An Online Bibliography on Schema Evolution

Erhard Rahm
University of Leipzig
rahm@informatik.uni-leipzig.de

Philip A. Bernstein
Microsoft Corporation
philbe@microsoft.com

Abstract

We briefly motivate and present a new online bibliography on schema evolution, an area which has recently gained much interest in both research and practice.

1 Introduction

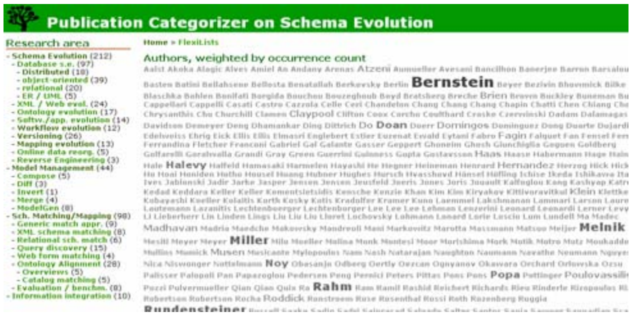
Schema evolution is the ability to change deployed schemas, i.e., metadata structures formally describing complex artifacts such as databases, messages, application programs or workflows. Typical schemas thus include relational or object-oriented (OO) database schemas, conceptual ER or UML models, ontologies, XML schemas, software interfaces and workflow specifications. Obviously, the need for schema evolution occurs very often in order to deal with new or changed requirements, to correct deficiencies in the current schemas or to migrate to a new platform.

Effective support for schema evolution is challenging since schema changes may have to be propagated, correctly and efficiently, to instance data, views, applications and other dependent system components. Ideally, dealing with these changes should require little manual work and system unavailability. For instance, changes to a database schema S should be propagated to instances data and views defined on S with minimal human intervention.

Schema evolution has been an active research area for a long time. However, the need for powerful schema evolution has been increasing and many papers have appeared recently. Moreover, commercial database systems such as IBM DB2, Oracle and Microsoft SQL Server have started to support online schema evolution capabilities. Thus, although several previous bibliographies and surveys exist [13, 14], there is value in producing an up-to-date bibliography.

One reason for increased interest is that widespread use of XML and web services has led to new schema types and usage scenarios of schemas for which schema evolution must be supported. For example, data integration architectures, such as enterprise information integration (EII) and enterprise application integration (EAI), are now common and must be able to deal with schema changes for data sources, global schemas and ontologies. The growing importance of such problems has prompted recent work on generic metadata management, e.g., schema matching and model management. Such approaches can help automate schema evolution tasks, e.g., generation and adaptation of mappings between schemas. The goal of our online bibliography is to provide a comprehensive and up-to-date collection of publications on schema evolution. We are not limiting ourselves to database schema evolution but also consider related fields such as ontology evolution, software evolution and workflow evolution. We have found these evolution problems are often similar so that proposed solutions may be transferable to different fields. For example, ontology evolution has similarities to previously investigated evolution problems in object-oriented database systems.

Our bibliography on schema evolution is accessible on the web under **http://se-pubs.dbs.uni-leipzig.de**. We use our content categorization tool Caravela [1] to categorize publications along multiple hierarchical dimensions. Bibliographic entries typically contain the abstract, full-text link and current number of citations from Google Scholar. The system provides many ways to search and browse the categories and papers. Publication entries can be added, corrected and categorized collaboratively (wiki-like) by many users. Automated data import from files or web sites like Google scholar is also supported for authorized users. Fig. 1 shows the current start page of the bibliography indicating the categories (on the left) and author names sized according to the number of available papers.



Publication Categorizer on Schema Evolution

Research area

- Schema Evolution (212)
  - Database s.e. (97)
    - Distributed (18)
    - object-oriented (39)
    - relational (20)
    - ER / UML (5)
  - XML / Web evol. (24)
  - Ontology evolution (17)
  - Softw./app. evolution (14)
  - Workflow evolution (12)
  - Versioning (26)
  - Mapping evolution (13)
  - Online data reorg. (5)
  - Reverse Engineering (3)
- Model Management (44)
  - Compose (5)
  - Diff (3)
  - Invert (1)
  - Merge (4)
  - ModelGen (8)
- Sch. Matching/Mapping (98)
  - Generic match appr. (9)
  - XML schema matching (8)
  - Relational sch. match (6)
  - Query discovery (15)
  - Web form matching (4)
  - Ontology Alignment (28)
    - Overviews (5)
    - Catalog matching (5)
  - Evaluation / benchm. (8)
- Information integration (10)

