

# Teaching database modelling on an undergraduate course: from semantic models to object orientation

T. Mohorič

*Faculty of Electrical and Computer Engineering,  
The University of Ljubljana, 61000 Ljubljana,  
Slovenia*

## Abstract

Rapid changes in information technology require from us to reconsider from time to time the relevance of faculty curriculum. This paper describes the reasoning related to the teaching of the subject Database Modelling and Design. First, the aims and objectives of the teaching process are considered and then the syllabus is presented. In the conclusion the return information on the quality of graduates is discussed.

## 1 Introduction

One of the main concerns of the present age is the rapid speed of environmental change. Past knowledge is dating fast. We now need to be asking what students should acquire, and to what extent it is necessary to adapt to the requests coming from industry and the economic sphere which want get graduates who would be able immediately to start productive work. What should be the ratio between basic and applicative knowledge, what should be the programs and the teaching techniques which are likely to benefit students most? All these questions also apply to the teaching of the subject *Database Modelling and Design* which will be discussed in the sequel.

Let us first introduce our university environment. The Faculty of Electrical and Computer Engineering is one of 20 members of the University of Ljubljana which was founded in 1919. Our faculty runs two curricular topics *Electrical Engineering* and *Computing and Informatics*.

The topic *Computing and Informatics* includes a 3-year course and a 5-year course, the latter leading the students towards the title *dipl.ing.* (eq. to B.Sc).

## 342 Software Engineering in Higher Education

Within the 5-year course the first year of study is the same for all students; in next years, apart from a few general subjects, there is a division into three specialities: Logic and Systems (eq. to computer hardware), Software, and Informatics.

Where are Database Systems taught? During the 3-year course the students attend the subject Introduction to Database Systems in the second year, while in the 5-year course the subjects *Principles of Database Systems* and *Database Modelling and Design* are taught in the third year. The subject Principles of Database Systems is common for both specialities Software and Informatics. It encompasses the fundamental chapters from the book by Elmasri and Navathe [8]. The subject Database Modelling and Design is taught to Informatics students only.

## 2 Aims and Objectives and How to Reach Them

The learning objectives are typically in contradiction. While the graduate should acquire the knowledge he should be able to use directly in his professional work, he should also attain the basic knowledge which would enable him to follow the developments in Informatics and related fields, and make him responsive to unpredictable future situations. Between both kinds of knowledge a proper balance should be achieved, but this is not always an easy task.

**Basic knowledge.** Which basic knowledge should the graduate acquire by *Database Modelling and Design*? Among the basic, long-lasting elements of knowledge one can certainly include the comprehension of the basic concepts of modelling, different kinds of models (i.e. data, semantic, knowledge, object models), and corresponding modelling techniques.

**Application oriented knowledge.** Which practical skills should the graduate be able to master when leaving the school? The skills which the environment desires, are often quite difficult to identify, since they are changing with time and are often conditioned by the needs of the production programs of big firms. In principle, the graduate of Informatics has nearly unlimited possibilities [11] as far as employment is in question. The problem of building or re-engineering information systems appears in all firms, whether or not these systems relate to computer field. Thus, the economy certainly expects from graduates a knowledge of information systems development and also knowledge of database design as a constituent part of IS. The intention of *Database Modelling and Design* is that the students should acquire instantly operative skills, such as design of classical formatted databases (in the context of IS development), and design of OO databases, which are at the moment highly demanded.

In addition to technical education graduates should also acquire some knowledge of the business world. The great majority of information systems

support the business systems of companies, so it is expected that the graduates should at least be able to talk to their users about the business problems, and therefore be context oriented [6]. There is the point at which we meet the "expectation gap" - the firms expect that the students will possess exactly those skills which they are at the moment badly lacking.

**Prerequisites.** Which are the required prerequisites? From the mathematical point of view, the most important is a knowledge of first-order logic and discrete structures with emphasis on graphs and lattices. Equally important is knowledge of surface-structured data models (hierarchical, network, relational data model) and database management systems. Of course, we must also not forget obligatory proficiency in English - most available publications in computing are published in English.

In order to reach the established objectives, all the activities related to students should be suitably co-ordinated: *lectures*, *laboratory practice* and *off-study activities*. Most important of all, perhaps, is the content of the lectures. The quality and the content of the lectures depend, of course, almost entirely on the teacher, whose duty is to lecture on relevant themes persuasively and understandably. Also, up-to-date textbooks play an important role. Laboratory practice and the system of homework assignments enable the student to check and to strengthen the knowledge acquired by attending lectures and studying the professional literature.

It is also useful to involve the students in work on real world projects so they become familiar with the real problems arising from information systems building. During their work the students should be under the supervision of their tutors so that they are not engaged only on routine work, but are also able to get hands-on experience with global problems.

### 3 Database Modelling and Design Syllabus

How did we compose the program of the lectures and what did we want to achieve? This we shall try to present in the sequel. Similar programs are established at some other departments of different universities [10].

