

Database Design for Quality

Donatella Castelli, Elvira Locuratolo

Istituto di Elaborazione dell'Informazione

Consiglio Nazionale delle Ricerche

Via S. Maria, 46

56126 Pisa, Italy, . Telephone: +39 50 593406/403. Fax: +39 50 554342.

email: [castelli, locuratolo]@iei.pi.cnr.it

Abstract

The approach employed to define a formal design methodology for the development of easy to use, flexible, efficient and correct database systems is described.

Keywords

Database system quality attributes, design for quality, methodological tools.

1 INTRODUCTION

Good design is always the best approach to a quality product. But what is a *good design* and how to obtain it? In our opinion, a design is a *good design* if it results in a system which meets desired quality attributes. A good design cannot be thus extemporized but it must be defined carefully using appropriate methods.

In this paper, we propose an approach to define methods for good design. This approach will be described by defining ASSO (Castelli and Locuratolo, 1994), a formal database design methodology which ensures the development of *easy to use, flexible, efficient* and *correct* database systems. The approach consists of sequence of steps. Each step is characterized by a goal, a proposed solution and a demonstration. The initial goal requires to define a database design methodology for the development of systems that meet fixed quality attributes. The proposed solution is a definition of the database design methodology, whereas the demonstration is the mean to establish if the proposed solution satisfies the goal. If the solution is correct, the proposed solution becomes the next goal of the sequence. By transitivity, the final solution, defined by a set of methodological tools integrated appropriately, satisfies the initial goal.

The quality attributes of database systems developed by ASSO are widely recognized in the database area, however the existing database design methodologies (Batini, Ceri and Navathe 1992; Rumbaugh et al., 1991; Booch, 1991; Coad and Yourdon, 1991) prioritize either easiness of use and flexibility or efficiency. Moreover correctness is often treated only partially. In order to achieve the desired quality attributes, ASSO has been designed as a methodology consisting of two stages: conceptual design and refinement. The conceptual

design focuses on the construction of the conceptual schema, i.e. the specification of the database structure and behavior. An extended semantic model has been employed at this stage to guarantee both easiness in specifying the conceptual schema and flexibility in reflecting the changes occurring in real life.

The refinement stage transforms the conceptual schema into a schema acceptable by an object database management system which is nowadays considered the most promising with respect to efficiency. The conceptual and the object schemas are also represented formally as Abstract Machine models (Abrial, in press). Formality is used by the database designer to prove correctness; however, it is introduced as an orthogonal feature of the database schemas, i.e. it is completely transparent to the application user. As a consequence, correctness does not degrade the other quality attributes.

The rest of the paper is organized as follows: Section 2 lists the database system quality attributes; Section 3 describes the approach and discusses its benefits. Conclusions are given in Section 4.

2 THE QUALITY ATTRIBUTES

The approach adopted to define ASSO assumes that the database system quality attributes to be accomplished during the database design have been established. They are considered to be all equally important, i.e., none of them is prioritized. These attributes are:

easiness of use: the ability of the database system to provide a schema which can be understood easily and which can facilitate the writing of applications.

flexibility: the ability of the database system to provide a schema which permits the applications to be easily modified to reflect the changes occurring in real life.

correctness: the ability of the database system to provide a schema which satisfies its specifications expressed through the conceptual schema.

efficiency: the ability of the database system to access and store the database information using a limited amount of time and storage.

The quality attributes listed above can be classified as *external attributes* (Meyer, 1988) since they are detected directly by both the database application programmers and the database application users. The quality attributes perceptible only to the database designer are said *internal attributes* since they are the key to ensure that the external attributes are satisfied.

The approach illustrated in the next section begins taking into account the external attributes since only they really matter in the end.

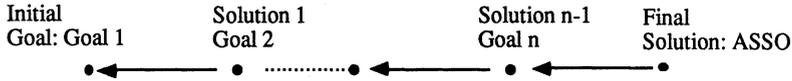


Figure 1 The stepwise approach.

