Data Processing Modeling in Decision Support Systems

Concepción M. Gascueña[1] and Rafael Guadalupe[2]
[1]*Department of Computing, Polytechnic of Madrid University, Carretera de Valencia Km7, 28031 Madrid, Spain*
[2]*Department of Topographic, Polytechnic of Madrid University, Carretera de Valencia Km7, 28031 Madrid, Spain*

Keywords: Multidimensional Models, Data Processing in Multidimensional Databases, Data Processing in Data Warehouses, Data Processing in Decision Support Systems, Virtual factEntity.

Abstract: Due to the advancement of both, information technology in general, and databases in particular; data storage devices are becoming cheaper and data processing speed is increasing. As result of this, organizations tend to store large volumes of data holding great potential information. Decision Support Systems, DSS try to use the stored data to obtain valuable information for organizations. In this paper, we use both data models and use cases to represent the functionality of data processing in DSS following Software Engineering processes. We propose a methodology to develop DSS in the Analysis phase, respective of data processing modeling. We have used, as a starting point, a data model adapted to the semantics involved in multidimensional databases or data warehouses, DW. Also, we have taken an algorithm that provides us with all the possible ways to automatically cross check multidimensional model data. Using the aforementioned, we propose diagrams and descriptions of use cases, which can be considered as *patterns* representing the DSS functionality, in regard to DW data processing, DW on which DSS are based. We highlight the reusability and automation benefits that this can be achieved, and we think this study can serve as *a guide* in the development of DSS.

1 INTRODUCTION

One of the challenges of Software Engineering (SE), is to propose: rules, process, guidelines and models that address Software development: quickly, efficiently, in a specific and unambiguous manner and resulting in a quality product. Methodologies are proposed continually, with varying degrees of complexity and agility; leading teams in a certain direction during the software development process, also referred to as software life cycle. In recent years, SE has acquired great importance and, increasingly, less software developments that being undertaken without prior planning. In SE the Cases of use (CU), are considered by most members of the scientific community as a technique, not necessarily object-oriented, which allows us to model the functionality of a software system at a high level of abstraction, and with no regard to the programming paradigm in which the system will be implemented.

Decision Support Systems DSS, are based upon historical databases containing large amounts of data. They try to extract the information processing the data in a certain way; allowing managers to make decisions and predict future trends. "*Predicting the future by studying the past*." However, DSS are not always based on databases built for this purpose, sometimes using transactional databases, something we don't consider efficient. We believe the DSS must be based on data warehouses (DW), or multidimensional databases (MMDB); and following specific, multidimensional (MM), data models; which reflect the multidimensional semantics and lead to analysis from the earliest stages of system development. In this work we are using MM and CU for modeling processing data in DSS.

This paper is structured as follows: Section 2 includes a study on related works in MMDB and on the representation of functionality in the development of Software Systems. In Section 3, we present our proposal. Section 4 includes an example using our proposal. In section 5, some conclusions and future work are offered.

2 RELATED WORKS

Most DSS development proposals are mainly concerned with the database on which they are built

M. Gascueña C. and Guadalupe R..
Data Processing Modeling in Decision Support Systems.
DOI: 10.5220/0003982601330138
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 133-138
ISBN: 978-989-8565-10-5

upon, (Kimball, 1996), (Imon, 2002), (Mazón, 2006). To develop this DB, data models have been shown, as in (Tryfona, 2003), (Torlone, 2003), (Malinowski, 2004), (Luján-Mora, 2006), (Gascueña, 2006). There are authors that propose using transactional database models, as (Malinowski, 2004), (Tryfona, 2003), however other authors propose using specific models that treat the semantic MM in a specific manner, as (Kimball, 1996), (Torlone, 2003), (Gascueña, 2008c). In recent years, the importance given to MM models has increased, and there are even some proposals that try to represent spatial-temporal data behavior within them, as in (Malinowski, 2005), (Parent, 2006), (Gascueña, 2008a), (Bimonte, 2008). This leads us to stress the value that the scientific community is giving to MM models used in the development of the DW or MMDB. Regarding the processing of data, there are some works as in (Gascueña, 2008b), where an analysis is performed, while separating the concepts of basic data and derived data. They use models to represent both data types, and they propose an algorithm responsible for the automatic gathering of the data derived from the DW. However there are few proposals regarding the data processing functionalities of DSS.