The phases of modelling the database [3] encompass the real world abstraction, its representation by means of conceptual models, and its mapping into the target model (Fig 1). In like manner the contents of *Database Modelling and Design* is structured:

- Real world perception: classification, aggregation, generalisation / specialisation, meaning triangle
- Conceptual models: extensional semantic networks, conceptual graphs, ER data model, Object data model

## 344 Software Engineering in Higher Education

- Transition to target models

### 3.1 Real World Perception

There exist the different starting points for the modelling. It can be based on direct perception of the real world, on its description by natural language, or by using data structures obtained from an existing system which should be re-engineered. Our conviction is that this phase is the most important, and should be taught most carefully.

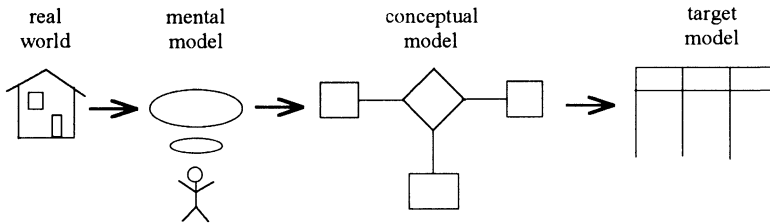


Figure 1: The modelling framework

The starting point, presented in fig. 1, presumes that the designer builds the mental model in his mind by means of direct perception of the real world. The building of the mental model incorporates abstractions, such as classification, aggregation and generalisation.

*Abstraction* is the selective examination of certain aspects of a problem domain. The goal of abstraction is to isolate those aspects which are important for some purpose and to suppress those that are unimportant [4]. All abstractions are, of course, incomplete due to the omission of insignificant facts.

*Classification abstraction* is used for defining one concept as a class of real-world objects characterised by common properties. The same real-world object can be classified in multiple ways. Aggregation abstraction defines a new class from a set of other classes representing its component parts.

*Generalisation abstraction* defines a subclass relationship between two classes, and all the abstractions defined for the superclass are inherited by its subclasses. By subclass relationship classes can be partially ordered and represented by the Hasse diagram. Multiple inheritance at OO systems should also be stressed.

The next phase is the transfer of the mental model to a medium (e.g. paper) so that others especially users can gain an insight into it. For mental model representation the symbols should be used which is wonderfully illustrated by meaning triangle (Fig 2) from [1]. For the denotation both, the classes and the objects, the symbols are used, where the objects are denoted by object identifiers and classes by class names.

### 3.2 Conceptual models

A data model is a collection of concepts that can be used to describe a set of data and of operations to manipulate the data. When a data model describes a set of concepts from a given reality, it is called a conceptual model. Many different models exist, only some of which are mentioned in our curriculum, i.e. extensional semantic networks [2], conceptual graphs[1], ER data model, [3] and object data model [4,5,7].

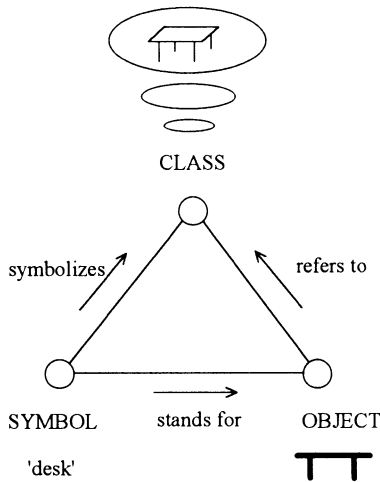


Figure 2: The meaning triangle

**Extensional Semantic Networks.** The approach to the design using ESN can be worthwhile if the starting point of the design is the natural language description of the real world. ESN is basically a hierarchical acyclic graph, the nodes of which are concepts - objects and actions; arcs are ordered pairs of concepts, and both belong to grammatical categories. By means of ESN, complex statements can be represented which can be quantified by using a universal and existential quantifier.

**Conceptual graphs.** CG form a knowledge representation language closely related to the first-order logic. In the graphs, the concept nodes represent entities, attributes, states, and events, while the relation nodes show how the concepts are interconnected. By use of CG it is possible to represent the intensional as well as extensional situations due to the concepts, which can be generic or individual. There exist also actors, so CG covers in a way also data-flow diagrams.

**Entity Relationship data model.** This model is certainly the one most widely used, especially as the intermediate model at relational database design. There

## 346 Software Engineering in Higher Education

are many derivatives of this model. The model and the associated design methodologies, which we chose to lecture, has been acquired from [3] in slightly modified form. The ER model is especially handy because there exists a palette of development tools which use this model, they can be find also in the MS DOS environment.

**Object data model.** This describes the structure of objects in a system - their identity, their relationships to other objects, their attributes, and their operations. The object model provides the essential framework into which dynamic and functional models can be placed. Objects are the units into which the real world is divided. Lately, this has become the most promising model because it is expected that object orientation will be used in the majority of new software designs, due to the long-expected reusability of software.

