An Empirical Evaluation of a Historical Data Warehouse

Carlos Neil, Marcelo De Vincenzi, Claudia Pons

Facultad de Tecnología Informática
Universidad Abierta Interamericana
Buenos Aires, Argentina
{carlos.neil,medevincenzi,claudia.pons}@uai.edu.ar

Abstract. Computing is widely regarded as a scientific discipline that emphasizes on three different perspectives: mathematics, present in the development of formalisms, theories and algorithms; engineering, linked to the goal of making things better, faster, smaller, cheaper and, finally, the science that can be defined as the activity to develop general and predictive theories that allow these theories to be evaluated and tested. However, research in software engineering rarely describes explicitly its research paradigms and standards to assess the quality of its results. Due to a growing understanding in the computer science community that empirical studies are needed to improve processes, methods and tools for the development and maintenance of software, an emerging area in software engineering is developed: the Empirical Software Engineering. This subarea is one step down in the claims of scientificity but it aims to address this shortcoming. The objective of this work is to conduct an empirical corroboration for developing a method of a Historical Data Warehouse, the temporal data model and the associated query interface.

Keywords: Empirical Software Engineering, Historical Data Warehouse, Design Method, Graphical User Interface.

1 Introduction

Computing is widely regarded as a scientific discipline that emphasizes on three different perspectives: mathematics, present in the development of formalisms, theories and algorithms; engineering, linked to the goal (inherent to any branch of engineering) to do things better, faster, smaller, cheaper and, finally, the science, which can be defined as the activity to develop general and predictive theories that can describe and explain observed phenomena and where these theories are also evaluated and tested (1). The most established scientific disciplines have elaborated explanations about their research strategies. In those disciplines, notions of falsification, theories, laws, paradigms and research programs are present. However, research in software engineering rarely describes explicitly its research paradigms and standards to assess the quality of its results (2). In particular, Snodgrass (3) showed that, although the research in the field of databases presents significant developments in engineering and mathematics, the scientific perspective has been poorly developed. In the same paper, he notes that in over 10,000 papers on database, studied between 1975 and 2000, only 37 of them mentioned the phrase ¨hypothesis testing¨, and of those, fewer than a dozen actually applied that method.

Software Engineering is a field without much historical background since it is less than four decades old. This idea can be supported by means of the fact that its first publications and conferences were held in the late 1960s (in the 1950s there was some research but it was confusing and without any significant publication) and its academic presence did not begin to separate from Computer Science until the early 1980s (4).

Due to a growing understanding in the computer science community that empirical studies are needed to improve processes, methods and tools for software development and maintenance (5), an emerging area in software engineering is developed: the Empirical Software Engineering. This subarea is one step down in the claims of scientificity but it aims to address this shortcoming. This discipline focuses on experiments in software systems (products, processes and resources) and proposes a rigorous experimental approach to them. In particular, this branch of software engineering is concerned with the design of experiments, data collection and the development of laws and theories from them.

As previously stated (the need to assess proposed developments in software engineering), the following research aims to perform an empirical corroboration of the design method of the Historical Data Warehouse, the associated temporal multidimensional model and the query interface considering their impact on the environment as close as possible to the one that will be used in actual practice.

To summarize the work that was done on empirical corroboration, we highlight: the creation of a new simplified temporal data model for detecting the variation of the values of attributes, entities and relationships that change in time. Then we, using this model, designed a new data storage structure that combines and integrates a Data Warehouse with the Historical Data Base into a single model. Furthermore, from the proposed models, we created a design method that, starting with a conceptual data model and by successive transformations, allowed us to obtain a logical representation of the Historical Data Warehouse in a relational data model. Then we implemented it by using the Model Driven Development.

On the other hand, linked to the information retrieval by the end user, we present a graphical environment automatically derived from the Historical Data Warehouse using the Model Driven Development, for queries on the temporal data structure that allows the automatic generation of sentences SQL for both typical queries of a Data Warehouse and queries for a Historical Data Base.

