

## Chapter XII

# Evolutionary Database: State of the Art and Issues

**Vincenzo Deufemia**

*Università di Salerno, Italy*

**Giuseppe Polese**

*Università di Salerno, Italy*

**Mario Vacca**

*Università di Salerno, Italy*

### INTRODUCTION

Waterfall methodologies can poorly cope with changes, making maintenance considerably an expensive process. For this reason, incremental and iterative methodologies were introduced (Larman & Basili, 2003). They view system development as a step-by-step process, with the introduction of new functionalities to meet user needs. The main problem arising in both paradigms is the complexity in facing changes. Therefore, an increased automated support in this task would result in a reduction of efforts and costs, especially

in incremental methodologies, because it would make them more systematic.

Changes are often necessary to reflect the continuous evolution of the real world, which causes frequent changes in functional requirements. This entails frequent modifications to the software, yielding a gradual decay of its overall quality. For this reason, many researchers in this field have developed software refactoring techniques (Mens & Tourwé, 2004). Software refactoring is intended as the restructuring of an existing body of code, aiming to alter its internal structure without changing its external behavior. It consists of a

series of small behavior preserving transformations, which altogether can produce a significant software structural change. Moreover, system modifications resulting in changes to database structure are also relatively frequent (Roddick, 1995). These changes are particularly critical, since they affect not only the data, but also the application programs relying on them (Ambler & Sadalage, 2006; Karahasanovic, 2001).

Several disciplines have faced the problem of managing the effects of database schema changes. The interest in this topic has consistently grown, as shown by several surveys and bibliographies recently published (Roddick, 1992; Roddick, 1995; Li, 1999; Rahm & Bernstein, 2006). In particular, *schema modification* has faced the problem of changing the schema of a populated database. In addition to this, *schema evolution* pursues the same goal, but it tries to avoid loss of data. Alternatively, *schema versioning* performs modifications of the schema, but it keeps old versions to preserve existing queries and application programs running on it. Although schema versioning faces the problem of query and application programs preservation, it considerably increases the complexity and the overhead of the underlying DBMS. Finally, *database refactoring* aims to modify the database schema, and to change the corresponding application programs accordingly. In other words, the database refactoring is the process of slowly growing a database, modifying the schema by small steps, and propagating the changes to the queries. This is considered the major technique in the development of evolutionary database (Ambler & Sadalage, 2006).

With the introduction of evolutionary databases methodologies, the research area of schema evolution and versioning has been naturally broadening to embody new and more challenging research problems. In this chapter, borrowing the term from Ambler et al. (2006), we call this new research area *Evolutionary Database*.

This chapter discusses the main issues concerning evolutionary database and then we

survey several models and tools proposed for their solution.

## BACKGROUND

Schema evolution and schema versioning have been pioneer research areas facing problems due to the introduction of changes in database schemas.

Generally accepted definitions of schema evolution and schema versioning are provided in a paper by Roddick (Roddick, 1995), which can be considered as a sort of manifesto for this research topic. According to these definitions, the aim of schema evolution is to facilitate changes in the database schema without losing existing data. If these changes are seen as producing different versions, then the need to store the data of all the versions naturally arises. Therefore, the goal of schema versioning is to allow users to access the data of old versions (retrospective access) as well as those of new ones (prospective access).

Nowadays, researchers of the database area agree that schema evolution and versioning yield two main issues: *semantics of changes* and *change propagation* (see, for example, Peters, et al. 1997 or Franconi, et al., 2000). The first problem requires determining the effects of changes on the schema, whereas the second deals with the effects on the data.

### Example 1

Let us consider the database of the employees of a University described by the schema *S* represented by the relation

**R**(*Employee\_ID*, *Last Name*, *First name*, *Department\_ID*, *Salary*, *Address*)

where *Employee\_ID* is the primary key. Let us suppose that it is required that the database be in third normal form (Elmasri & Navathe, 2006). The addition of the functional dependency

6 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/evolutionary-database-state-art-issues/20693](http://www.igi-global.com/chapter/evolutionary-database-state-art-issues/20693)

## Related Content

---

### Practical Case Study of a Web-Based Tutor Payment System

Tanguy Chateau, Cecile Leroy, Johanna W. Rahayuand David Taniar (2003). *Web-Powered Databases* (pp. 88-120).

[www.irma-international.org/chapter/practical-case-study-web-based/31425](http://www.irma-international.org/chapter/practical-case-study-web-based/31425)

### Semi-Supervised Event Extraction Incorporated With Topic Event Frame

Gongqing Wu, Zhuochun Miao, Shengjie Hu, Yinghuan Wang, Zan Zhangand Xianyu Bao (2023). *Journal of Database Management* (pp. 1-26).

[www.irma-international.org/article/semi-supervised-event-extraction-incorporated-with-topic-event-frame/318453](http://www.irma-international.org/article/semi-supervised-event-extraction-incorporated-with-topic-event-frame/318453)

### Processing Ill-defined Spatial Operators in Geographic Databases: A Common-sense Approach

Ramesh Subramanian (1999). *Journal of Database Management* (pp. 19-35).

[www.irma-international.org/article/processing-ill-defined-spatial-operators/51210](http://www.irma-international.org/article/processing-ill-defined-spatial-operators/51210)

### An Approach for Evolution-Driven Method Engineering

Jolita Ralyte, Colette Rollandand Mohamed Ben Ayed (2005). *Information Modeling Methods and Methodologies: Advanced Topics in Database Research* (pp. 80-101).

[www.irma-international.org/chapter/approach-evolution-driven-method-engineering/23010](http://www.irma-international.org/chapter/approach-evolution-driven-method-engineering/23010)

### How to Achieve Fuzzy Relational Databases Managing Fuzzy Data and Metadata

Mohamed Ali Ben Hassine, Amel Grissa Touzi, José Galindoand Habib Ounelli (2008). *Handbook of Research on Fuzzy Information Processing in Databases* (pp. 351-380).

[www.irma-international.org/chapter/achieve-fuzzy-relational-databases-managing/20360](http://www.irma-international.org/chapter/achieve-fuzzy-relational-databases-managing/20360)