THE USE OF FORMS TO INCREASE THE QUALITY OF A CONCEPTUAL DATA MODEL

Liviu PERNIU
Francisc SISAK
Delia UNGUREANU
Catalin BUJDEI

Automatics Department, "Transilvania" University of Brasov, M.Viteazu Street, no.5, 500174,
Brasov, Romania, phone/fax: +40 0268 418836,
liviuperniu@yahoo.com, sisak@unitbv.ro,
delia@deltanet.ro, cbujdei@vision-systems.ro

Misunderstandings in real world representations will conduct, sooner or later, to some errors arising due either to data redundancy or information lost. A severe analysis has to be provided in data models respect to requirements, logic business representation and so on to avoid major impact of the errors. Such a technique increases the accuracy and the quality of the model. The aim of the paper is to propose a set of forms to help database developers in a correctness and reliable model development.

Keywords: errors, conceptual model's errors, forms for errors correction, data-information conversion, model's quality.

1. INTRODUCTION

Our new society, flooded by many types of information on many channels, needs much more complex strategies and tools for store, process and especially for real good interpretation of all this information. Someone needs to translate information in data and organize data in such a manner that they can be retrieved promptly and exactly to reach the target [3]. In this situation the most efficiently solution is to store data in databases and use them in diverse applications.

It is well known that databases are considered to be electronic replica of traditional libraries. In these electronic libraries the librarian has a new name, on the rise of his competencies, and this name is known as database administrator. Database is considered to be a set of interrelational data, and Database Management System is the software to control and manage the access to data.

Unfortunately, such a metaphor falsifies the real mean of databases, because they don't work with information like a library does, they work with data, and someone needs to translate data in information.

Just a simple data storing in a database doesn't offer the guaranty for reliable and useful information. Data themselves have no sense. Much more, they are just a mass of letters and digits and therefore someone can retrieve from there a variety of interpretations, more of them possibly wrong. In the same time the wish of the users is to manage information not data.

A reliable data model gives the chance to translate information to data and data back to information without a misleading account of their true nature [4]. To retrieve information means to put data in a specific context, so they get a true meaning of a situation. To get information means to put the available data in some context, so data provide a true meaning. To reach this target it's very difficult due to the complexity of real world and represents the most important source of errors for a good information retrieving [5].

It is a good practice to be aware that you deal with wrong information, but in many cases this is very difficult to do and you may use some of bad information. The use of bad information may lead to decisions that may represents a real disaster for the user (is it well known that "it's better to have no information at all rather than use wrong information").

One major cause for raising errors in databases is databases' complexity (because they model a complex world) and in these circumstances the human mind cannot make it ("the capacity of human mind to avoid mistakes in a judgement chain is very low"). For understanding of the real world it is necessary to build models. Models are simplified representations of the real world and help someone for a better understanding. It is much easier to work with models than with the real situation, and they help to make decisions in a simplified environment [5].

Models refer just the essential characteristics of the real world, not the peculiar, irrelevant characteristics, so they simplify the real world and lead to mistakes [1]. The objectives of modelling are both to understand the semantics, and to facilitate the communication. The issues appeared in real world analysis has to be considered in the database schema design, because schema is a rigid structure and application programers face with such a complex work in the case of a inadequate schema that they need, in many cases, to rebuild schema. In such a situation the efficiency of design stage is very low and time (critical factor in software production) for ending the project to the deadline forces a lot of efforts from design team.

For this reason, if a model does not exists or a model is improper or accurate enough then it represents the main source of errors in the incipient stages of design. There is no unique model for a specific situation to map into a schema. The best practice in this case is to create more than one model for every situation. To choose the appropriate model of a specific situation from many possible others means to provide a complex analysis with members of design team and database client [1].

An incomplete analysis, building inappropriate solutions, misundestanding of the real situation, superficial approach of phenomena, are some of the reasons for a spectacular project failure.