The CU is the most widely employed technique to model Software systems functionalities. However, these are almost always used in a particular way for each system; they are "tailored" by the applications that they model. We think it would be desirable to propose CU "*patterns*" that could be reused by most systems that need the same functionalities. There are some initiatives that tackle generalized problems, such as in (Guttorm, 2005) who proposes using CU to represent the supposed potential threats that a system could face, modeling both the functionality and threats of systems, They name these, cases of bad use, *misuse cases*. In (Kantorowitz, 2003) a framework is proposed, oriented on CU, to build, automatically, graphical user interfaces (GUI). They also attempt to reuse these CU in different applications. In (Luján-Mora, 2006) the MM semantics are specified using class diagrams and they propose new artifacts aimed at collecting such semantics. They include an example of how to specify two data requirements by two CU. But the proposed CU, are entirely dependent upon the discussed requirements. In this paper we propose a general reusable CU, a “*pattern*”, which may be used as a guide in the development of DSS to the end of modeling the data processing functionality.

3 PROPOSAL

We are framing this paper within the Software Engineering into the Analysis Phase of software life cycle. We will use data models and CU to propose a guide for development of DSS; proposing, *on one hand,* appropriate conceptual MM data models that reflect the basic starting data required to develop a DW. And *on the other hand*, we will use CU to represent the functionality of any DSS, regarding data processing, and that will allows us to obtain, dynamically and automatically derived data. The MM data models used in this study were shown in (Gascueña, 2006) and completed in (Gascueña, 2008a). To obtain dynamically derived data, we have used the algorithm presented in (Gascueña, 2008b).

3.1 Data Models

In this section we offer a brief introduction of conceptual MM model named FactEntity (FE), to better understand our proposal.

The MM models should represent the data focused to analysis at the earliest stages of the DSS development. They try to represent a *fact* object of study, from different perspectives or *dimensions* and with different *levels* of detail or granularities. *Levels* are obtained by grouping *basic data* from different criteria. With different criterion are formed different *hierarchies.* A *hierarchy* contains a set of levels grouped according to a criterion. A *dimension* can have multiple hierarchies. A *fact* consists of a set of *fact measurements*.

The *FE* model distinguishes between *basic data* (existing data) and data obtained by processing the basic data according to the analysis criteria, also called *derived data*. *Facts* and *dimensions* are combined to obtain the named *factEntities*. The *factEntities* can be *basic* and *virtual*. The *Basic factEntities BfE*, are obtained through the dimensional levels of minimum granularity (leaf levels) and *basic fact measures*. The named *Virtual factEntities VfE*, are obtained through the processing of basic data. The rules by which each factEntity contains a single level of each dimension and a set of fact measures are complied with. Though sometimes this set could be empty. In figure 1, we see the constructors, elements, relationships and functions used by the FE model, representing the MM semantics.

Hierarchies are classified according to the involvement their “*path Rollup”* (moving from a lower to a higher level) has over fact measures. Next

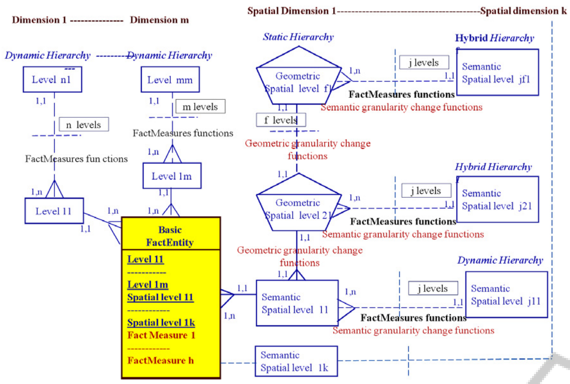

Figure 1: Basic FE model completed with the functions that will apply on fact measures when the Rollup is run.

we see these:

- *Dynamic hierarchy* (its route involves changes in fact measures).
- *Static hierarchy* (its route does not involve changes in the fact measures).
- *Hybrid hierarchy* (is a mixture of the two previous types).