Finally, we created a prototype based on Eclipse technology, which implements the design method of the Historical Data Warehouse, graphical query interface and the creation of SQL statements. In Figure 1 we show a Graphical User Interface screen.

In Section 2, we discuss general concepts of quantitative research. In section 3, we detail the objects of study of the evaluation. In Section 4, we describe the research work. In Section 5, we present the development of the research work. In Section 6, we show the data from the questionnaires. In Section 7, we perform the analysis of the responses. And finally, in Section 8, we present the conclusion of the work.

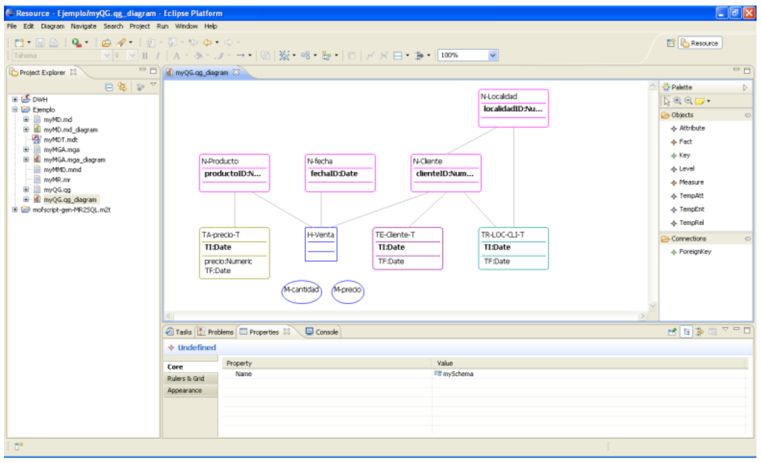

Fig. 1. Graphical interface prototype screen

2 Qualitative Research

While research methods can be classified in various ways, a widely accepted classification is the one that divides them into quantitative and qualitative. The former is especially convenient to the study of natural phenomena or objects. Furthermore, the study of cultural and social phenomena requires other methods, which are not based on formal experiments or theories, but on interviews, questionnaires, documents and reactions observed by the researcher. This group associated with cultural and social phenomena falls within the scope of qualitative methods (6). The qualitative approach is inductive and has the distinction of not being linear but iterative and recurrent; the alleged stages are further actions into the research problem, where the task of data collection and analysis is permanent (7). Following this line of argument, the research focused on the construction of new objects (i. e. processes, models, methods, techniques, etc.) is engineering in nature, in the sense that the object of study is the construction of new tools (i. e. methods, models, etc.) for software construction. This type of research is related to the implementation and use of these new products. The problems in this field require the study of social and cultural factors and seek to answer questions such as: what are the factors that a given software process are not accepted in the company? Why a software development tool is more or less accepted than another? Such problems cannot be addressed solely by the traditionally called "scientific methods", i.e. by purely quantitative methods; they are problems that must be addressed by qualitative methods (8).

2.1 Controlled Experiments

The type of qualitative research can be developed through a controlled experiment, where one or more independent variables of a testable hypothesis are manipulated to measure its effect on one or more dependent variables. The controlled experiment is to determine in precise terms how the variables are related and, specifically, whether there is a causal relationship between them. Each combination of values of the independent variable is considered a different field of study. The simplest way to perform a controlled experiment is represented merely by two independently variable levels (for example, the use of a tool vs. the non use of a tool) (9).

Controlled experiments in software engineering often involve students meeting paper and pencil tasks, the main reason is that they are more accessible, easier to organize and generally without cost. However, this approach is often criticized for lack of realism and therefore, researchers propose that the experiments should be performed on real tasks on real systems by professionals using its technology development in their normal working environments (10).

