Towards Multidimensional Requirement Design

Estella Annoni, Franck Ravat, Olivier Teste, and Gilles Zurfluh

IRIT-SIG Institute (UMR 5505, University of Paul Sabatier
118 Route de Narbonne, F-31062 TOULOUSE CEDEX 9
{annoni, ravat, teste, zurfluh}@irit.fr

Abstract. Data warehouses (DW) main objective is to facilitating decision-making. Thus their development has to take into account DW project actor requirements. While much recent research has been focused on the design of multidimensional conceptual models, only few works were done to develop models and tools to analyze them. Despite specificities (OLAP, historicization, ...) of these requirements, most of the previous works used an E/R or UML schemas which do not allow designers to represent these specific principles. A main property of DW is that they are derived from existing data sources. Therefore, Extraction-Transformation-Loading (ETL) processes from these sources to the DW are very useful to define a reliable DW.

In this paper, we fill this gap by showing how to systematically derive a conceptual schema of actors requirements using a rule-based mechanism. We provide a conceptual model which tries to be close to user vision of data by extending an object-oriented multidimensional model. Hence, in the early step of DW engineering, designers can formalize actors requirements about information and ETL processes at the same time to make easier understandability, confrontation and validation by all DW actors.

1 Introduction

Building a data warehouse (DW) that satisfies tactical requirements with respect to existing data sources is a very challenging and complex task since it affects DW integration in companies. In addition to tactical requirements in traditional information systems, data warehouse development takes as input requirements existing source databases called system requirements. Moreover, we distinguish strategic and tactical requirements from tactical requirements. The strategic requirements correspond to key performance indicators which make it possible to take decisions about high-level objectives; they are expressed by DW business group. On the other hand, tactical requirements represent functional objectives expressed by end-users group. These two latter requirements are complementary of each other. Therefore, we split DW requirements into three groups as in [1] to analyze separately each one according to their specificities. Hence, the design must distinguish between tactical and strategic requirements and can easily design and handle all DW inputs (tactical, strategic, system).

Previous works of DW design methods implies that the requirements are specified by a classic E/R model. However, Kimball in [2] argues that this model cannot be used as the basis for a company data warehouse. Other works use object-oriented modelling because of the popularity of the UML model which results in reusing of models, tools

A Min Tjoa and J. Trujillo (Eds.): DaWaK 2006, LNCS 4081, pp. 75–84, 2006.

and so forth. But this model has the same drawback as argues Kimball for the E/R model, which is the lack of ability for DBMS to navigate for DW project. Moreover, these works do not exploit one of the great advantages of the object-oriented model which is the definition of the operations.

In this paper, we focus on tactical and strategic requirement analysis step of our method [3] in order to model a data warehouse system in precise, complete and user-friendly manner. We provide a model to represent tactical and strategic requirements close to decision-makers' vision. This model represents both information and processes related to these information. In DW development, they are defined before the design of the multidimensional conceptual schema.

In the following sections, we present progressively our running example. The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 presents our tactical requirement analysis. Section 4 describes our strategic requirement analysis. Finally, Section 5 points out the conclusion and future work.

2 Related Work

The DW requirement specificities imply proposition of several design methods different from those of traditional information systems. (IS). However, as [1] argue DW requirement analysis process has not been supported by a formal requirement analysis method. These authors define the three groups of actors, but they do not provide any model to represent DW requirement specificities of these three levels.

Most of the previous propositions of DW design implies that this step is already done. Main works do not use specific methods for this step as [4], [5], [6]. The authors of [4] consider this issue by collecting and filtering the tactical requirements using a natural-pseudo language. This expression is interesting because it is close to the natural language but it requires DW designer to handle informal and ambiguous tactical requirements. Likewise the authors of [7], they only take into account information in conceptual schema and processes associated are not analyzed. Moreover, [5] and [6] use UML diagrams resulting in tactical requirement analysis process but they do not specify explicitly how to analyze DW requirements and integrate their specificities. All these works do not distinguish between tactical and strategic requirements and thus do not handle their specificities.

