Chapter 6

A Method to Ease Schema Evolution

Lex Wedemeijer
ABP, The Netherlands

Maintenance on the Conceptual Schema of a database is necessary when the current data structure can’t meet changed functional requirements. An additional requirement, not expressed in the demand for change, is to minimize the impact of change. The problem of minimizing impact of change on the data is often postponed to the implementation phase when data have to be migrated into the new structure. We propose a method to address the problem of Conceptual Schema evolution in an earlier phase, and introduce the notion of Majorant Schema to support the application of the method. The advantage of the approach is that a more graceful schema evolution is ensured because a broader range of design alternatives is investigated. Other benefits are the early attention for the impact of change, and a better preparation of the data migration effort. Experiences show that the method is primarily suited to minor changes.

INTRODUCTION

Operational database applications in a dynamic business environment will be subject to continuous change. When faced with structurally new requirements, most maintenance practitioners start with determining a new Internal Schema (IS) that meets the new functional requirements. The issue of migrating the current set of data into the new schema is not often addressed at this time. Even less attention is paid to formulating the change demands on the level of the Conceptual Schema...
(CS), and to deciding on the best possible way to incorporate the change into the
CS, thereby ensuring a graceful schema evolution. At best, a quick sketch is made
that outlines the major differences between the old and new CS, indicating which
entities and relationships are affected to some extend. Going directly from old to
new IS has several disadvantages. Design choices that are conceptual in nature
are made on the wrong level of abstraction. Design alternatives may be over-
looked. It will quickly lead to a deterioration of conceptual documentation. Mini-
mal impact of change on the IS level doesn't necessarily mean a small impact of
change on the business level.

In section 2 we will propose a method that lets the designer decide on the
changes to be made on the CS, before proceeding with the Internal Schema de-
sign. In section 3 we introduce a new type of schema called the Majorant Schema
(MS). This schema can be used as the centre-piece in the method. To establish
the MS, schema integration techniques are applied in a new way. Section 4 illus-
trates the benefits of our approach by showing a simplified example, derived from
an actual business case. Some related work is discussed in section 5.

THE MAJORANT-SCHEMA METHOD

Many maintenance approaches take the current database structure as starting
point. It is our opinion that this is inadequate, because the coherence between the
current CS and the new one will be lost, and too little use is made of flexible design
solutions that are present in the old schema. We propose to solve this problem by
augmenting the common approaches with a number of design steps in order to
ensure coherence.

Our approach is depicted in Figure 1. The starting point of our method is
placed at the point where structurally new requirements are put to the mainte-
nance engineers. The method then proceeds to establish a sound CS that meets
the new requirements, then to design a new database structure.

1. The first step in applying the method is establishing what has to be changed, i.e.,
the current CS. This requires more than just taking out a copy of the original CS
design (if available) because in many business situations, the IS will have devi-
ated from the CS. While each deviation will be small, their number can add up.
To retrieve the current CS from available database information, Reverse-Engi-
neering techniques are called for (Henrard, Englebert, Hick, Roland, and Hainaut,
1998).

2. Step 2 is the creation of a Conceptual Schema that meets all new and existing
functional requirements. To a large extend, execution of this step can and will
be independent of the first one.

3. In Step 3 several preliminary activities are executed before we can proceed
with the conflict analysis:
Separation of Concerns in Model-Based Development of Distributed Real-Time Systems


www.igi-global.com/chapter/separation-concerns-model-based-development/36341?camid=4v1a