/2001 - ISBN 2-7462-0216-6.

[17] Trujillo, J. C., Luján-Mora, S., Song, I., *Applying UML for designing multidimensional databases and OLAP applications*. In Advanced Topics in Database Research, K. Siau (Ed.), Vol. 2, Hershey, PA: Idea Group Publishing, pp. 13-36, 2003.

[18] Tsois, A., Karayannidis, N., Sellis, T., MAC: Conceptual Data Modeling for OLAP. *In Proceedings of the International Workshop on Design and Management of Data Warehouses DMDW'2001* Interlaken, Switzerland, June 2001.