2.2 Techniques of Data Collection

For qualitative analysis, as well as for the quantitative analysis, data collection is essential, but its objective is not to measure variables to make inferences or statistical analysis. What is sought in a qualitative study is to obtain data from people, contexts or situations that will become information from its critical analysis. Being human beings, the data of interest are concepts, perceptions, beliefs, emotions, etc. (7). The interviews and questionnaires, both data collection tools are focused on questions about specific issues. In the questionnaires, the questions can be of two types: open or closed. The first does not define in advance the response alternatives, serves to deepen an opinion or reasons for behavior, and requires more processing time. The closed-type questions contain answers categories or alternatives that have been defined, they are easier to code and prepare for analysis but limit the sample answers and do not accurately describe what people have in their mind. An interview, on the other hand, is to obtain information through a conversation of a professional nature. In the interview, the assessment tool is the questionnaire, but with the difference that it is the observer who makes the questions and these are not always fixed (11). In the qualitative approach, as there is not so much interest in the generalization of the results, non-probability or directed samples are a suitable option in the choice of the study group (7).

3 Objects of Empirical Study

The empirical evaluation was performed on three objects of study: the storage structure called Historical Data Warehouse, the design of a Graphical User Interface, derived from a Historical Data Warehouse and finally, the design method that encompasses both proposals under the Model Driven Software Development. Then we make a short description of each of them (for more information, see (12))

3.1 Historical Data Warehouse

A Data Warehouse is a "subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management's decisions" (13). A distinctive characteristic of the Data Warehouse is that time is one of the dimensions for the analysis (14, 15) but this refers to the moment when a transaction was made, therefore, it does not specify how or when the values of the entities, attributes and relationships associated with these transactions have varied through time. Although the Temporal Data Warehouse considers, besides the temporal dimension, other aspects related to time (16, 17, 18), this model considers only the changes that occur in the Data Warehouse schema, both in dimensions and in hierarchies. Therefore, a problem to be solved in this type of multidimensional structure, considering the need for registering values that allow to evaluate trends, variations, maximum and minimum, is how we shape the values of entities, attributes or relationships that may vary in time in the design of the multidimensional structure; since, although the needed data of information was stored, the temporal search mechanisms would be complex (19).

The Historical Data Warehouse is a new structure of data storage that combines and integrates a Data Warehouse and a Historical Data Base in a single model. This model includes, besides the main analytical fact, temporal structures related to the levels of dimensional hierarchies that allow to record data and retrieve information that varies in time. The conceptual model of the Historical Data Warehouse is composed of a *fact* and a set of dimensions; the latter are represented by simple or multiple hierarchical levels (temporal and not temporal) (12).

3.2 Queries on a Historical Data Warehouse

A distinguishing characteristic of the On-Line Analytical Processing tools is its ability for grouping the measures by means of the election of one or more dimensions; the multidimensional visualization allows to see the different levels of the hierarchies in each dimension in a logical way, allowing the user to get a more intuitive understanding of the data. The existing commercial software provides different views of the multidimensional elements. In order to facilitate the queries, a graphical language would have to operate on an explicit graphical view of the conceptual multidimensional scheme; in the same way, the On-Line Analytical Processing queries would have to be expressed using a graphical representation in an incremental form (20). It is a challenge for non-technical users to specify queries on a complex database structure (21). Due to the complexity in the formulation of non-trivial searches various approaches have been proposed to make them more accessible to a wider range of users (22). Queries become even more complex when the data is stored in a Data Warehouse historical structure (23). The objective of research in this area consists in simplifying the information search mechanisms on the Historical Data Warehouse; the strategy consists in making use of a graphical environment without considering implementation details.

The proposed Query Graphics allows the Historical Data Warehouse users to formulate queries on the storage structure which, subsequently, will be automatically translated to SQL statements. The Query Graphics aims to simplify the visualization of the Historical Data Warehouse, so that the users can perform their queries automatically, just by putting marks in the graph and supplying actual values to the parameters. The Query Graphic is derived from the Historical Data Warehouse and the set of possible queries is established by identifying common query patterns. The Query Graphic consists of a root node that represents the main fact and a set of connected nodes, each of which represents different levels of hierarchies, measures, temporal attributes, temporal entities and temporal interrelation (24).