3 THE APPROACH

This section presents the approach which has been adopted to define ASSO, a formal database design methodology for the achievement of quality database systems. The approach consists of a sequence of steps. Each step is characterized by a goal, a proposed solution and a demonstration. The initial goal requires to define a database design methodology which meets fixed external attributes. At each step, the proposed solution is a definition of the database design methodology, whereas the demonstration is the mean to establish if the proposed solution satisfies the goal. If this is true, the proposed solution becomes the next goal of the sequence, otherwise a new solution must be proposed. In order to satisfy the goal, the solution must have a set of properties which suffices to meet the goal requirements. Step by step the proposed solution refines the previous one, i.e. new methodological tools are chosen to be integrated into the previous solution. The sequence ends when all the significant choices have been done. The final solution is given in terms of methodological tools integrated appropriately. Figure 1 illustrates this approach: each step is represented by two successive points linked by an arrow directed towards the first point. The points represent respectively the goal and the proposed solution, whereas the arrow represents the goal satisfiability. Exploiting transitivity, the solution proposed for satisfying the final goal is also a solution for satisfying the initial goal and thus it permits the development of database systems which meet the desired external attributes.

3.1 The step sequence

Let us now introduce the sequence of steps.

Step 1

Initial Goal: Goal 1

Define a database design methodology for the development of an easy to use, flexible, correct and efficient database system.

The following solution has been proposed to satisfy the initial goal:

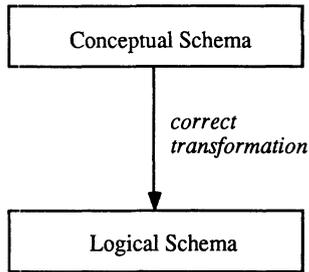


Figure 2 Solution 1.

Solution 1:

The methodology comprises

1. two modelling levels:

- (a) the *conceptual schema*, i.e. an high level description of the database structure and behavior;
- (b) the *logical schema*, i.e. a description of the database structure and behavior given in terms of a model supported by an efficient database management system;

2. a *correct transformation*, i.e. a semantic-preserving transformation from the conceptual schema to the logical schema.

This solution is illustrated in Figure 2.

Demonstration 1:

The semantic-preserving transformation guarantees that the logical schema is a correct implementation of the conceptual schema. As a consequence the applications can be written in an easy and flexible way referring the conceptual schema and at the same time they can be supported by an efficient database management system. □

Remark 1:

Although current database design methodologies use the above two modelling levels, they generally do not comprise a correct transformation to link them. As a consequence, they support a conceptual design which can be specified at an high-level description, but the database applications must be written referring the low level logical schema thus not achieving all the desired quality attributes.

Solution 1 can now be interpreted as a new goal to be satisfied.

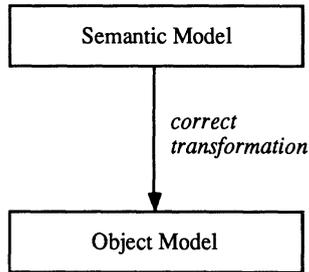


Figure 3 Solution 2.

Step 2

Goal 2:

Define a methodology which satisfies the properties expressed in Solution 1.

We propose the following solution to satisfy Goal 2:

Solution 2:

The methodology comprises

1. two modelling levels:
 - (a) the conceptual schema modelled by an *extended semantic model*, i.e., a semantic data model extended with an high level behavioral model;
 - (b) the logical schema, modelled by an *object model*;
2. a semantic-preserving transformation from the conceptual schema to the logical schema.

This Solution is illustrated in Figure 3.

Demonstration 2:

Semantic data models are considered to be the most appropriate models for representing the conceptual schema (Bouzeghoub and Metais, 1991) since they provides the concepts essential for supporting the application environment at a very high non-system specific level (Navathe, 1992). Extending them with an appropriate high-level behavioral model is sufficient to guarantee property (a) of Goal 2.

The object systems (Deux et al., 1990) have achieved a remarkable level of efficiency in managing information. The choice of a model which is supported by these systems suffices

to satisfy property (b) of Goal 2 . □

Remark 2:

The models we have chosen for the two schemas are methodological tools frequently used within the database design methodologies. What distinguish our methodology from the others is the correctness of the link introduced between the two schemas.