Besides, some approaches of requirements gathering has been provided. [2] considers that the main task is the choice of the business process. According to his experience, the author describes a modelling of the project from the functional requirements. Thus, he considers only tactical requirements. The approach presented in [8] shares similarities with ours, e.g the distinction between tactical and strategic requirements, but it does not define requirement about ETL processes.

3 Tactical Requirement Analysis

3.1 Collection of Tactical Requirements

In order to avoid managing tactical requirements in an informal and ambiguous way, we recommend to use a sample of representative analyses used by decision-makers shown

in table 1. These analyses are presented by multidimensional tables. With this tabular representation, the fact of decision-making process is analyzed according to some points of view related to the company. The fact with its measures can be analyzed according to dimensions with several granularity levels. Requirements are represented as a point in a multidimensional space. This representation is close to decision makers' vision of data.

Table 1. Cost evolutions of Acts during the three last years

Acts.		Time.Year		
Cost		2003	2004	2005
Nature.Family	Nature.Sub_family			
Nurse	Bandage	1 147,43	3 445,14	4 624,12
	Vaccine	3 601,85	5 319,81	7 420,95
Surgical	Aesthetic	115 999,26	69 059,42	173 170,09
	Dental	8 958,61	111 429,62	63 769,32

In addition, users may not have a tabular representation. Hence, we collect their requirements by a natural-pseudo language. With this representation, one can define the facts, dimensions with their measures and parameters respectively. Constraints and restrictions on these elements can also be added using a query as follows:
ANALYZE ACTS
WHEN Costs >500
ACCORDING TO Nature.Family, Nature.Sub_family, Medical_Crew
FOR Time.Year IN (2003, 2004, 2005)

Transforming from a query written in a natural pseudo-language to a tabular representation is easy and involves only user data. The query template's major disadvantage is that it is not user-friendly because restrictions on the elements must be expressed in predicates. Basically, we favor tabular representation in the remainder of this paper. Consider a simple example of a medical company that delivers medical acts and wants to analyze the cost of these actsas presented in table 1.

During the analysis, from multidimensional tables we collect also the requirements related to ETL processes such as historicization, archiving, calculation, consolidation and refreshment. In fact, we consider the only functionality which concerns users (i.e reporting) because the other functionalities (i.e loading and storage) concern source systems. In spite of this, these five processes are the most used through reporting manipulations. In addition, many other interests in DW development were argued [9] and [10]. We use a Decisional Dictionary that we define as an extension of the classic data dictionary with columns dedicated to ETL processes. From row headers and column headers, designers formulate the inputs of the decisional dictionary. To fill line by line the other columns (e.g field type, field constraints and calculation, consolidation, historicization, archiving, and refreshment rules), designers must use cell values. Thus, from the multidimensional table 1, we obtain the Decisional Dictionary sketched in figure 1.

This dictionary provides a general view of tactical requirements about information and processes on these information. But, this view is not close to user vision of data. We will provide a model which is better adapted to decision-maker's vision of data in following sections.

Field Name	Field Description	Field Type	Field Constraints	Calculation Rules	Consolidation Rules	Historicizati on Rules	Archiving Rules	Refreshment Rules
Cost	Cost of medical act realized by date	Double		Volume of an act * unit_price of an act per medical act	Function = sum, average, min, max, stddev, var, count	Duration =3 years	Duration = 10 years Function =sum	Frequency= week Mode =merge
Act_code	Code of medical act	Text	Not null			Duration= 3 years	Duration= 10 years Function= sum	Frequency= year Mode =merge
Sub_family	Sub family of medical act	Text				Duration =3 years	Duration = 10 years Function =sum	Frequency= year Mode =merge
Family	Family of medical act	Text				Duration =3 years	Duration = 10 years Function =sum	Frequency= year Mode =merge
Date_day	Date of medical act	Date	Not null			Duration =3 years	Duration = 10 years Function =sum	Frequency= year Mode =merge
Month	Month of medical act	Date				Duration =3 years	Duration = 10 years Function =sum	Frequency= year Mode =merge
Quarter	Quarter of medical act	Date				Duration =3 years	Duration = 10 years Function =sum	Frequency= year Mode =merge
Semester	Semester of medical act	Date				Duration =3 years	Duration = 10 years Function =sum	Frequency= year Mode =merge
Year	Year of medical act	Date				Duration =3 years	Duration = 10 years Function =sum	Frequency= year Mode =merge