As we show in Figure 1, the Static and Hybrid hierarchies represent spatial characteristics. We see that the BfE counts with representatives of the dimensional leaf levels and fact measures. Also, the diagram represents both, the functions to be applied to achieve higher levels in the hierarchy (this is of specially interest in changing spatial granularities), and the analysis functions to be applied on fact measures, once the rollup between the dimensional levels has been performed (this is necessary as to perform basic data processing and obtaining derived data).

3.2 Cases of Use

In this proposal we present a generic CU model aimed at picking up DSS functionalities in regard to the processing of basic data. This intends to be a *guide* for developers and analysts of these systems.

3.2.1 CU Diagram

In Figure 2 we can see the *To Generate Virtual factEntities* diagram, which represents a main CU named *Generate Virtual factEntity* VfE_CU, and four associated CU: Create Table, Create Materialized View, Create View, Other. All of them count with the <<extend>> label. This provides the functionality the ability to store the VfE both, inside and outside the DW, and also in various, different, ways, leaving the final choice up to the user (analyst manager).

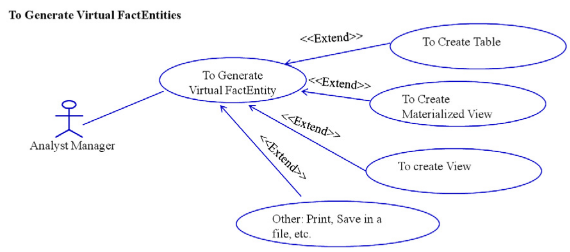

Figure 2: CU diagram: *To Generate Virtual FactEntities*. This shows how the VfE storage could be chosen in the analysis.

3.2.2 CU Description

In Tables 1, 2 and 3 we can observe the VfE_CU description. This is a generic CU that defines the minimum functionality required in any DSS, needed for the processing and gathering of derived data, from a DW. To develop this CU we have used the data model of Figures 1. Table 1 contains the principal scenario or *typical course of events,* functionalities. The head of this CU has been omitted since it is not relevant for this work.

The VfE_CU performs the following tasks: *First*, the user requests to generate VfE, the system asks for the order in which dimensions will be crossed to get all possible VfE. *Second*, the system calls the Gascueña algorithm, which in turn obtains all the possible forms of data crossing between dimensions and fact measures. *Third*, the system presents the user with a listing of the obtained VfE. *Fourth*, the user chooses to generate a certain VfE (this action

Table 1: Events typical course of Generate VfE_CU.

Events Typical Course	
USER	**SYSTEM**
1.- Requests to Generate Virtual FEs	
	2. Asks dimensions in the desired order
3. Chooses ordered dimensions by analysis to perform	
	4. Generates VfE listing applying the algorithm Gascueña
	5. Shows list of the VfE generated
6. Chooses the fact measures and the VfE by to run	
	7. Shows a listing with different options to store data: table, view, materialized view, others.
8. Chooses option to store in a table of DW.	
	9. Calls To Create Table CU
	10. Asks conformity to save data in the DW
11. Agrees	
	12.Ask if he wish to run another VfE from those obtained in step 5.
13. Chooses option: No	
	14.Ask if he wish to generate other VfE listings changing the dimensions order.
15. Chooses option: Not continue	
	16. Close option Generate VfE.

obtains and process data of basic DW, respective of the VfE structure chosen). *Fifth*, the system obtains and presents the data and requests an option towards the data treatment, as it is shown in the diagram in figure 2. *Sixth*, the system allows the execution of as many VfE as needed by the user. The system will also allow obtaining other VfE listings, taking dimensions in different order, and as many times as the user wants. All this is explained in detail in Table 1, which has 16 steps. In Tables 2 and 3 we observe some alternatives, which we have considered more important, to VfE_CU's typical course.

Table 2: Events alternative courses contemplate various options for storing structures and data of VfE.

