EnviroInfo 2005 (Brno)
Informatics for Environmental Protection - Networking Environmental Information
 ISBN: 80-210-3780-6

Communicating Database Design: A Case Study Using the Object Role Modelling (ORM) Method

Jürgen Reinhard[1], Peter Jakob[2], Ann-Marie Jakob[3]

Abstract

After forty years of database development it is still difficult to communicate database design to project-related people who lack relevant IT knowledge. In an IT project set up to create a database application for phenology data within the overall LWF (Long Term Forest Research) project the Object Role Modelling (ORM) method was introduced in order to overcome this difficulty and to establish a common basis for communication between IT experts and all other project-related people. One of the aims of the project was to evaluate the use of the ORM method for future projects. The application and usefulness of this modelling method, which operates on the conceptual level, was to be evaluated against two main criteria, namely, whether the ORM method leads to better communication among project-related people and how easily the ORM method can be used in future projects.

1. Introduction

In 2004 the Long-Term Forest Ecosystem Research (LWF) project in Switzerland initiated an IT Project in order to establish a database application for their phenology data. Although it was a small project it was difficult to integrate the partly divergent requirements of different groups including LWF researchers and non-academic users as well as international organizations (e.g. ICP Forests - the International Co-operative Programme on Assessment and Monitoring of Air Pollution Effects on Forests) and national organizations (e.g. MeteoSwiss, the national weather service). Two sources of data with different properties had to be integrated into the database: firstly, data from MeteoSwiss and secondly, data from the LWF itself.

1.1 Phenology within the LWF

The mission of the international LWF project is to improve our understanding of how natural and anthropogenic stresses affect forest ecosystems in the long term (Cherubini and Innes 2000). A network of 17 research sites located all over Switzerland monitors the long-term response of forest ecosystem components and processes to the most relevant stress factors. The main emphasis is on atmospheric deposition, the biogeochemical cycles, the climate, the soil, the ground vegetation and the trees

Phenology is the study of the annual cycles of plants (and animals) and how they respond to seasonal changes in their environment. Within the framework of this case study only trees and shrubs were examined. Relevant phenological events in both MeteoSwiss and LWF data sources included the timing of flower emergence, sequence of bloom, fruiting, and leaf drop in autumn. Within the LWF Phenology project the extent of a specific phenological stage is estimated at a certain point in time.

[1]Fhtw-Berlin, Ostendstr. 25, D-12459 Berlin, email: reiniberl@freenet.de web: www.bui-fhtw.de
[2] Swiss Federal Research Institute WSL, Zürcherstrasse 111, CH-8903 Birmensdorf, email: peter.jakob@wsl.ch
[3] Haslenstrasse 22, CH-8903 Birmensdorf, email: amjakob@swissonline.ch

1.2 Phenology within Meteo Swiss

Founded in 1951, the Swiss Phenology Network currently has 160 research plots across Switzerland monitoring 69 different phenological stages of 26 plant species. 33 of the plots are located in woodlands and were therefore of relevance to this case study (Vasella 1997: 6). Within the Swiss Phenology Network, in contrast to the LWF, phenological events are monitored by estimating the date of occurrence of a specific phenological event (Vasella et al. 1997: 31).

2. Database Design with the Object Role Modelling Method

It was expected that the different requirements and data sources would necessitate the creation of a generic data model, which has the advantage of increased flexibility. Future requirements such as additional data-sources can often be integrated without the need for drastic structural changes to the model. On the other hand, it is difficult for project-related people without special IT knowledge to verify and validate a generic data model. In order to overcome this difficulty and to establish a better common communication basis among all project-related people, the Object Role Modelling (ORM) method was introduced (Halpin 2001:22). One of the aims of the project was to evaluate the use of the ORM method for future projects.

2.1 The Basics of Object Role Modelling (ORM)

ORM is a graphically-oriented modelling methodology which adds another abstraction layer to the data-modelling process. The common Entity Relationship (ER) approach deals with two "lower" abstraction levels, namely the logical and the physical layers of the data-modelling process. ORM allows the description of the so-called „Universe of Discourse" (UoD), meaning the section of the "real world" described by the data model, on a high conceptual level using a simple sub-set of a natural language e.g. English, German, Irish etc. Because ORM operates on a high level of abstraction (the conceptual layer) it facilitates the discussion of database design in natural language with all project-related people. By applying ORM, fundamental design faults can be eliminated or avoided in the initial stages of the design process. It also enables the IT expert to verify the proposed data model design (e.g. the completeness of applied constraints) against "real world" examples of data supplied by a domain expert.