Fig. 1. Decisional Dictionary of tactical requirements

3.2 Formalization of Tactical Requirements

From a sample of representative decision-making analyses and the Decisional Dictionary, the DW designer can formalize the multidimensional tactical requirements with our model called Decisional Diagram. In order to guide this task, we provide specific transformation rules from tactical requirements to our model. We intend to achieve a proposal model with the following properties:

- It is close to the way of thinking of data analyzers,
- It represents information and operations related to these information in early steps of DW design,
- It handles separately information and processes in the same model.

Our model is inspired from the object-oriented multidimensional model of [11] which verifies the principles of the star schema [2]. Facts and dimensions are represented by a class of a stereotype. It takes into account ETL processes. To define these processes, we need to associate a behavior to an attribute.

For this same problem, [12] represents attributes as first class modelling elements. But, the authors argue that: ”an attribute class can contain neither attributes nor methods”. Thereby, it is possible to associate two attribute classes but it is impossible to associate methods to an attribute as we expect. Hence, we propose to add the stereotype ”attribute” to the methods only applied on attributes but not applied on all the fact-class or dimension-class. To define precisely on what attribute the method is applied, the attribute is its first parameter. Our method has UML advantages and it offers models and tools for DW problems presented in the following paragraphs.

The ETL processes are defined at class or attribute levels. For each ETL process, we associate a concept called "informativity concept" which is mentioned at attribute level. Informativity concepts are placed with data visibility. We model informativity concepts and the processes associated as follows:

- h : historicize(p, d, c): historicization process at a period p for a duration d with a constraint c ,
- a: archive(p, d, c, fct): archiving process at a period p with a duration d, a constraint c and an aggregate function fct,
- * : refresh(f, m): refreshment process with a frequency f and a refresh mode m,
- c : calculate($\{v_i\}^+$): calculation with parameters v_i,
- s : consolidate(l): consolidation with the level l of consolidation chosen from [13]'s four levels to get meaningful aggregations.

To transform an expression of tactical requirements into a Decisional Diagram, we describe a three rule-based mechanism. It is composed of some structuring rules, well-formedness rules and merge rules :

- the structuring rules enable designers to organize the project environment into one or several Decisional Diagrams. They also make it possible to model facts and dimensions with their measures and parameters respectively into fact and dimension-classes. Some rules help to define the above-mentioned processes from the Decisional Dictionary,
- the well-formedness rules check whether the schema resulting from the analysis of tactical requirements is well-formed. They make it possible to control schema consistency,
- the merge rules indicate how to merge several Decisional Diagrams according to project environment from object names . They take into account fact and dimension-classes in common.

The complete rule-based mechanism is defined in the technical report [14]. Below, we present its application to the table 1 of our running example. We start by applying structuring rules, more precisely on the environment that lies in all multidimensional tables of tactical requirements.

- Rule EI1: the project environment about tactical requirements is composed of one Decisional Diagram because we have one multidimensional table.

Thus, for each multidimensional table we apply first the structuring rules related to facts and its measures, then we apply dimensions and parameters ones. When we apply structuring rules of facts and measures, we find out:

- Rule SI1: the fact "Acts" is transformed into the fact-class "Acts",
- Rule SI2: the measure "Cost" of fact "Acts" is transformed into the attribute "Cost" of fact-class "Acts",
- Rule SI3: the measure "Cost" is calculated, historicized, refreshed, archived, consolidate because it is calculated from the volume of acts and the unit_price per act, historicized every year for three years, refreshed every week according to the merge operation and archived every year for ten years. Therefore, we add the property of informativity "c/h/*/a/s" to the attribute "Cost" of the fact-class "Acts",