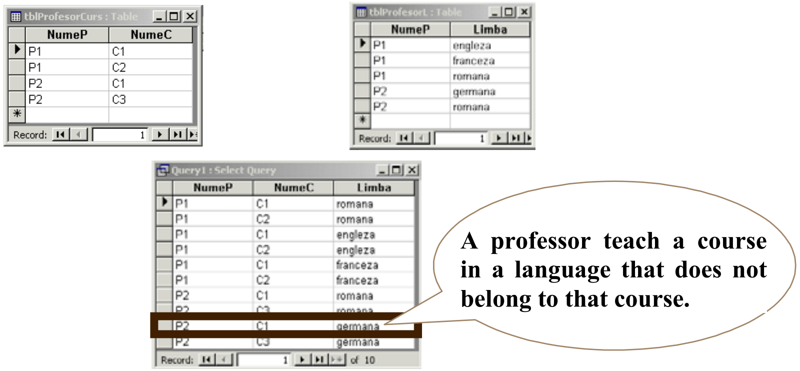
2. CASE STUDY

Considering the above mentioned issues, in a database comprising data about a specific faculty there are data referring to professors that teach a specific course, and simultaneously there data about langauges that professors speak. In the case of incorrect database schema, if you want to know languages for teaching courses, you might retrieve a set of records as you can see in the figure 1.

As you can see in the figure 1, from the database point of view everything is all right, but from the business logic point of view the result is wrong because a professor teach a course in a language that does not belong to that course.

Unlike file applications based, where everything is in function of application needs, the database applications based impose a new paradigm. It is very important to prevent errors in the starting phase of modelling by developing a methodology for clarifying of what it has to do with data. Otherwise, errors tend to propagate afterwards in the following stages of development causing cascade errors that are in a latent state in the physical model.

Such errors are very difficult to be observed and they lead to failures in some conditions. For this reason, they are stepped by testing phase in the end of physical model development.



NumeP	NumeC
P1	C1
P1	C2
P2	C1
P2	C3

NumeP	Limba
P1	engleza
P1	franceza
P1	romana
P2	germana
P2	romana

NumeP	NumeC	Limba
P1	C1	romana
P1	C2	romana
P1	C1	engleza
P1	C2	engleza
P1	C1	franceza
P1	C2	franceza
P2	C1	romana
P2	C3	romana
P2	C1	germana
P2	C3	germana

A professor teach a course in a language that does not belong to that course.

Fig. 1. Retrieving erroneous results from an incorrect schema.

It is important to reduce the *conceptual errors* or *logical errors* to avoid the failure of the system due to the impossibility of inserting data, deleting data or updating data (to avoid retrieving false informations). For this and other reasons that are beyond the scope of this paper, to increase data availability and to avoid many of the conceptual and logical errors, we developed a set of forms that are introduced subsequently in this paper**.** As we presented in this paper, the probability of error appearance in databases is in function of data-information transformation process (and vice versa) is faulty due to an inappropriate or a lack data model development mismatching the system or the modeling situation.

3. THE FORMS

They are the result of a thorough analysis and help database developers across the entire work within conceptual model development stage. All these forms are tied together and are revised and validated later in another stages of the database development. In the following we'll introduce a brief description of these forms.

a. Requirements analysis form

This form records the general information involving examining the business being modeled, interviewing users and managing to assess the current system and to analyze future needs, determining requirements for the business as a whole. This form helps for a better understanding of the situation to be modelled, using the results after free discussions with the user and shows an understanding by depicting essential characteristics, requirements and specifications that can be improved by modifier characteristics form.

b. Initial characteristics form

This form shows the main information about model and it is developed in the preliminary analysis stage of it. You fill this form for every characteristic you discover after the first identification of main characteristics. This form collect information about characteristic's function and the type of that characteristic.

c. Modifier characteristics form

You fill this form in the stage of the model preliminary analysis. You must update this form when a characteristic is removed or edited. This form collect information about characteristic's function and the type of that modified characteristic.

d. Initial model form

This form collect information about the first conceptual model you develop, like these: data identification of the model, data administer of the model, difficulty of the model appreciation, the origin of the model, and the goal of the model.