Alternative Curse 2)	Alternative Curse of step 8 of Events typical Curse
USER	SYSTEM
	7. Shows a listing with different options to store data: table, view, materialized view, others.
8. Chooses option to store data by materialized view.	
	9. Calls To Create Materialized View CU
	10. To return to step 10 in Events typical course
Alternative Curse 3)	Alternative Curse of step 8 of Alternative Curse 2
USER	SYSTEM
	7. Shows a listing with different options to store data: table, view, materialized view, others.
8. Chooses option to store data by view.	
	9. Calls To Create View CU
	10. To return to step 10 in Events typical course
Alternative Curse 4)	Alternative Curse of step 8 of Alternative Curse 3
USER	SYSTEM
	7. Shows a listing with different options to store data: table, view, materialized view, others.
8. Chooses option other (not store data in DW): to print, to store data in a file, etc.	
	9. Calls Other CU
	10. To return to step 10 in Events typical course

Table 2 describes alternatives to the so called "Create table CU", (step 8 of events typical course). There are various options: Create materialized views CU, Create views CU and Others CU. Table 3 describes alternatives to run additional VfE (option: Yes, step 13 of the typical course of events); and alternatives to obtain new lists of VfE, choosing dimensions in different orders (option: Yes, step 15 of events typical course). Both, the typical course as alternative courses may contain more options, but here, they have not been considered since they do not bring greater value into our discussion.

Table 3: Events alternative courses that show the ability to implement different VfE; and the ability to obtain new lists of VfE choosing dimensions in different orders.

Alternative Curse 5)	Alternative Curse of step 13 of Events typical Curse
USER	SYSTEM
	12.Ask if he wish to run another VfE from those obtained in step 5.
13. Chooses option: Yes	
	14. To return to step 5 in Events typical course
Alternative Curse 6)	Alternative Curse of step 15 of Events typical Curse
USER	SYSTEM
	14.Ask if he wish to generate other VfE listings changing the dimensions order.
15. Chooses option: Yes	
	16. To return to step 3 in Events typical course

3.2.3 Gascueña Algorithm

Let's briefly define the Gascueña algorithm, for further details please refer to (Gascueña, 2008c). We describe it in three stages.

First: Given a set of *n* dimensions, we obtain all possible combinations, in groups of 1, 2,...,*n-1* and *n* dimensions. We apply the follow formula (1):

$$[D_i,...,D_p]/\forall i\in[1,..,n] \wedge \forall p\in [i+1,...,n] \wedge (p> i \text{ OR } p= \emptyset). \quad (1)$$

Second: The *Cartesian product* is applied on each of the previous subgroups, taking into account that in some application domains, the order in which we choose the elements to make up the subgroup will be significant.

Third: The Virtual factEntities are obtained by adding to the Cartesian subgroups obtained in the previous step the respective fact measures. We then apply the following formula (2):

$$VfE=([D_i X\ldots XD_p],\{G_j(me_j)\})\text{-}(BfE). \quad (2)$$

Where: $(D_i X\ldots XD_p)$ represent the *Cartesian Product.* And $(G_j(me_j)$ is the set of compatible functions G_j with the basic fact measure (me_j). It excludes the Basic fE).

4 APLICATIONS

Next we will develop a practical example in which we will apply our proposal.

We consider it desirable to study the damage caused by insect plagues in agriculture of certain Earth zones over time. The spatial area is divided into plots, and these are grouped into cities. It is necessary to store the % of extension of each plague on each plot in a given and determined moment of time. The plagues are exterminated, or attempted to, through the use of different technologies. The study requires storing existing technologies and effectiveness of such in the treatment of infected plots. The effectiveness is measured by the % of deaths caused by the treatment. The evolution of plagues on each plot is checked weekly. The spatial areas will be represented by spatial data with *geometric shapes, such as: surfaces, lines and points that can be indistinctly used. The % extension of plague and % deaths will be studied from different perspectives and details*: *Time:* week, year; *Zones:* plot, city; *Technical:* technical type; *Plague:* plague type, family and order.