- Rule SP1: the ETL processes of facts and measures are defined from the properties of informativity associated to each attribute. Thus, we define the operations from the columns with the same names in Decisional Dictionary. If the constraints are the same for all the attributes of a fact-class per process, the operation is defined at class level. Otherwise, we define an operation per attribute which has its own constraints. In our running example, all the operations of fact-class "Acts" are at class level, except Calculate operation which is specific to the measure "Cost". Therefore, we define the followings :
  - the operation Calculate(Cost, Volume, Unit_price)<<attribute>> means the attribute "Cost" is calculated with the parameters Volume and unit_price. The computation is done by the end-user group. The operation is at attribute level,
  - the operation Historicize(year, 3, NULL) means the attribute of fact-class "Acts" is historicized for the three previous years (because in the table 3 years is analyzed p=year and d=3) without constraints (c=NULL),
  - the operation Refresh(week, merge) means the attribute of fact-class "Acts" is refreshed every week (f=week) according to the merge operation (m=merge),
  - the operation Archive(year, 10 , NULL, sum) means the attribute of fact-class "Acts" is archived for ten years (p=year and d=3) by summing (fct=sum) without constraints (c=NULL),
  - the operation Consolidate(1) means all the aggregate functions can be applied on the attribute of fact-class "Acts" (l=1).

When we apply structuring rules of dimensions and parameters, we find out:

- Rule AI1: the dimensions "Nature" and "Time" are transformed into dimension-class "Nature" and "Time" respectively,
- Rule AI2: the parameters of dimension "Nature" attributes ("Family" and "Sub_family") are transformed into attributes of dimension-class "Nature". The attributes of "Time" dimension-class are the classic "Year, "Semester", "Quarter", "Month" and "Day_Date",
- Rule AI3: the properties of informativity h/*/a are associated to attributes of dimension-classes "Time" and "Nature" because they are historicized every year for three years, refreshed every year according to the merge operation, archived every year for ten years after historicization with sum aggregate function,
- Rule AP1: the ETL processes of dimensions and parameters are defined according to the same criteria as that of the ETL processes of facts and measures. The operations of dimension-classes "Time" and "Nature" are the same because the constraints related to each process per dimension are the same. Moreover, the operations are at class level because the constraints are the same for each parameter per dimension and process. Hence, we define the following operations :
  - the operation Historicize (year, 3, NULL) means the attributes are historicized for the three previous years without constraint,
  - the operation Archive (year, 10 , NULL, sum) means the attribute of fact-class "Acts" is archived for ten years by summing without constraint,
  - the operation Refresh(year, merge) means the attribute of dimension-classes "Time" and "Nature" are refreshed every year.

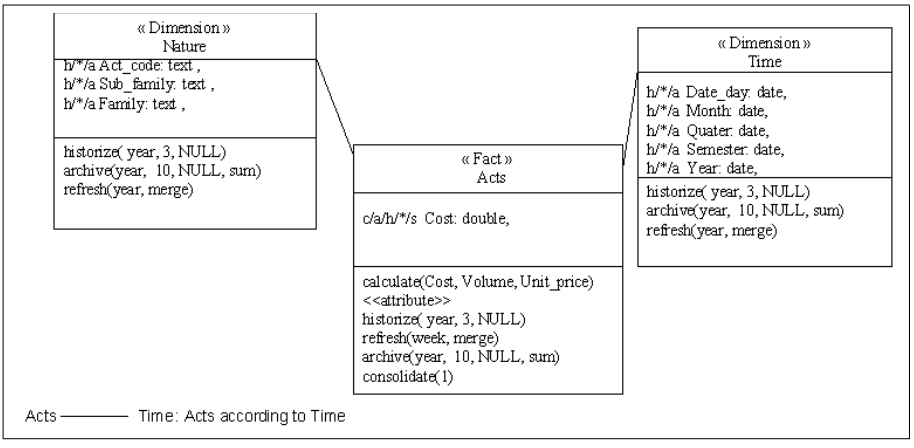

Fig. 2. Decisional Diagram of tactical requirements

We get to the Decisional Diagram as represented in figure 2. This simple diagram is well-formed according to formedness rules. We have only one Decisional Diagram for tactical requirements, therefore we do not need to apply the merge rules.