Home » FlexiLists

Authors, weighted by occurrence count

Aalst Akoka Alagic Alves Amiel An Andany Arenas Atzeni Aumueller Avesani Bancilhon Banerjee Barron Barsalou
Basten Batini Bellahsene Bellosta Benatallah Berkovsky Berlin Bernstein Beyer Bezivin Bhowmick Bilke
Blaschka Bohlen Bonifati Borgida Bouchou Bouzeghoub Boyd Bratsberg Breche Brien Brown Buckley Buneman
Cappellari Cappelli Casati Castro Cazzola Celle Ceri Chandelon Chang Chang Chang Chapin Chatti Chen Chiang
Chrysanthis Chu Churchill Clamen Claypool Clifton Coox Corcho Coulthard Croske Czerwinski Dadam Dalamagas
Davidson Demeyer Deng Dhamankar Ding Dittrich Do Doan Doerr Domingos Dominguez Dong Duarte Dujardin
Edelweiss Ehrig Eick Ellis Ellis Elmasri Englebert Estier Euzenat Ewald Eytani Fabro Fagin Falquet Fan Fensel
Ferrandina Fletcher Franconi Gabriel Gal Galante Gasser Geppert Ghoneim Ghosh Giunchiglia Gegoen Goldberg
Golfarelli Goralwalla Grandi Gray Green Guerrini Guinness Gupta Gustavsson Haas Haase Habermann Hage
Hale Halevy Halfeld Hamasaki Harmelen Hayashi He Hegner Heineman Henrard Hernandez Herzog Hick
Hu Hoai Honiden Hutho Housel Huang Hubner Hughes Hursch Hvasshovd Hänsel Höfling Ichise Ikeda Ishikawa
Ives Jablonski Jadir Jarke Jasper Jensen Jensen Jeusfeld Jeeris Jones Joris Jouault Kalfoglou Kang Kashyap
Kedad Keddara Keller Keller Komentsietsidis Kensche Kenzie Khan Kim Kim Kiryakov Kittivoravitkul Klein Klettke
Kobayashi Koeller Kolaitis Kurth Kosky Katis Kradolfer Kramer Kuno Laemmel Lakshmanan Lammari Larson
Lautemann Lazanitis Lechtenboerger Lechtenborger Lee Lee Lee Lehman Lenzerini Leonard Leonardi Lerner Levy
Li Lieberherr Lin Linden Lings Liu Liu Liu Lloret Lochovsky Lohmann Lanard Lorie Lesciu Lum Lundell Ma Madec
Madhavan Madria Maedche Makovsky Mandreoli Mani Markovitz Marotta Massmann Matsuo Meijer Melnik
Mesiti Meyer Meyer Miller Milo Mueller Molina Monk Montesi Moor Morishima Mork Motik Motro Mutz
Mullins Munick Musen Musicante Mylopoulos Nam Nash Natarajan Naughton Naumann Navathe Neumann
Nica Niswonger Nottelmann Noy Obasanjo Odberg Oertly Oezcan Ognyanov Okawara Orchard Orlowska Ozsu
Palisser Palopoli Pan Papazoglou Pedersen Peng Pernici Peters Pittas Pons Pons Popa Pottinger Poulovassilis
Pozzi Pulvermueller Qian Qian Quix Ra Rahm Ram Ramil Rashid Reichert Richards Rieu Rinderle Rizopoulos
Robertson Robertson Rocha Roddick Runstroem Ruse Rusenthal Rossi Roth Rozenberg Ruggia
Rundensteiner

As of October 2006 more than 300 papers on schema evolution and related fields are categorized. We broadly assign papers within the following categories:

- Database schema evolution
- XML schema evolution
- Ontology evolution
- Software evolution and
- Workflow evolution.

Furthermore we have separate categories for popular solution approaches, such as

- Schema versioning and
- Model and mapping management.

Each category is usually divided into several subcategories. Using our categorization tool these categories can easily be extended or refined as needed, i.e., we support evolution of the categorization schema. In the following we briefly list some specific aspects of the research categories covered by the bibliography.

2 Research categories

2.1 Database schema evolution

Some papers characterize types of schema changes for different data models, in particular relational, object-oriented (OO) and XML databases. These changes can be propagated to instances of the schema immediately or lazily. They may also be propagated to dependent views, something that today's commercial database systems do automatically for simpler schema changes (e.g., 1 table).