3.3 Design Method for a Historical Data Warehouse

Several methods have been proposed that allow to derive the conceptual multidimensional schema from data sources within the organization and/or from the user's requirements (see (15, 25)), most of these methods must be done manually (26). A solution to this problem is proposed by Model Driven Software Development, this approach has become a new paradigm of software development that promises improvements in the software construction based on a model-driven process and supported by powerful tools. In the Model Driven Software Development, the software construction is done through a model-driven process and supported by powerful tools that generate code from them. This new paradigm aims to improve productivity and software quality generated by reducing the semantic leap between the problem domain and the solution (27).

The proposed design method consists in successive automatic and semiautomatic transformations of models, that begin with an ER model and that finally allow to obtain a temporal multidimensional model expressed as a set of tables in the relational model. The complete proposal includes the automatic creation of a Graphic Query interface derived from the Historical Data Warehouse that by means of marks on a graph automatically creates temporal and decision-making SQL query statements (24).

The work is complemented with the creation of a prototype based on ECLIPSE technology, which implements the design method of Historical Data Warehouse, the Graphic Query interface and the automatic execution of SQL statements (28).

4 Research Work

The empirical research conducted to assess the proposal of the Historical Data Warehouse, the design method and the associated graphical query interface, was qualitative and was developed through a controlled experiment, through questionnaires with open-type questions. The research was conducted with students from the Master of Information Technology of the school of Information Technology of the Universidad Abierta Interamericana. The objectives in this research can be summarized in the following items: assessment design method of the Historical Data Warehouse, the storage structure and the Graphical User Interface.

4.1 Hypotheses

The proposed design, the methods, and the Graphical User Interface initially imply three hypotheses which, although not explicitly raised (12), can be considered as such and, therefore, will be corroborated empirically, namely: H1, the method of proposed design increases applications developer`s productivity; H2, the presented storage structure improves the decision-making process; H3, using the Graphical User Interface improves the user decision maker`s productivity.

We establish therefore three hypotheses of cause-effect, which use respectively a dependent variable, VD, (the dependent variable, the object of the investigation) and an independent variable, VI, (the variable explanatory factor likely to explain the dependent variable).

- H1: the proposed method (VI) increases applications developer`s productivity (VD).
- H2: the proposed storage structure (VI) improves the decision-making process (VD).
- H3: using the Graphical User Interface (VI) enhances user decision maker`s productivity (VD).

The study group consisted of six students from the Master of Information Technology, therefore they are assumed to have homogeneous characteristics. The study group, for simplifying purposes, assumes the role of application`s developer to evaluate H1 and H2 and end-user for H3.

5 Research Development

The study group was instructed, at three different levels, on the main characteristics of the design method (H1), the storage structure (H2) and query interface (H3). The explanation was made avoiding comment on the benefits and limitations of the proposal. In addition, students were asked to describe answers as detailed as possible.

The research was made on a prototype based on Eclipse technology, which implements the design method of the Historical Data Warehouse, the Graphic Query interface and the automatic execution of SQL statements.

In the first case (design method) an explanation of the proposed method and the characteristics of the prototype was performed, then each participant in his workstation, used the tool in the design of the Historical Data Warehouse from a source data model (ER model) given. After the experience, they were given a questionnaire. In the second case (storage structure) the main features of the temporal multidimensional model were explained. After the experience, they were given a questionnaire. Finally, in the third case (query interface), they where explained of the characteristics of the interface and its use; then the group was offered different types of queries, both temporal and decision-making ones, and were encouraged to resolve them first manually using SQL statements and then using the Graphical User Interface. After the experience, they were given a questionnaire.

5.1 Hypotheses Assessment Questionnaires

There were three different types of open-type questions, each of them aimed to determine how the hypotheses were evaluated by the students. We considered three issues raised in the scenario: the design method (H1), the storage structure (H2) and the Graphical User Interface (H3).