4 Strategic Requirement Analysis

4.1 Collection of Strategic Requirements

Decision-makers need a synthesis view of the data and their requirement are related to key indicators of enterprise-wide management as shown in table 2. In the context of our contract with I-D6 company which is specialized in decision-making, we notice that strategic indicators composed tables which have only one dimension e.g Time dimension. The indicators are also present in other multidimensional tables expressing tactical requirements. Hence, we consider these tables that we called strategic tables in order to define the kernel of indicators of the future DW. Then, we collect strategic requirements as tactical requirements.

4.2 Formalization of Strategic Requirements

As the strategic requirements are represented through measures which are only depending on the ”Time” dimension important to handle these tables with the structuring rules

Table 2. KPIs multidimensional table

	Time.Month	
	January	February
Cost	235025	355186
Day cost	7568	9135
Average cost per act	752	882

EI2. This rule declare any table as a not suitable table when it is not organized by a dimension in column and eventually a dimension in row and when its cells do not match the fact measures.

Hence, DW designers transform the tables by taking into account that each indicator is not a "secondary measure". We called "secondary measures" the measures which can be calculated from other measures called "main measures". In Decisional Diagrams of strategic requirements, the secondary measures are not formalized in order to insure the consistency of the diagrams and to assess the diagrams of the three types requirements. To formalize strategic requirements, the DW designers also define a Decisional Dictionary. The kernel of Decisional Diagrams can be defined from the multidimensional tables of their requirements and the Decisional Dictionary.

In our running example, we must transform the multidimensional table 2 into a multidimensional table with a dimension "Time" and a fact "Acts". At the beginning, this fact contains three measures. But among these measures, two of them can be calculated from the other. The measure "Day cost" and "Average cost per act" can be calculated from the measure "Cost". Therefore, the multidimensional table is structured with "Time" dimension columns where the fact "Acts" has one measure "Cost". The Decisional Dictionary of strategic requirements contains the same rows of "Time" dimension and Acts fact as Decisional Dictionary of tactical requirements. The kernel of Decisional Diagram is composed of the Decisional Diagram represented in figure 3.

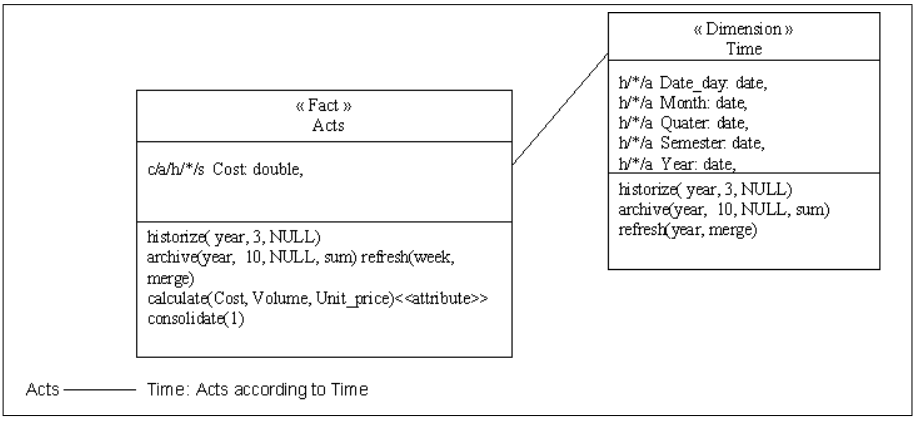

Fig. 3. Decisional Diagram of strategic requirements

In our running example, we have only a strategic table. Thus, we get to the Decisional Diagrams. From the tactical Decisional Diagram and the strategic Decisional Diagram, designer must merge the fact-classes, dimension-classes and attributes present in the two diagrams by using merge rules (presented below). Designers apply these rules with kernel Decisional Diagram as a reference in order to keep its multidimensional objects. The merge rules are :

– FUS1: merge dimension and fact-classes by adding attributes and ETL operations. It makes it possible to gather Decisional Diagrams with the same facts which have common dimension-classes,
– FDS1: merge dimension-classes by adding attributes and ETL operations to define a constellation. It makes it possible to gather Decisional Diagrams with different facts which have common dimension-classes.

Then, after the confrontation of tactical and strategic requirements, designers can assess result of the first confrontation with the result of system requirement analysis. Before the design of the conceptual schema, designers can evaluate whether strategic and tactical requirements can be satisfy since the first iteration of requirement analysis step. If there is inadequacy, this iteration of our DW design method is closed and a new iteration begins in order to enclose the three types of requirements together.