ORM modelling forms part of Microsoft Visual Studio .net Enterprise Architect. For this case study a free stand-alone version of VisoModeler (version 3.1.c.792) was used. Historically, ORM has its origins in NIAM (Natural Language Information Analysis), which was developed in Europe in the 1970's. In ORM, the UoD is represented as objects and their interactions with each other i.e. the roles the objects play. In contrast to other modelling techniques such as the ER method, ORM does not use the concepts of entities and attributes. Instead, in ORM every attribute represented on an ER diagram is an object, which makes it possible to design constraints between them. ORM models are therefore more comprehensive than equivalent ER models, however, their resulting complexity can lead to a loss of clarity. To counteract this difficulty, it is often necessary to break down a large ORM model into smaller modules.

ORM models cannot replace ER models as they both have different functions. For example, ORM models do not show the resulting DBMS structure as the ORM notation is not intended for that purpose. It is therefore difficult to identify foreign key constraints, tables or primary keys in an ORM model. However, it is possible to directly transform an ORM model into an ER model.

2.2 ORM Development within the Context of the Case Study

Learning a new modelling method such as ORM is quite time-consuming. For the designers of this phenology application, it was often useful to transform the ORM model into a familiar ER model and discuss the results. Sometimes a reverse approach was used: first a rough ER model of one part of the UoD was designed and afterwards the same part of the UoD was designed in ORM. This proved to be useful for learning the ORM notation and also for gaining a deeper understanding of the models, both ER and ORM.

Diagram 1 illustrates a detail of the ORM model and shows the advantage of using the ORM approach. As mentioned above in Chapters 1.1 and 1.2, there are two different ways of classifying phenological events. The first method describes a phenological event by estimating when such an event will occur or has occurred. In this case, the object (attribute) “Estimated_date” has to be entered. The second method estimates to what extent a certain event has occurred by the date of the visit. Here, the objects “Pheno_extent” and “Visit_date” have to be entered. Depending on the project, one of these two estimation methods has to be applied: they are mutually exclusive and only one of them is permitted.

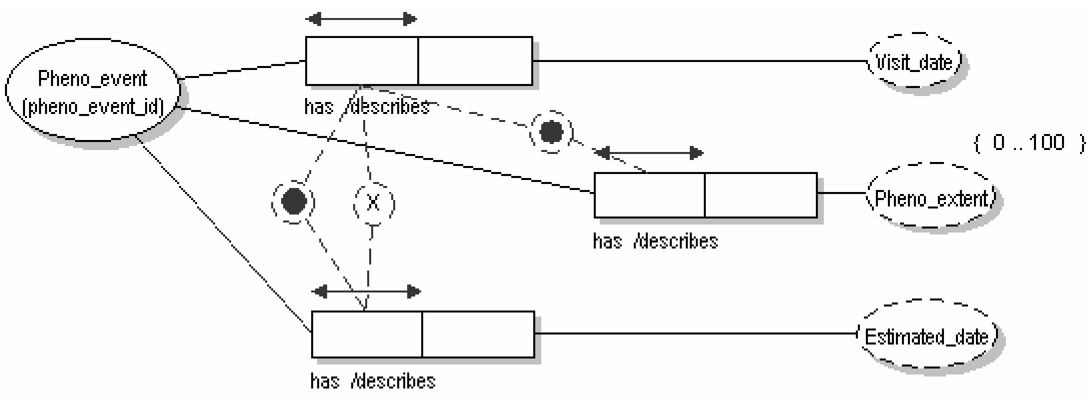

Fig. 1: Two different ways of classifying phenology events in ORM notation

The fact that at least one of the estimation methods has to be applied is shown in Figure 1 as two encircled dots with related connectors. The fact that only one of the two methods is permitted is shown by the encircled ‘X’. There is no way of representing these facts in ER notation. See Figure 2 below.