The following questions were aimed at assessing the impact of using the design tool from the considerations expressed by the working group on the use of the prototype.

(H1.a.) For the design of a storage structure is initially important to determine user requirements. The proposed method starts by using the application source data model, in particular an ER model. Does that allow an effective design of the Historical Data Warehouse? Evaluate the benefits and drawbacks of this approach.

(H1.b.) The method proposes the marking of attributes, temporal entities and relationships and the primary fact of analysis, from which it generates the transformation process. Analyze and evaluate the proposal in terms of the intuitive method and its effectiveness in determining temporary constructions.

(H1.c.) The next step in the design is the marking of an Attribute Graph for the determination of the temporal and non temporal dimensions, hierarchies and measures. Analyze and evaluate the proposal in terms of the intuitive method and its effectiveness in determining the main characteristics of the Historical Data Warehouse.

The second set of questions were designed to evaluate the benefit of the storage structure that integrates a Data Warehouse and the Historical Data Base in one model from the explanation of the storage structure and different temporal and decision-making queries that can be performed on it.

(H2.a.) In a Data Warehouse the time is implicit in the storage structure, but it refers to the time when the transactions were performed, but it does not specify how or when the values of the entities, attributes and relationships associated with these transactions have varied through time. Evaluate the integrated model for the two storage structures.

(H2.b.) The data model allows to perform both temporal and decision-making queries. Does the data structure facilitates queries and extends the range of the same? Evaluate advantages and disadvantages of the model.

(H2.c.) The model allows, in addition to the typical decision-making queries, to determine how temporal attributes, entities and relationships vary through time. Analyze the types of generic queries that are solved and detail the advantages and disadvantages.

Finally, the following questions were aimed at assessing the benefits of the graphical query interface that allows, through marks, to automatically resolve temporal and decision-making queries from the use of the prototype by queries on the data structure.

(H3.a.) The traditional way to obtain information about a storage structure is through queries using SQL statements; they generally are iterative and often complex for non-technical users. Evaluate the query interface from its use.

(H3.b.) Do you think that the Graphical User Interface is an intuitive alternative to the queries? Evaluate advantages and disadvantages.

(H3.c.) The Graphical User Interface allows to resolve queries by markings on the graph. Evaluate the advantages and disadvantages of the marks on the graph.

6 Data Obtained

From the questions made, the answers by the students evaluated are described, literally. Each box corresponds to the response of each student evaluated in the order in which the questions were presented. We divided the report into three parts: The design method, the storage structure and the Graphical User Interface.

Technical details: Date: June 19, 2010, Time: 12:30, Venue: School of IT, UAI, San Juan 960. Buenos Aires City. Participants: 6 students of Master of Information Technology, School of Information Technology of the Universidad Abierta Interamericana.

6.1 Design Method

Below are the answers to the questions about the design method (H1.abc):

a)	Yes. It seems to be a practical and quick way to generate a Historical Data Warehouse.
b)	The method is very intuitive and effective.
c)	It is easy to dial in the graph and thus determine the properties that we want or do not want to keep.

a)	I believe that it allow an effective design of a Historical Data Warehouse, but I am not able to assess the benefits and drawbacks because I do not have other models to achieve this goal in mind.
b)	It is intuitive. The effectiveness cannot be assessed with the standard of proof made, nor its sturdiness. It sounds promising.
c)	It is fairly intuitive. It is positive that we can eliminate unnecessary attributes for queries in graphs.

a)	The ER model allows me to gain clarity in defining user requirements, allowing us to define entities and relationships involved in the model.
b)	It is very important to set attributes, entities and relationships as we can determine temporal fluctuations that affect the development of the business and get their measurements.
c)	It is very important to identify dimensions, measures and hierarchies through the graph. Using my knowledge and experience I can identify them better.