We now continue by proposing Solution 2 as the new goal to be satisfied.

Step 3

Goal 3:

Define a methodology which satisfies the properties given by Solution 2.

The solution proposed at this step is the following:

Solution 3:

The methodology comprises

1. two modelling levels:
 - (a) the conceptual schema, modelled by both an extended semantic model and an equivalent *formal model*;
 - (b) the logical schema, modelled by both a model supported by an object system and an equivalent *formal model*;
2. a *formal* correct transformation from the conceptual schema to the logical schema.

Figure 4 shows Solution 3.

Demonstration 3:

This solution satisfies Goal 3 since it associates an equivalent formal method with each schema and specializes Property 2 of the previous solution by choosing a set of possible correctness preserving transformations. □

Remark 3:

The idea of formalizing all the phases of the software design to prove the process correctness is well consolidated in the software engineering area where several proposals exists (Wirth, 1971; Partsch, 1990, Abrial, in press). Solution 3 transfers this idea to the database area. In our proposal, formality is a mean to prove correctness, but it is not visible to the database users. This implies that correctness does not degrade the other external attributes.

Figure 4 reflects that formality is an internal attribute since it concerns only the internal components of the database design.

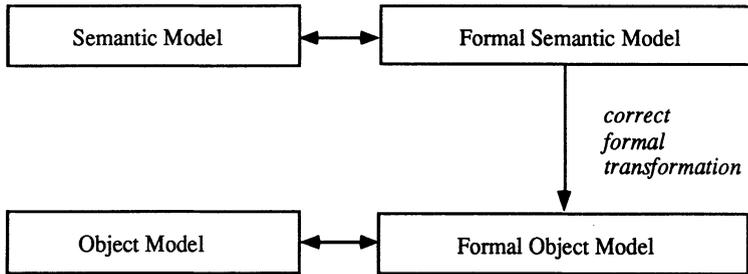


Figure 4 Solution 3.

Step 4

Goal 4:

Define a methodology which satisfies the properties expressed in Solution 3.

The following solution has been proposed to satisfy this goal.

Solution 4:

The methodology comprises

1. two modelling levels:
 - (a) the conceptual schema, modelled by both an extended semantic model and an equivalent *Abstract Machine model*;
 - (b) the logical schema, modelled by both a model supported by an object system and an equivalent *Abstract Machine model*;
2. a formal correct transformation from the conceptual schema to the logical schema defined by a *stepwise refinement*.

This solution is illustrated in Figure 5.

Demonstration 4:

This solution is a specialization of the previous one since it selects a particular formal model for the conceptual and logical schema formalization. Furthermore, it selects a special type of correct transformation, i.e. the stepwise refinement. It thus satisfies Goal 4. □

Remark 4:

We formalize our schemas using the Abstract Machine(AM) model.

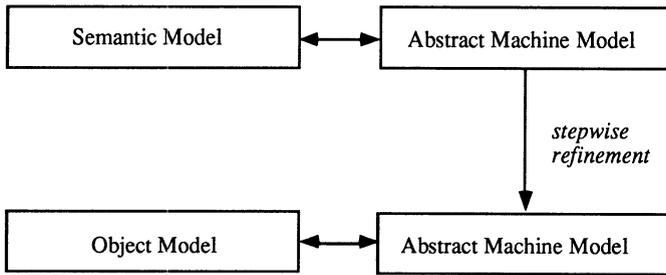


Figure 5 Solution 4.

AM is a mathematical model based on first order logic and a subset of set theory. It supports a model-based approach to the specification, i.e. a specification in which both a model of the state of a system and the operations which constitute its interface are described. This characteristic renders the AM model particularly suitable to be used for database system specifications. The AM model is employed within the B-Method (Abrial, in press), a formal method introduced in the software engineering area to construct correct programs, i.e. programs which satisfy their specifications. This method supports a design made of two stages, specification and refinement, both carried out within the same formal framework. The specification stage builds an AM incrementally. The refinement stage consists of a sequence of steps. At each step, an AM closer to an implementation is proposed and proof obligations which express the correctness of the new AM model with respect to the previous one are proved.

