Predictive Database Schema Evolution

Hassina BOUNIF, Stefano SPACCAPIETRA

Database Laboratory (LBD)
Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland
hassina.bounif@epfl.ch, stefano.spaccapietra@epfl.ch

Abstract: This paper proposes a new approach for organizing the evolution of existing database system. This approach aims at predicting potential paths to future evolution, using a combination of different techniques such as a requirements analysis step focusing on both current and future requirements, domain ontology exploration and knowledge extraction from the web. Predictions are used to model an operational and easy-to-evolve database schema. Evolution algorithms are tailored to choose the best evolution alternative among the possible ones at any given moment.

1 Introduction

Existing database systems show little flexibility in terms of supporting changes to the organization of the data. Yet, database-centered information systems need to evolve. A first reason is that real world changes and the information system has to keep in tune with it. Another organizational reason for evolution may be that the perception of the real world has changed, or the part of the real world that is of interest expands, because of new user or application requirements have arised. Finally, there are also technological reasons leading to database evolution, such as changes in the functionality of the supporting computer infrastructure or internal system reorganization tailored for performance improvement.

2 Existing Approaches

Schema evolution has been extensively addressed in the past and a variety of techniques have been proposed to perform evolution in the smoothest possible way without disrupting the continued operation of the database. The proposed solutions can be categorized as mainly following one of the following approaches:

- Modification: The old schema and its corresponding data are replaced by the new schema and its new data. This may lead to information loss (e.g., some existing data is not relevant anymore in the new setting and is therefore trashed) and inability for applications using the old schema to use the new database [BKK87]. This makes the approach undesirable in most real cases, yet is the one that is most popular with existing DBMS.
Versioning: The old schema and its corresponding data are preserved (and continue to be used by existing applications), but a new version of the schema is created that incorporates the desired changes [KC88]. Many different techniques exist to organize schema versions that stem from a sequence of evolutions, to cope with how applications can access the data that is relevant to them in a versioned schema environment, and to progressively transform existing data to meet the evolved specifications.

3 Motivating Application

This work is carried out within the IM2\(^1\) research project that focuses on interactive and multimodal access to multimedia and meta-information. Project results are to be demonstrated using as application framework that consists of meeting (audio and video) recordings that are stored and annotated in multiple ways. Objectives of the project include developing functionality to acquire media data from smart meeting rooms (rooms specially equipped for audio-video recording of discussions and presentations at meetings), meta-data defining meeting recordings, and annotation data produced by different research groups (on speech recognition, speech transcription, video analysis, dialog acts analysis, document analysis, etc.).

The need for schema evolution support in this context is crucial because the research groups in the project develop a wide variety of annotations to characterize aspects of interest to them regarding the content of multimedia data. The format of the definition of annotation types and values is likely to change frequently during the project, as the type of annotations that will be used is unpredictable and continuously evolves to catch up with results from ongoing research and shifting interest level in topical issues the groups are addressing. Each of these changes can trigger the need for evolving the schema of the IM2 database.

4 The Proposed Approach

Departing from traditional approaches, we aim at taking a new perspective at the problem of schema evolution. Our goal is to prepare for evolution by anticipating the changes before they occur. The process is divided into three phases (requirements analysis, modeling and planning for evolution, evolution).

4.1 Requirements analysis

As in traditional information system design methodology, we assume that to start with a detailed requirement analysis is performed, involving the IS designers as well as users and application managers. We are not interested in the technique used for this task, but

\(^1\) Interactive Multimodal Information Management (http://www.im2.ch), Swiss National Competence Centre in Research (NCCR), supported by the Swiss National Research Fund
require that the resulting output be defined in terms of an ontology of requirements, such as in e.g. [Gu98].

The difference with traditional methodologies is that we explicitly include future requirements in the analysis to be performed. Future requirements may be partly characterized using the same techniques as for current requirements, i.e. based on the knowledge within the organization. They can also be investigated using other techniques that explore knowledge external to the organization. Relevant external knowledge may be found in domain ontologies that at least to some extent describe the activity domain of the organization. Similarly, web pages somehow connected to the activity domain are also worth being explored. For instance using knowledge extraction and data mining techniques.

4.2 Modeling and planning for evolution

In this phase, requirement analysis is turned into formal data structure specifications. First, the database designer defines the initial database schema that translates current requirements. This schema supports database operation till the next evolution phase. Second, the database designer defines one or more “evolution schemas”, each one translating a subset of the future requirements that forms a homogeneous evolution step. It is expected that several evolution schemas be defined as evolution can follow different paths, corresponding to different evolution strategies of the organization. We intuitively regard each evolution schema as representing a consistent unit of evolution. If alternative evolution paths exist achieving the same evolution process, we plan building cost models that would enable choosing the best evolution strategy with respect to the situation that actually exists at the time evolution has to be performed.

4.3 Evolution

The evolution phase implements the decision to make the database evolve. An essential input is the definition of the goal to be achieved through evolution, hereinafter stated as an updated description of the reality of interest to the organization database. We sketch below an iterative algorithm that considers two alternative situations and assumes the history of data evolution is kept. The alternative situations are

- **Pre-planned simple evolution**: the evolution at hand (representing the new reality) matched one of the evolution schemas. If there are several candidate sub-schemas, the best one is chosen according to cost model. Changes to the existing database are performed according to the selected evolution schema. This is illustrated in figure 1.
• **Pre-planned complex evolution**: a number of evolution schemas together fulfil the requirements from the new reality. These schemas are then merged before performing changes to the existing database. This is illustrated in figure 2.

**Conclusion**

This work is currently in the very preliminary stage. The methodology is sketched, the next step will be to precisely define the content of information exchanges between the different techniques. A parallel effort will determine which strategy for unplanned evolution is appropriate in our context.

**References**