There are many papers on schema evolution for OO database systems [2,11], since evolution is intrinsic to the design processes they support. By contrast, there are still relatively few papers on schema evolution in distributed systems—possibly a good opportunity for future research.

2.2 XML schema evolution

The semi-structured nature of XML offers more flexibility in coping with schema changes and lower cost, due to such features as optionality of schema parts and multiple schemas per database [4].

2.3 Ontology evolution

Ontologies exhibit the same evolution problems as database schemas, but have some different constructs, such as controlled vocabularies, taxonomies, and rule-based knowledge representation, and hence have some different types of changes. Often, an ontology contains both schema-like conceptual metadata plus its instances; changes to metadata and instances need to be considered together. A domain ontology may be used in many applications, resulting in dependencies between distributed systems. So far, most papers focus on ontology matching and versioning aspects of evolution [6, 7].

2.4 Software evolution

The generation of a new software version shares many of the problems of schema evolution. Instead of schemas, we have program interfaces or class hierarchies. Instead of mappings or views, we have usage relationships and dependencies between program modules. Research papers classify different software evolution and maintenance scenarios [8]. Many papers focus especially on object-oriented software development [9]. Some describe change support tools.

2.5 Workflow evolution

Workflows are long-running activities. Instead of schemas and databases, we have workflow specifications and executing workflow instances. So changing a workflow specification (e.g., change/add/drop an activity) requires different actions than changing a database schema [5].

2.6 Version management

One major approach to schema evolution is the use of user-controlled, explicit versions [7, 14]. For example, the need to propagate changes is reduced by preserving older versions of schemas. Although versioning is rarely used for database schema evolution, it is a very common approach to software evolution and will likely be important for XML, web service, and ontology evolution.

2.7 Model and mapping management

High-level operators on schemas and mappings are useful for generating views and other mappings and adapting them after schema changes.

- There is a big literature on schema matching [12], which can help determine what has changed. Schema evolution is a simple case for schema matching since most of the schema remains unchanged.
- Given a result from schema matching, there are query discovery techniques [10] to generate an executable (instance-level) mapping between the old and evolved schema.
- Given a mapping from an evolved schema to the old schema and an existing view over the old schema, mapping composition can be used to produce an updated view [15].
- Scripts of match, compose and other operators have been published for a variety of complex schema evolution scenarios [3].

3 References

1. Aumüller, D., Rahm, E.: Caravela: Semantic Content Management with Automatic Categorization. Univ. of Leipzig, 2006
2. Banerjee, J.; Kim, W.; Kim, H.; Korth, H. F. Semantics and Implementation of Schema Evolution in Object-Oriented Databases. *Proc. SIGMOD 1987*
3. Bernstein, P.A.: Applying Model Management to Classical Meta Data Problems. *Proc. CIDR 2003*
4. Beyer, K.; Oezcan, F.; Saiprasad, S.; Van der Linden, B.: DB2/XML: Designing for Evolution. *Proc. SIGMOD 2005.*
5. Casati, I.; Ceri, S.; Pernici, B.; Pozzi, G. Workflow Evolution. *Proc. Int. Conf. on Conceptual Modeling (ER),* 1996
6. Doan, A.; Madhavan, J.; Domingos, P.; Halevy, A.: Learning to Map between Ontologies on the Semantic Web. Proc. *WWW2002*
7. Klein, M.; Fensel, D.: Ontology Versioning on the Semantic Web. *Proc. Int. Semantic Web Working Symposium*, 2001
8. Lehman, M.M; Ramil, J.F.: Software Evolution - Background, Theory, Practice. *Inf. Process. Lett.* 88(1-2), 2003
9. Lieberherr, K.J., Xiao, C.: Object-Oriented Software Evolution. *IEEE Trans. Software Eng.* 19(4), 1993
10. Miller, R.; Haas, L.; Hernandez, M.: Schema Mapping as Query Discovery. *Proc. 26th VLDB*, 2000
11. Ra, Y; Rundensteiner, E.: A Transparent Schema-Evolution System Based on Object-Oriented View Technology. *IEEE Trans. Knowledge and Data Eng 9(4),* 1997
12. Rahm, E.; Bernstein, P. A.: A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, 2001
13. Roddick, J.F.: Schema Evolution in Database Systems: An Annotated Bibliography. *SIGMOD Record 21(4),* 1992
14. Roddick, J.F.: Survey of Schema Versioning Issues for Database Systems. *Information and Software Technology*, 37(7), 1995
15. Yu, C.; Popa, L.: Semantic Adaptation of Schema Mappings when Schemas Evolve. *Proc. VLDB 2005*