### 3.3 Transition to target models

There remains one further step to be done. All the models should be transformed into the form as understood by DBMS, used for implementation. Each conceptual model should be described using the concepts of the (surface-structured) target model. By transforming ER model into a relational one, the normalisation can be avoided by following certain rules. In our course the special attention is focused on the relational and object databases design.

## 4 Laboratory Practice

The laboratory practice program is devised by the teacher, and the teaching assistant and is supervised by the latter. Apart from the scheduled laboratory practice, the students are able to do their homework at all other times, freely using the laboratories and corresponding equipment.

The LW program depends to a great extent on the existing hardware and software. The predominating platform in our laboratories is still MS DOS, but it remains sufficient for programming in C and C++ languages (Borland's Turbo C and C++). For more demanding projects there exists a range of RISC workstations (mostly HP, IBM, SUN and Silicon Graphics) using the UNIX operating system. We have also installed different DBMSs, such as Allbase/SQL, Ingres, Oracle in Interbase, which are used in the context of database design and application design in the relational environment. The situation with development tools, however, is somewhat worse. For Allbase/SQL databases we can use HP4GL, and for the design of Ingres databases there are Ingres/ABF, Windows4GL and Westmount.

The students' projects encompass mainly the design of relational databases by means of CASE tools, as mentioned before. A few prototypes of OO databases were also implemented, and these are now used as test-beds. Thanks to the Globewide Network Academy and its courses, which are accessible on Internet

(WWW), we recently introduced the self-study of C and C++, which has become very popular with students.

## 5 Conclusions

We were trying to present our approach to the teaching of *Database Modelling and Design*. Our main concern is to lecture the adequate themes by following the development of the field which is changing rapidly. We have been trying to follow the curricula of related institutions and to incorporate all good elements into our programs. But, are we successful?

It is not easy to answer the question: Did our graduates obtain enough of the skills they need, and are they the right ones? In our environment we try to derive this information from different sources. The first direct source is the annual meeting of our former students, at which we try to get their opinion about the usefulness and appropriateness of the knowledge they acquire during their studies, and of course, to discover what they most miss. The second source is simply the observation of what kind of posts our graduates obtain and in what kind of firms. The third source is that of the firms which pass on to us data about their opinion of the quality of our graduates, and about their strategic plans.

According to the information gained from our former students we are moving in more or less the right direction, while from the firms we have quite often heard that as observed elsewhere [9] one of the flaws is the graduates' lack of practical experience. This is something we shall try to solve in our forthcoming curriculum.

## References

1. Sowa, J.F., *Conceptual structures: Information processing in Mind and Machine*, Addison-Wesley, Reading, Mass., 1984.
2. Janas, J.M. & Schwind, C.B., Extensional Semantic Networks: Their Representation, application, and Generation, in *Associative Networks - Representation and Use of Knowledge by Computers*, ed. N.V.Findler Academic Press, New York, 1979.
3. Batini, C., Ceri, S. & Navathe, S.B., *Conceptual Database Design*, The Benjamin/Cummings Publishing Company, Redwood City, Ca., 1992
4. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. & Lorensen, W., *Object-Oriented Modeling and Design*, Prentice-Hall International, Englewood Cliffs, N.J., 1991
5. Khoshafian, S., *Object-Oriented Databases*, John Wiley & Sons, New York, 1993

## 348 Software Engineering in Higher Education

6. Trautth, E.M., Farwell, D.W. & Lee, D., The IS Expectation Gap: Industry Expectations versus Academic Preparation, *MIS Quarterly*, 1993, **3**, 293-307
7. Lieberherr, K. & Xiao, C., Formal Foundations for Object-Oriented Data Modeling, *IEEE Transactions on Knowledge and Data Engineering*, 1993, **3**, 462-478
8. Elmasri, R. & Navathe, S.B., *Fundamentals of database Systems*, Addison-Wesley, Reading, Mass., 1993
9. Undergraduate Teaching in the Department of Electronic and Electrical Engineering at the University of Surrey, *WWW pages*, <http://www.ee.surrey.ac.uk/EE/AnnualReport/UG.html>, 1994
10. Curriculum for the Study of "Applied Computer Science" at the University of Klagenfurt, *WWW pages*, [gopher://borneo.ifi.uni-klu.ac.at/00/Teaching/Curriculum%20for%20the%20study%20of%20%27Applied%20Computer%20Science%27](http://gopher://borneo.ifi.uni-klu.ac.at/00/Teaching/Curriculum%20for%20the%20study%20of%20%27Applied%20Computer%20Science%27), 1994
11. Degree Courses at University of Ulster, Faculty of Informatics, *WWW pages*, <http://www.infc.ulst.ac.uk/informatics/courses/degrees.html>