The stepwise refinement has been chosen for both the reasons: a) the implementation of a semantic model by means of an object model is a complex task which can be better approached with a stepwise refinement; b) the refinement supported by the B-Method is a stepwise refinement.

Step 5

Goal 5 is the last goal of the ASSO definition process. It imposes to satisfy the following requirement:

Goal 5:

Define a methodology which satisfies the properties expressed in Solution 4.

Solution 5: the final solution

The methodology comprises

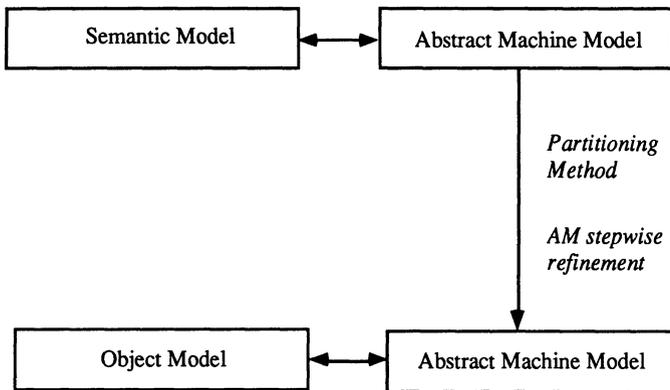


Figure 6 ASSO.

1. two modelling levels:
 - (a) the conceptual schema, modelled by both an extended semantic model and an equivalent Abstract Machine model;
 - (b) the logical schema, modelled by both a model supported by an object system and an equivalent Abstract Machine model;
2. a formal correct transformation from the conceptual schema to the logical schema defined by a stepwise refinement which consists of *two phases*: the *Partitioning Method* and the *Abstract Machine refinement*.

The final solution is illustrated in Figure 6.

Demonstration 5:

The stepwise refinement has been decomposed in two phases corresponding respectively to data refinement and behavioral refinement since the transformation is a complex task which can be better approached considering statics and behavioral aspects of schemas independently.

The Partitioning Method (Locuratolo and Rabitti, in press) transforms step-by-step the formal semantic schema in a schema with features of object models. These models enjoy the property that each object belongs to one and only one class. At each step, a new schema equivalent to the previous one is defined. This guarantees both the correctness of this phase and the compatibility of the resulting schema with the Abstract Machine Model.

The Abstract Machine Refinement defines a sequence of schema transformations from the output of the Partitioning Method to a formal object schema which can be translated easily into an object logical schema. At each step, the database designer proposes a new schema which reformulates the behavioral definition of the previous schema and proves first order formulas to establish the correctness of the step. By exploiting transitivity, also

the correctness of this refinement phase is guaranteed. \square

Remark 5:

In semantic models an object can belong simultaneously to any intersection of subclasses, whereas in object data models no object can belong simultaneously to two subclasses if one of them of them is not a subclass of the other. This characteristic of object models limits the flexibility in reflecting changes occurring in the real life. The Partitioning Method defines a link between class of objects at semantic level and class of objects at object level thus combining the useful features of both semantic and object models.

The Abstract Machine Refinement is a particular B-Method refinement. As a consequence, formulas simpler than the corresponding B-Method formulas are used to ensure the correctness of this phase.

The step 5 completes the methodology definition process. By exploiting the transitivity of the goal satisfiability, it follows that ASSO satisfies also the initial goal, i.e. it is a methodology for the development of database systems which meets the chosen external attributes.

3.2 The approach benefits

The approach demonstrates ASSO to be a database design methodology for quality; i.e. the database systems developed by ASSO achieve the chosen external quality attributes. The demonstration is organized step-by-step. As a consequence, if a proposed solution does not meet the goal properties, only a solution for this step must be given again.

