

An Ontology-based Approach for Database Evolution

N. Lammari*, J. Akoka**, I. Comyn-Wattiau***

* Laboratoire CEDRIC-CNAM, 292 Rue Saint Martin, 75141 Paris Cedex 03, France

** Laboratoire CEDRIC-CNAM et INT

*** Laboratoire CEDRIC-CNAM et ESSEC

{lammari,akoka,wattiau}@cnam.fr

Abstract: Many research efforts have led to robust database initial design methodologies. However, database evolutions suffer from lack of structured approaches. Using high-level conceptualization models like ontologies, we describe an original approach to database evolution management. Our aim is to help the maintenance engineer in 1) characterizing the evolution, 2) finding the best techniques and tools to deal with this evolution. Our guidance approach provides this help by linking formally a change ontology and a maintenance method ontology through an algorithmic approach. A historization process of evolutions enriches the set of information delivered to the maintenance engineer. Our future work consists in prototyping this approach.

1. Introduction

Recent studies have mentioned that 50% to 70% of companies software budget are absorbed by maintenance and evolution activities [CMW02]. Moreover, 80% of information system staff's time and effort are devoted to maintenance [ST00]. We claim that these costs are highly due to the lack of methodology guiding designers in unanticipated database and software evolution management. Since most of these softwares are based on database systems, we take a particular interest in the evolution of database systems.

The present paper synthesizes a work in progress dealing with database change management. We propose an ontology-based approach allowing us to guide maintenance engineers in the process of choosing relevant approaches, strategies, techniques and tools to handle database application changes. In order to perform such a choice, we provide:

- An ontology of change that categorizes all evolutions that may occur in a database.
- An ontology of evolution methods that organizes approaches, strategies, techniques and tools described in the literature, developed as research demonstration tools, and/or commercial tools.
- An algorithm for the mapping between the two ontologies in order to select the relevant methods for a characterized change
- A history database structure enabling recording of past changes characterized by cost, duration and other specific data.

The approach is enriched by a process for querying and updating this change history database.

The remainder of the paper is organized as follows. The next section presents the two ontologies. Section 3 describes the guiding process. Section 4 concludes by presenting future works.

2. The ontologies

We address the problem of database systems evolution by providing the designer with a guidance approach for managing evolution. This approach exploits two ontologies. The first one describes the changes occurring in a database (e.g.: Fig 2.1). It can be considered as an explicit conceptualization of changes. Its purpose is to provide information regarding the characteristics of database change concepts, and to compare two concepts to determine if they might be related. This ontology classifies an evolution (or a change) into one or more categories depending on the significance of change, its semantic preservation, the information and application dependency, etc. Sub-nodes are connected via ISA links as indicated by the arrows. Additional links are used: "may generate" to list potential changes that can be generated by a change, "is synonym of" to unify concept naming, and "is instance of" that associates an instance list with a given change.

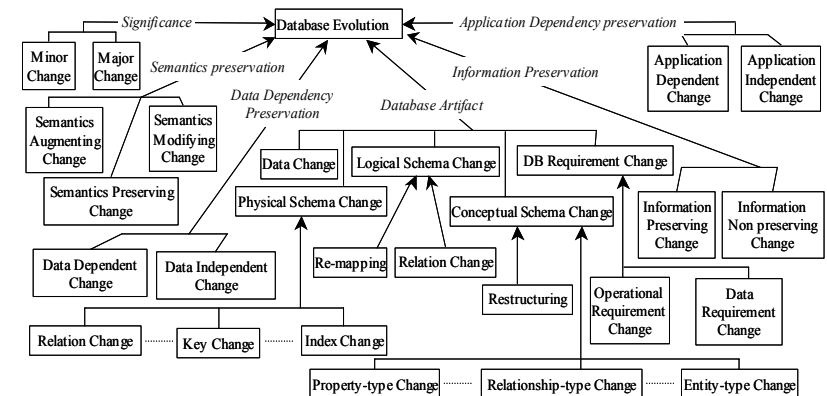


Fig 2.1: The ontology of database changes

The second ontology describes evolution methods useful for database change management (e.g.: Fig 2.2). It organizes approaches, strategies, techniques and tools described in the literature, including research demonstration tools, and/or commercial tools. As for the ontology of change, it is also described by a semantic network where links such as IS-A, "is instance of", "can be applied for", etc. are represented.

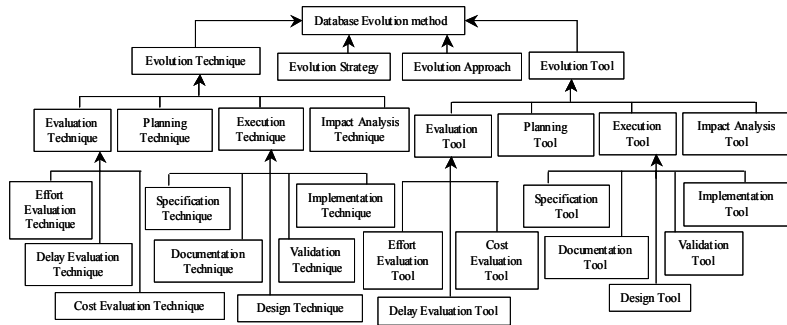


Fig 2.2: The ontology of database evolution methods

3. The guidance approach

The general architecture of the guidance approach is composed of four phases (e.g.: Fig 3.1):

- The first phase consists in characterizing the change being considered, such as a business rule modification leading to a change in the logical schema and even in the code. The characterization allows us to locate the change within the first ontology. The internal representation of this step is a vector V. Each component of the vector defines the correct location of the change in each specification of the ontology.
- The second phase performs a search to confrontate the vector V with the second ontology. This vector is associated with a set S of approaches, techniques, strategies and tools by means of "can be applied for" links.
- During the third phase, the maintenance engineer, guided by the approach, compares the proposed techniques and tools contained in S to the ones that have already been used, by checking the change history database. The latter describes numeric values characterizing these changes, and data about cost, duration needed to implement each change and quality obtained. At the end of this phase the maintenance engineer performs the choice inside the list, by confronting the current change with the change history.
- In the fourth phase, the maintenance engineer implements the change.
- The last phase memorizes data about this implementation in the history database.