a)	I think the ER model allows an effective design. However, not many companies have updated the ER model to take it as a starting point to apply the transformation. It would be necessary to start with a reverse engineered from the tables for the ER model.
b)	It seems intuitive.
c)	It seems intuitive.

a)	It is effective because it allows to have the entities and relationships clearly identified because they are needed for knowledge of the model.

b)	It is effective in the determination of temporary constructions as it allows queries on variations over time
c)	It allows in the graph to prune the attributes that are not important and thus make a more efficient model.

a)	It seems appropriate because the data model shows the essence of the business and, therefore, the appropriate technical personnel is able to suggest a starting model.
b)	It seems simple enough and in this simple practice it seems to be effective.
c)	Ditto the previous answer.

6.2 Storage Structure

Below are the answers to the questions about the storage structure (H2.abc):

a)	The integrated model allows complex queries on the data model and its temporal relationship.
b)	The data structure facilitates queries in wide range of them.
c)	The model allows a wide range of possibilities in terms of queries that can be performed and the interrelationships that it allows.

a)	It obviously allows to get lots of information, particularly with regard to decision-making; of course it is also another level of complexity and resources and access speed.
b)	It facilitates and extends the range of the queries and it allows to make combinations. So when there are a lot of variables to access the information the alternative used by this development is enhanced. To describe the disadvantages I should compare it with other data structures, but I was unable to assess it during this experience, although I think it is impractical for static queries bounded.
c)	It seems that it enables a large number of typical SQL queries.

a)	It is very important to set the two storage structures built into a model because I may get oscillations or changes in data over temporal attributes (queries)
b)	Yes, I think the integrated model is optimal for temporal or decision-making queries because it allows me to see the structure of the business.
c)	Yes, it let me get queries with the effect of time which is very important in business decisions for future guidelines. For this, it is necessary to be clear about where I set my model attributes, relationships, hierarchies and dimensions

a)	It sounds like a good idea to make a more complete analysis.
b)	I think this structure facilitates the assembly of the queries, and give the possibility to conduct further queries by people without knowledge of SQL.
c)	Generic queries resolved are the most important and I am not sure if it can solve complex queries.

a)	The query graph model integrates transaction structure and historical data in a single model.
b)	It allows to create the SQL query quickly by selecting the order of query.
c)	It solves temporal queries about attributes that we selected as temporal. It resolves decision-making queries having marked previously the entities for that query by selecting the type of function and defining the order that will generate a SQL query.

a) Obviously it is more complete by allowing increasing complexity of queries relating to temporal changes.
b) Undoubtedly it enhanced the queries by complemented them with the variation attributes.
c) The possibility of including temporal changes of the attributes can help to understand certain behaviors of the entities.

6.3 Graphical Interface

Below are the answers to the questions about the graphical interface (H3.abc):

a) The interface is great and facilitates the queries, without having to write in SQL, or handle that language.
b) Yes, the interface is an excellent alternative for making queries by the user in decision making aspects.
c) It is very simple to use and provides good functionality in the queries.

a) It is a good idea but the interface is still in a rudimentary state: we all had to ask how to use it and an untrained user could not do anything with them. They said it is in the experimental stage and it is so.
b) Of course, I refer to the previous comment, it is very a good idea but it still did not come out of that state.
c) The model would be very good if there were "marks", because today we have to check the status of all graphs for any query: there is nothing in the graph to indicate their selection or order.

a) The traditional way is tedious as I need to know the language (syntax) to establish queries in a timely manner, limiting its use if the person is non technical.
b) It facilitates the identification of items according to my knowledge in the business management treatment.
c) The marks on the graph let me clearly define the queries allowing to identifying temporal or non temporal attributes, orders and other elements involved in the execution of the query and then conclude with the definition of the query in SQL.

a) On the graphical interface, you have to use it a bit to understand the criteria by which you have to choose the entities and dimensions. Once this is understood, it is not complex. To generate SQL queries is quite simple and can be performed by a person without knowledge of SQL.
b) As I said in the previous point, it takes a few minutes to understand the logic of the interface.
c) The marking is simple once you understand the logic.

a) It is simpler because we have to define what attributes are temporal and what entities serve to decision making process.
b) Viewing the model and positioning in the entities defined for the decision making process, it is easy to define what the function to be consulted is and on which groups identifying with a number the order of the group.
c) Yes. Ditto previous answer.

a) It is an easier form for non-technical users and it will be even easier in the finished product, with a more friendly Graphical User Interface
b) Yes, in the practice we perform in the classroom.
c) No disadvantages found within the scope of practice.