At each step no specific solutions is imposed by this approach but several solutions are possible. The only constraint is that the proposed solution must have properties which suffices to meet the goal. In particular, our choices have been driven with the aim of selecting, when possible, consolidated tools whose contribution to the achievement of the fixed quality attributes is well understood. In doing so, we have turn our attention mainly towards tools typical of the database area. However, when these proved insufficient to our purposes, as for the definition of the correct transformation, we have directed our attention towards methodological tools of other areas.

Another benefit is that choices can be made gradually. At each step the attention is focused on few aspects. This simplifies the task of understanding which are the most appropriate methodological tools with respect to various factors such as tools at disposal, optimization criteria, etc..

The approach can be generalized to the case in which a goal can be decomposed in subgoal. When this generalization holds, the approach can be applied independently to each subgoal. This generalization is particularly useful for database design methodology of complex database systems.

Finally, although this approach has been described for the definition of ASSO, it is not strictly related to the database area, i.e. it can be applied to define a general design methodology.

4 CONCLUSIONS

This paper introduces a stepwise approach for defining a database design methodology for quality. It consists of a finite sequence of steps. Each step is associated with a goal, a proposed solution and a demonstration to establish if the solution satisfies the goal. This approach permits to demonstrate that the resulting methodology favors the development of database systems which achieve the following quality attributes: easiness of use, flexibility, efficiency and correctness.

A direction for future work is the generalization of this approach to the case in which the external attributes are not completely compatible. In the case a trade-off concept must be considered.

5 REFERENCES

- Abrial, J., R. (in press) *The B-Book*. Cambridge Press.
- Batini, C., and Ceri, S. and Navathe, S.B. 1992. *Conceptual Database Design: An Entity-Relationship Approach*. Redwood City, California: Benjamin Cummings.
- Booch, G., 1991. *Object Oriented Design with Applications*. Redwood City, California: Benjamin Cummings.
- Bouzeghoub, M. and Metais, 1991. E., Semantic Modelling of Object Oriented Databases. In *Proceedings of the 17th International Conference on Very Large Data Base, 1991*.
- Castelli, D. and Locuratolo, E. 1994. ASSO: A Formal Database Design Methodology. In *Information Modeling and Knowledge Bases VI, Stockholm, 1994* edited by H. Jaakkola and H. Kangassalo. Finland: IOS Press, 145-158.
- Castelli, D. and Locuratolo, E. 1995. Enhancing Database System Quality through Formal Design. In *4th Software Quality Conference, Dundee, 1995* edited by I.M. Marshall, W.B. Samson and D.G. Edgard-Nevill. Dundee. 359-366.
- Deux, O. and et.al., The Story of O2, 1990. *IEEE Transaction on Knowledge and Data Engineering*, 3(1), 91-108.
- Locuratolo, E. and Rabitti, F. (in press). Conceptual Classes and System Classes in Object Databases. *Acta Informatica*.
- Meyer, B., 1988. *Object-Oriented Software Construction*. International Series in Computer Science. University Press, Cambridge: Prentice-Hall.
- Navathe, S. B., 1992. Evolution of Data Modelling for Databases. *Communication of the ACM*. 35(9).
- Partsch, H. A. 1990. *Specification and Transformation of Programs*. Texts and Monographs in Computer Science. Springer-Verlag.
- Rumbaugh, J. and Blaha, M. and Premerlani, W. and Eddy, F. and Lorensen, W. 1991. *Object-Oriented Modeling and Design*. Prentice-Hall.
- Wirth, N., Program Development by Stepwise Refinement. 1971. *Communications of ACM*. 14. 221-227.

6 BIOGRAPHY

Donatella Castelli received her degree in computer science from the University of Pisa in 1983. Currently, she is a researcher at Istituto di Elaborazione dell'Informazione-CNR, Italy. Her interests includes database and multimedia design methodologies, formal methods, and re-engineering methods.

Elvira Locuratolo received her degree in mathematics from the University of Napoli in 1978. Currently, she is a researcher at Istituto di Elaborazione dell'Informazione-CNR, Italy. Her interests includes information modelling, database design and re-engineering methods.