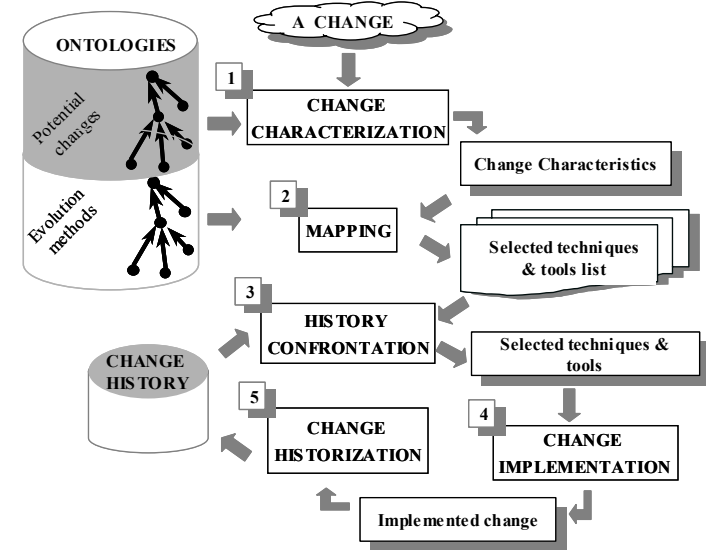


Fig 3.1: The guidance approach

4. Conclusion

The paper presents an ontology-based approach for database evolutions that is under development. This work aims, through our proposition of the two ontologies (ontology of database changes and ontology of database evolution methods), to the unification of existing works [BLN86], [CAL03], [Ch01], [DBK98], [Fe03], [KS96], [Ki95], [LS82], [Me03]. It also aims to provide maintenance teams with realistic experimental database evolution environments. It consists of a flexible approach leading to an effective and efficient management of database evolution.

This work can also be extended to take into account not only database evolution but, more generally, software evolution.

Our future goal is the development of a guidance tool based on this approach to support the database evolution. The experimentation of the tool will allow us to validate and / or improve our ontologies.

References

- [BC86] Biskup J., Convent B.: A Formal View Integration Method. Proceedings of ACM SIGMOD International Conference on Management of Data, Washington, 1986.
- [BLN86] Batini C., Lenzerini M., Navathe S.: A comparative Analysis of methodologies for Database Schema Integration. ACM Computing Surveys Journal 18 (4), 1986.
- [CAL03] Comyn-Wattiau I, Akoka J, Lammari N, A Framework for Database Evolution Management, USE03, 2nd International Workshop on Unanticipated Software Evolution, in conjunction with ETAPS03, Warsaw, 2003.
- [CMW02] Cremer K., Marburger A. and Westfechtel B., Graph-Based Tools for Re-engineering. Journal of Software Maintenance and Evolution: Research and Practice, Vol 14, 2002, pp 257-292.
- [Ch01] Chapin N, Hale J, Khan K, Ramil J, Than W.G.: Types of Software Evolution and Software Maintenance, Journal of Software Maintenance and Evolution, 2001.
- [DBK98] Davidson S., Buneman P., Kosky A.: Semantics of Database Transformations. LNCS 1358, Edited by L. Libkin and B. Thalheim, 1998.
- [Fe03] Felici M: Taxonomy of Evolution and Dependability, USE03, 2nd International Workshop on Unanticipated Software Evolution, in conjunction with ETAPS03, Warsaw, 2003.
- [KS96] Kashyap V., Sheth A.: Semantic and Schematic Similarities Between DataBase Objects: A context-based Approach. VLDB Journal, 5 (4), 1996.
- [Ki95] Kim W., Choi I., Gala S., Scheeval M.: On Resolving Schematic Heterogeneity in Multidatabase Systems. Modern Database Systems, ACM Press, 1995.
- [LS82] Lientz, Swanson, Distributed Systems Development and Maintenance. In Management of Distributed Data Processing, Akoka, J. (ed.), North-Holland, Publishing Company, Amsterdam, 1982.
- [MIR94] Miller R. J., Ioannidis Y. E., Ramakrishan R.: Schema Equivalence in Heterogeneous Systems: Bridging Theory and Practice. Information System Journal, 19 (1), 1994.
- [Me03] Mens T., Buckley J, Zenger M, Rashid A.: Towards a Taxonomy of Software Evolution, USE03, 2nd International Workshop on Unanticipated Software Evolution, in conjunction with ETAPS03, Warsaw, 2003.
- [ST00] Schach S. R. and Tomer A., A Maintenance-oriented Approach to Software Construction. Journal of Software Maintenance and Evolution: Research and Practice, Vol 12, 2000, pp 25-45.