7 Evaluation of Responses

From the evaluation of the responses above, we can draw some conclusions on the proposals submitted. The objective in the evaluation of the design method was, first, to establish whether to start the design by data model was an effective option. By the answers, this approach seems to be appropriate as a design method. The premise of what happens when I do not have the initial input and the proposal to use reverse engineering to get the data model opens a future research area. It also opens the possibility of evaluating other requirements capture, which eventually could lead to a data model that fits the proposed method. The marking of the model as a means to obtain temporal constructions and the main fact of analysis was, in general, considered intuitive, although the idea was reduced due to the limitations of the prototype used. However, with some practice, it was easy and friendly. Finally, the evaluation of the marking in the Attribute Graphs for determining temporal and non-temporal dimensions, measures and hierarchies was generally similar to the previous replies.

In the evaluation of the integrated storage structure, a positive opinion can be seen, because, in general, it is considered an innovative proposal that facilitates combined queries allowing a wider range of temporal or decision-making queries.

Finally, the use of the graphical interface was, after a brief training, simple and intuitive. The automatic query creation was a simple way to obtain information for non-technical users. While currently marking is implemented through modification to model properties, in general, people assessed that the idea is applicable and consider the marking is simple once you understand the logic.

8 Conclusion

The work was designed to perform a method of empirical corroboration of the Historical Data Warehouse design, the graphical query interface and the creation of SQL statements, using a prototype based on Eclipse technology. The study group consisted of six students from the Masters in Information Technology; therefore they have homogeneous characteristics. There were three different types of groups open-type questions, each of them aimed to determine how the hypotheses were evaluated by the students. From the assessment and taking into account its limitations, we consider that the proposal constitutes a valid alternative in the design of temporal multidimensional structures, both in the proposed design method and in the storage structure and in the graphical query interface.

Finally, we believe that the research presented makes a contribution to the scientific perspective in the area of software engineering as aligned with the approach that the Empirical Software Engineering proposes. We focus our work on experiments on software systems and propose an experimental approach on them. By creating cause-effect type hypotheses and from a controlled experiment (qualitative research), we confronted these hypotheses with reality, in order to draw a conclusion about the usability features of the methods, models and interfaces proposed.