To offer a solution to this study we propose building a DSS, which allows us to analyze the effectiveness of anti plague treatments, and aid us in choosing the best decisions regarding the treatment of new emerging plagues. The DSS will consist of a MMDB or DW complete with spatial treatment. Furthermore, the system allows the data processing of DW on demand, in an easy and quick manner. Figure 3 shows the proposed *FE Basic model* as a solution for the storing of the input data.

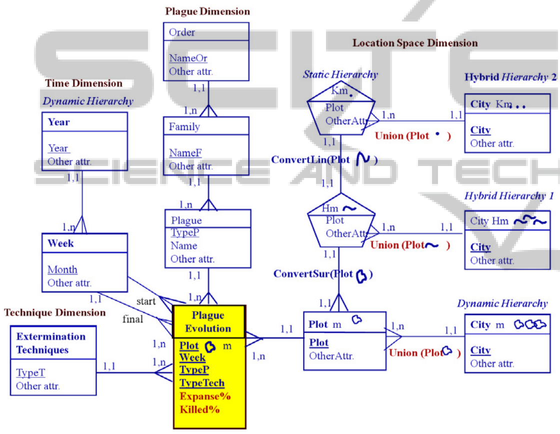

Figure 3: Basic FE model for Plagues Study.

We have identified the following dimensions: Time, Plague, Technique and Location Space. The *Time* dimension has two granularities: week, year. The *Plague* dimension has three granularities: type plague, family and order. The *Location Space* dimension has two semantic granularities: plot and city; and three geometric granularities (spatial representation): surface, line and point. Also this dimension form a *dynamic hierarchy*, a *static hierarchy* and three *hybrid hierarchies*. The “Plague Evolution” *basic factEntity* contains the primary keys inherited from the leaf level of the dimensions (underlined in the diagram). The week level has two relationships (start, final) with BfE. The *fact* under consideration contains two *fact measures*: Expanse% and Killed%. In the diagram, we can also observe the functions used to create higher levels, of both the *geometric* and *semantic* granularities, within the spatial dimension. In figure 4, we observe how the Basic FE model is completed with information regarding the functions to be used for the analysis, once the Rollup is made.

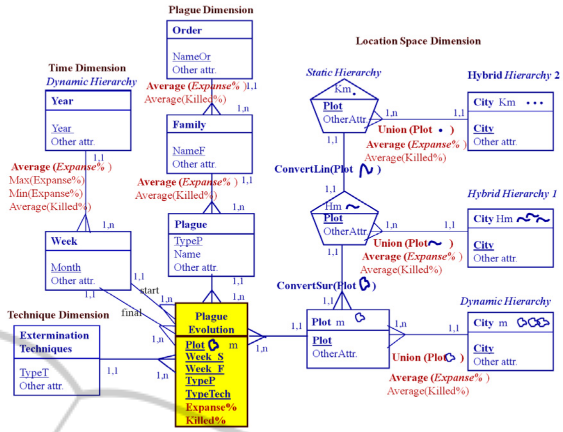

Figure 4: FE conceptual multidimensional model, prepared for processing data by “Plagues Study DSS”.

Now and here we could have included the CU models presented in Figure 2 and tables 1, 2 and 3, adapted to our example. But, if we study these models in detail, we note that it is necessary to include anything new in the descriptions and diagram of the VfE_CU. We observe that the CU model proposed is valid to represent the required minimum functionality required to process the derived data in this example.