PHAENO.PHENO_EVENT	
PHENO_EVENT PHENO_EVENT_ID	NUMBER (38)
VISIT_DATE	DATE
ESTIMATED_DATE	DATE
PHENO_EXTENT	NUMBER (3)

Fig. 2: ER notation of entity “phenology event”

2.3 Communicating Database Design with ORM

A so-called “verbalizer” is part of the ORM software. This tool supports IT experts to communicate better with domain experts. Listing 1 below shows part of the document generated for the model of this case study. The verbalizer helped the database designers not to overlook and omit any relevant facts in their discussions with domain experts.

```
Pheno_event(pheno_event_id) is an entity object type.
    Every Pheno_event is identified by one distinct pheno_event_id.
    Physical Microsoft Visual FoxPro datatype: Integer.
Pheno_event has Visit_date / Visit_date describes Pheno_event
    Each Pheno_event has at most one Visit_date.
Examples:
    Pheno_event 1 has Visit_date 12.5.2005.
    Pheno_event 3 has Visit_date 12.5.2005.
Pheno_extent is a value object type.
    Physical Microsoft Visual FoxPro datatype: Numeric(3,0).
    The possible values of 'Pheno_extent' are: 0 to 100.
Pheno_event has Estimated_date / Estimated_date describes Pheno_event
    Each Pheno_event has at most one Estimated_date.
Examples:
    Pheno_event 2 has Estimated_date 2.3.2005.
    Pheno_event 5 has Estimated_date 2.3.2005.
```

Listing 1: Model description generated by the ORM Verbalizer

It was shown in Chapter 2.2 that certain facts represented in an ORM model cannot be shown in an ER model. Listing 2 below shows the check constraints generated by the ORM software. This helps the data-modeller establish a consistent data model based on the requirements worked out with the domain experts. In ER notation these rules have to be formulated separately as so-called “business-rules”.

```
-- The constraint: Each Pheno_event has some Visit_date or has some Estimated_date.
-- is enforced by the following DDL.
alter table PHENO_EVENT
      add constraint Pheno_event_MR1
       check ((VISIT_DATE is not null) or
            (ESTIMATED_DATE is not null));

-- The constraint:  No Pheno_event that has some Visit_date has some Estimated_date.
-- is enforced by the following DDL.
alter table PHENO_EVENT
      add constraint Pheno_event_excl2
       check ((VISIT_DATE is null) or
        (ESTIMATED_DATE is null));
```

Listing 2: Generated check constraints for the two different phenology events

3. Conclusion

As mentioned above, the application of ORM in this project had two aims:

1. to establish a better communication basis among project-related people.
2. to evaluate the ORM approach for use in future projects.

The first aim was partly fulfilled. One difficulty was that an ORM model of one particular UoD generally looks more complex than the equivalent ER model because it contains more symbols. For this reason, domain experts involved in the design process had to be guided through the ORM model in order to understand the detail while maintaining an overview.

ORM proved to be useful for validating the model i.e. checking the completeness of constraints with the help of sample data sets. Being able to carry out this check was especially useful when designing our abstract generic model, where there is always a danger of overlooking and omitting some fundamental fact or property of the UoD. A considerable amount of time was spent on correctly formulating the roles. The resulting model was more meaningful and readable.

In relation to the second aim, the results remained ambivalent. As mentioned above, ORM proved to be useful in many ways. However, learning a new design method with its specific notation is a time-consuming process and tight time-restrictions on projects might prevent further use of this otherwise promising method.

Bibliography

Cherubini, P. and J. L. Innes (2000): Switzerland: The Swiss Long-term Forest Ecosystem Research Programme. In: Sprott, P. (Ed.): The International Long Term Ecological Research Network. Albuquerque, N.M., U.S. Long Term Ecological Research Network Office, pp. 56-59.

Jakob P., Ginzler C., Kräuchi N., Baltensweiler A. (2000): Modern ecology requires a state-of-the-art data acquisition and management methodology. Conference of European Statisticians. UN/ECE Work Session on Methodological Issues Involving the Integration of Statistics and Geography. Neuchâtel, Switzerland, 10-12 April 2000. Working Paper No. 19, 8 p. Available online: http://www.unece.org/stats/documents/2000.04.gis.htm

Talpin, T. (2001): Information modelling and relational databases: from conceptual analysis to logical design. Academic Press San Diego USA

Talpin, T. , Evans, K., Hallock, P., Maclean, B. (2003): Database modelling with Microsoft Visio for Enterprise Architects. Elsevier USA

Vasella, A., Brügger R. (2001): Impulsprojekt für das phänologisches Monitoring im Wald und für den Einbezug in die Langfristige Waldökosystemforschung. Schlussbericht April 2001. Universität Bern, Geographisches Institut.