References

1. Denning P. J., "What is Experimental Computer Science?" CACM 23(10):543–544, October, 1980.
2. Shaw, M. What makes good research in software engineering? Int. Jour. of Soft. Tools for Tech. Trans., 4(1):1–7, 2002.
3. Snodgrass, Richard T.: Towards a Science of Temporal Databases. 14th International Symposium on Temporal Representation and Reasoning (TIME'07) 0-7695-2936-8/07.
4. Lázaro M., Marcos, E.: Research in Software Engineering: Paradigms and methods. Workshop on philosophical Foundations of Information Systems Engineering, CAISE, Oporto, (2005).
5. Basili, V.R. The Role of Experimentation in Software Engineering: Past, Current, and Future, Proceedings of the 18th International Conference on Software Engineering, Berlin, Germany, March 25-29, pp. 442-449, (1996).
6. Myers, M. D. Qualitative Research in Information Systems. MIS Quarterly, 21:2, pp 241-242. Recuperado de MISQ Discovery. (1997)
7. Hernández Sampieri R. Fernández-Collado C. Baptista Lucio P. Metodología de la investigación. McGraw Hill. México. (2006).
8. Marcos, E. Investigación en Ingeniería del Software vs Desarrollo Software. Actas del 1er Workshop en Métodos de Investigación y Fundamentos Filosóficos en IS y SI. November, pp. 136-149. (2002)
9. Easterbrook S., Singer J., Storey M., and Damian D. Selecting Empirical Methods for Software Engineering Research. Guide to Advanced Empirical Software Engineering, (2007).
10. Sjøberg D.I.K., B. Anda, E. Arisholm, T. Dyba, M. Jørgensen, A. Karahasanovic, E. Koren, and M. Voka´c, "Conducting Realistic Experiments in Software Engineering" Proc. First Int'l Symp. Empirical Software Eng. (ISESE '2002), pp. 17-26, (2002).
11. Cataldi, Zulma, Lage, Fernando. Diseño y organización de tesis. Nueva Librería. Buenos Aires. (2004).
12. Neil C. "Diseño de un almacén de datos histórico en el marco del desarrollo de software dirigido por modelos". Tesis presentada a la Universidad Nacional de la Plata, Buenos Aires, Argentina, para optar al grado de Doctor en Ciencias Informáticas. La Plata, Argentina. (2010).
13. Inmon W. H. "Building the Data Warehouse". John Willey. pp 576. (2005)
14. Chaudhuri S., Dayal U.. "An overview of data warehousing and OLAP technology". ACM SIGMOD Record. Volumen 26. Número 1, pp. 65-74. (1997).
15. Golfarelli M., Maio D., Rizzi S. "Conceptual design of data warehouses from E/R schemes". Proceedings 31st Hawaii International Conference on System Sciences. Hawaii, USA. (1998).
16. Hurtado C., Mendelzon A., Vaisman A. "Maintaining data cubes under dimension updates". Proc. of the 15th Int. Conf. on Data Engineering. Sydney, Australia. (1999).
17. Eder J., Concilia C. "Evolution of dimension data in temporal data warehouses". Technical Report. (2000).
18. Eder J., Concilia C. Morzy T. "A model for a temporal data warehouse". Proc. of the Int. OESSEO 2001 Conference. Rome, Italy. (2001).
19. Neil C., Ale J. "A conceptual design for temporal Data warehouse". 31º JAIIO. Simposio Argentino de Ingeniería de Software. Santa Fe. Argentina. (2002).

20. Franck Ravat; Olivier Teste, Ronan Tournier; Zurfluh, Pilles. Algebraic and Graphic Languages for OLAP Manipulations International Journal of Data Warehousing and Mining, (2008).
21. Liu B. and Jagadish H. A spreadsheet algebra for a direct data manipulation query interface. In ICDE. (2009).
22. Reiss S.P. A visual query language for software visualization. In: Proceedings of the IEEE 2002 Symposia on Human Centric Computing Languages and Environments (HCC'02), pp 80–82. (2002)
23. Neil C, Pons C. Aplicando MDA al Diseño de un Datawarehouse Temporal. VII Jornada Iberoamericana de Ingeniería de Software e Ingeniería del Conocimiento. Lima, Perú. (2007).
24. Neil, C.; Irazábal, J.; De Vincenzi, M.; Pons, C. "Graphical Query Mechanism for Historical Data Warehouse within MDD," Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the , vol., no., pp.183-192, 15-19 Nov. (2010)
25. Tryfona N., Busborg F., Christiansen J. G. B. “starER: a conceptual model for data warehouse design”. In ACM Second International Workshop on Data Warehousing and OLAP (DOLAP'99). Missouri, USA. (1999)
26. O. Romero, A. Abelló. “MDBE: automatic multidimensional modeling”. ER 2008. California, USA. (2008).
27. Pons C., Giandini R., Pérez G. “Desarrollo de software dirigido por modelos. conceptos teóricos y su aplicación práctica”. McGraw-Hill Education. Buenos Aires, Argentina. pp 300. (2009).
28. www.lifia.info.unlp.edu.ar/eclipse/pages/tesis_neil.htm