5 CONCLUSIONS AND FUTURE RESEARCH

In this paper we have proposed a methodology, which attempts to serve as a *generalized guide* for the development of DSS following the Software Engineering guidelines. Our proposal is framed within the Analysis phase of the software development process life cycle. We have used MM data models and CU to lead the development. *On the one hand*, we offer the foundations to build a DB that collects MM semantics (to create the DW, main part of DSS). *On the other hand*, we model the data processing, defining the desired functionality through a CU model. We explain our proposal in three steps. *First*, we propose carrying out a conceptual multidimensional data model with the adequate structure required to store the basic or starting data in a DW. The model takes into account the analysis requirements. *Second*, the basic data model obtained in the previous step is completed with the operations and functions that we would want to use in the data analysis. This new model presents all the necessary elements needed for the processing of the data, allowing us to obtain new data structures for the derived data. *Third,* data functionality processing is modeled by a CU. In particular, it is defined and developed the Virtual factEntity CU. The VfE_CU details the minimum and necessary events sequence required for the basic data processing. These VfE_CU use an algorithm that interacts with data models, collecting the information represented in them, to generate, automatically and on-demand, all the possible VfE. The steps above outlined, can be considered to have a high level of abstraction and are independent of its implementation. We believe that the proposed CU can serve as a basic *pattern* in the development of DSS; which later may be completed and adapted to each particular situation, if necessary. Finally, we have presented an example in which we develop a case study using our own proposal.

Our future research is aimed at discovering other general behavioral patterns, which could guide the development of the DSS. In addition, we are interested in developing a tool that would allow us to describe and transform, automatically, the FE data models and the VfE_CU, into real systems. The FE model transformation will be made to implement the models in commercial DB manager Systems, under different paradigms: Relational, Object Relational or Object Oriented. The VfE_CU transformation will allow us to implement a basic interface, with the features described in this proposal, while also allowing for the possibility to choose programming languages among the most popular ones.

REFERENCES

Bimonte S., Tchounikine A., Berloto M., 2008. Integration of Geographic Information into Multidimensional Models. *ICCSA 2008*: International, 2008.

Gascueña C. M., Cuadra D., Martínez P., 2006. A Multidimensional Approach to the Representation of the Spatiotemporal Multigranularity. *ICEIS 2006*.

Gascueña C. M., Guadalupe R., 2008a. Some Types of Spatio-Temporal Granularities in a Conceptual Multidimensional Model. *7th International Conference, APLIMAT* Bratislava, Slovak.

Gascueña C. 2008b. Propousal of a Conceptual Model for the Representation of Spatio Temporal Multigranularity in Multidimensional Databases. *PhD Thesis. Polytechic University of Madrid*, Spain.

Gascueña C. M., Guadalupe R., 2008c. A Study of the Spatial Representation in Multidimensional Models, *ICEIS 2008*.

Guttorm Sindre, E Andreas L. Opdahl, 2005. Eliciting security requirements with misuse cases, in *the Journal of Requirements Eng*, Issue 10, pp 34–44.

Inmon, W. 2002. Building The Data Warehouse. *Jhon Wiley & Sons*.

Kantorowitz E., Lyakas A., Myasqobsky A.. 2003. A Use Case-Oriented User Interface Framework. *Software. SwSTE '03. IEEE International Conference on.*

Kimball R. 1996. The Data Warehouse Toolkit. *John Wiley&Sons Ed.*

Luján-Mora S., Trujillo J., Song Il- Yeol. 2006. A UML profile for multidimensional modeling in data warehouses. *DKE, 59(3), p. 725–769*.

Malinowski, E. and Zimanyi, E., 2004. Representing Spatiality in a Conceptual Multidimensional Model. *Proc. of the 12th annual ACM international workshop on GIS. Washington, DC, USA*.

Malinowski E., Zimanyi E., 2005. Spatial Hierarchies and Topological Relationships in the Spatial MultiDimER model. *Lecture Notes in Computer Science, page 17, Volume 356*.

Mazón J. N., Pardillo J., Meliá S. y Trujillo J., 2006. Modelado Multidimensional de almacenes de datos con MDA. XI *JISBD 2006*.

Parent C., Spaccapietra S., Zimanyi E., 2006. The MurMur project: Modeling and querying multi-representation spatio-temporal databases. *Information Systems, Volume 31, Issue 8*, Pages 733-769.

Torlone R., 2003. Conceptual Multidimensional Models. In Multidimensional databases: problems and solutions, pages 69-90, *Idea Group Publishing, Hershey, PA, USA*.

Tryfona, N., Price, R., Jensen, C. S., 2003. Conceptual Models for Spatio-temporal Applications. In *M. Koubarakis et al. (Eds.), Spatio-Temporal DB: The CHOROCHRONOS pg. 79-11. Berlin, Heidelberg*.