e. Model development status form

This form presents an abstract of the actual status of the model. For this you must provide: the number of situations you model, the number of tests you done, the number of tests you complete successfully, the total number of reported problems, the total number of fixed problems, the total number of specifications that you record for change, the total number of changes, the total number of suggestions and questions for a future analysis, the total number of solved suggestions and questions.

f. Model modification form

This form collects information for conceptual model modification. You fill it every time you receive a request for the model modification. For this, you have to provide: the necessity and importance of modification, the goal of modification, the requested time for change identification (error), the requested time for changing (repair), the type and the effect of the modification. When you discover errors you must specify the source of those errors (requirements, specifications, graphic representations etc.), the type of errors (semantic misunderstandings, model structure, relationships among entities, relationships among entities and environment, wrong data or derived attributes, wrong data correlation), error's characteristics (omission error if you pass over an entity or an attribute, administrative error, if you are wrong with an entity or an attribute, or transcription error).

g. Maintenance model form

This form is in the response to a modification request. Collects information about modification request (the type of modification, the reason of modification), modification implementation (data identification, the request time for model development, implementation and testing), entities affected by modification, the type of modification, the number of added entities or relationships, the number of removed entities or relationships, the number of edited entities or relationships.

h. Model estimates form

Collects information to form an opinion or judgement about model's parameters. Comprise information about: the starting of stages (requirements and specifications definitions, characteristics' definition, characteristics' implementation, characteristics' testing, integration testing, repairs, maintenance), estimate the size of the model, the total number of model's characterisitics (initial, new, modified).

i. Evaluation of model maintenance form

This form records the requested time for the model maintenance. Collects information about: total number of hours for maintenance operations, total number of hours for maintenance operations by maintenance types (correction, development, accomodation etc.), total number of hours to find out the causes of a defect appearance, or a request for improvement or accomodation, total number of hours for remodelling, total number of hours for model changing in the response to a modification request (this is a sum of requested time for model modification and the requested time for related documentation update), total number of hours for testing of edited or added characteristics (adds the time for integration testing of a new characteristic in the system), total number of hours for testing model as a whole, total number of hours for others maintenance activities (management, users or development team members appointments etc.)

j. Model completion statistics form

Collects the final statistics of the model. Comprises information about: the size of the model (number of entities, number of relationships, number of modifications, number of pages of documentations, modification's individuals), entities (new, initial, modified, simple modifications, major modifications), relationships.

4. CONCLUSIONS

There are some characteristics to mention in all of this work:

- conceptual model helps to diminish errors in database development because it offers a detailed analysis of requirements;
- the use of forms helps to diminish errors in database development because they offer a set of obvious and precise rules;
- using forms you can validate a conceptual model;
- the forms increase the quality of a conceptual model because they refine this model;
- decreasing the number of errors in conceptual model leads to decrease the cost of physical model and maintenance of databases.

In the future we'll develop a set of forms for a better construction of a logical model of information for information to data translation in database systems.

Even though this type of data management is very complex, can be brought to a modular, simplified form that reduces the probability of errors. In this case, the use of the helping tools is reduced to a minimum while the requests for functionality and performance are fully respected. In the same time another aspects are also respected: the rebuilding of the application based on users requirements, system management, interaction between the system components, the system security. It is very important to involve all the affected stakeholders: the system designers, the software developers, software administrators, and the final users. Even though such a project is a very extensive one, the advantages of such a system can be seen in the long run and are reflected in the cost of it. Good design have prove to its advantages during time.

5. REFERENCES

[1] Connolly, T. M., Begg, C. E.: Database Systems, Addison Wesley, 1998.

[2] DeMarco, T., Structured Analysis and System Specification, Dorset House Publishing Company, 1978.

[3] Hay, D., Introduction to Requirements Analysis, Prentice Hall, 2003.

[4] Hooks, I., Major Impact on Requirements, Software Technology Conference, 2002.

[5] Jones, C., Software Quality: Analysis and Guidelines for Success, 2000.

[6] Young, R. R., Defining the Real Customer Needs, Addison Wesley Professional, 2001.