5 Conclusion

This paper presents our model called Decisional Diagram for tactical requirement analysis process. We provide a method to derive a Decisional Diagram form tactical requirements and strategic requirements. The method uses a three rule-based mechanism which is composed of structuring, well-formedness and merge rules. These rules enable data warehouse (DW) designers to get to a Decisional Diagram with respect to tactical and strategic requirements about information and ETL processes.

Our proposal introduces a model between (tactical, strategic) requirements and the conceptual schema to tackle all DW requirements in the early steps of data warehouse design. It has the advantage of enabling DW designers to define ETL operation interfaces. Defining historicization, archiving, calculation, consolidation and refreshment processes during the early step of the DW design may contribute in reducing the important rate of ETL cost and time in a DW project.

As [9], [10] and [12] argue , few researches have been done to develop models and tools for ETL process. Therefore, in the near future we intend to enhance the understandability and user-friendliness of data mappings. These mappings will be useful as a document in DW project validation by its actors at conceptual and logical abstraction levels. Moreover, we are working on the definition of hierarchies during requirements analysis.

References

1. Bruckner, R., List, B., Schiefer, J.: Developping requirements for data warehouse systems with use cases, AMCIS (1999)
2. Kimball, R.: The data warehouse toolkit: practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc., New York, NY, USA (1996)
3. Annoni, E., Ravat, F., Teste, O., Zurfluh, G.: Les systèmes d'informations décisionnels : une approche d'analyse et de conception à base de patrons. revue RSTI srie ISI, Méthodes Avancées de Développement des SI **10**(6) (2005)

4. Golfarelli, M., Rizzi, S.: Methodological framework for data warehouse design. In: DOLAP '98, ACM First International Workshop on Data Warehousing and OLAP, November 7, 1998, Bethesda, Maryland, USA, Proceedings, ACM (1998) 3–9
5. Luján-Mora, S., Trujillo, J.: A comprehensive method for data warehouse design. In: DMDW. (2003)
6. Abelló, A., Samos, J., Saltor, F.: Yam2 (yet another multidimensional model): An extension of uml. In Nascimento, M.A., Özsu, M.T., Zaïane, O.R., eds.: IDEAS, IEEE Computer Society (2002) 172–181
7. Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., Paraboschi, S.: Designing data marts for data warehouses. ACM Trans. Softw. Eng. Methodol. **10**(4) (2001) 452–483
8. Giorgini, P., Rizzi, S., Garzetti, M.: Goal-oriented requirement analysis for data warehouse design. In Song, I.Y., Trujillo, J., eds.: DOLAP, ACM (2005) 47–56
9. Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Conceptual modeling for etl processes. In Theodoratos, D., ed.: DOLAP, ACM (2002) 14–21
10. Bouzeghoub, M., Fabret, F., Matulovic-Broqué, M.: Modeling the data warehouse refreshment process as a workflow application. In Gatziu, S., Jeusfeld, M.A., Staudt, M., Vassiliou, Y., eds.: DMDW. Volume 19 of CEUR Workshop Proceedings., CEUR-WS.org (1999) 6
11. Luján-Mora, S., Trujillo, J., Song, I.Y.: Extending the uml for multidimensional modeling. In Jézéquel, J.M., Hußmann, H., Cook, S., eds.: UML. Volume 2460 of Lecture Notes in Computer Science., Springer (2002) 290–304
12. Luján-Mora, S., Vassiliadis, P., Trujillo, J.: Data mapping diagrams for data warehouse design with uml. In Atzeni, P., Chu, W.W., Lu, H., Zhou, S., Ling, T.W., eds.: ER. Volume 3288 of Lecture Notes in Computer Science., Springer (2004) 191–204
13. Pedersen, T.B., Jensen, C.S.: Multidimensional data modeling for complex data. In: ICDE, IEEE Computer Society (1999) 336–345
14. Annoni, E.: ebipad : Outil de developpement des systemes d'information decisionnels. Technical Report IRIT/RR-2006-12-FR (2006)