Schema Evolution in Database Systems - An Annotated Bibliography

John F. Roddick

School of Computer and Information Science,
University of South Australia,
The Levels, SA 5095, South Australia

Roddick@UniSA.edu.AU

Abstract

Schema Evolution is the ability of a database system to respond to changes in the real world by allowing the schema to evolve. In many systems this property also implies a retaining of past states of the schema. This latter property is necessary if data recorded during the lifetime of one version of the schema is not to be made obsolete as the schema changes. This annotated bibliography investigates current published research with respect to the handling of changing schemas in database systems.

Introduction

Most database systems at some time or another require a change to their schema, due to either changes in the real world, a change in the application requirements or mistakes during systems analysis or design. When these changes occur database systems must provide schema manipulation tools with which the database administrator can modify the database. In many systems available commercially however, the database administrator must also make decisions on whether the data already held in the database is valid given the new schema. In many cases data is either deleted unnecessarily, misleadingly left in the database or the schema is made unnecessarily complicated by the retention of obsolete attributes.

As an example, consider a salary relation which holds the following fields:

Staff Id	Position Code	Salary
21677	G55	$33,000
21678	G56	$37,000
21680	A05	$45,500
21683	A09	$65,400
21687	G51	$32,000

Suppose also that the position codes currently extant are to be replaced with new codes based on new domains, for example, a position code based entirely on a domain of four-digit numeric integers.

The database administrator has significant problems arising from the retention of the current data such as:

i. is the position code attribute to be defined as alphanumeric despite the new position codes being numeric?

ii. is another field required to store the old codes, if so for how long do we retain this field?

iii. what about position histories and retired employees?

This is one of the simpler cases for which a change to the schema, in this case simply a domain change for an attribute, has resulted in semantic problems for the extant data. In cases like this it is clear that if the schema could evolve we could retain the old data as being applicable to an old schema definition and store new data under a new schema definition. More complex schema reorganisations (the deletion of a relation or an amendment to the class lattice) are

accompanied by more severe schema evolution problems.

Over the last few years research has started to investigate this problem and this bibliography looks at some of this work. Most of the works included here have an associated annotation. The work has been categorised broadly into three areas:

i. Those dealing with the relational model or its derivatives such as the NF^2 model;

ii. Those approaching schema evolution from an object-oriented perspective;

iii. Other miscellaneous work not categorised easily above.

Within each area the work is arranged alphabetically by first author. Every effort has been made to include all relevant research work in this area but inevitably some work will have been overlooked. For this we ask the readers forbearance.

The file `pub/bib/se.bib` containing these references can be obtained by anonymous ftp from `lux.levels.unisa.edu.au`. The author would welcome relevant additions which will be added to the file.

Schema Evolution and the Relational Data Model

Andany, J., Leonard, M. and Palisser, C. (1991) : *Management of schema evolution in databases,* 17th International Conference on Very Large Data Bases (VLDB), pp. 161-170.

Ariav, G. (1991) : *Temporally oriented data definitions: managing schema evolution in temporally oriented databases,* Data and Knowledge Engineering, Vol. 6, pp. 451-467.

This paper looks at a number of important issues pertaining to *Temporally Oriented Data Definition (TODD).* In particular, the author builds on previous temporal data modelling research and applies his cubic metaphor to the understanding of the complexities associated with schema evolution in order to present a *comprehensive TODM.* A discussion of schema change activities and their affect on the preservation of existing data and the circumstances that require an update to the application is given. The last section is a discussion of emerging issues proposed by the research so far in schema evolution and discusses in more detail a few of the papers listed here.

Clifford, J. and Croker, A. (1987) : *The historical relational data model (HRDM) and algebra based on lifespans,* 3rd IEEE International Conference on Data Engineering, Los Angeles, CA, IEEE Computer Society Press, pp. 528-537.

This paper presents an algebra based around the idea that database objects, both data and schema, have defined periods of applicability. An *historical relational data model (HDRM)* is proposed as an extension to the relational data model.

Dadam, P. and Teuhola, J. (1987) : *Managing schema versions in a time-versioned non-first-normal-form relational database.* Technical Report 87.01.001, IBM Heidelberg Scientific Center, Germany.

Dadam, P. and Teuhola, J. (1987) : *Managing schema versions in a time-versioned non-first-normal-form relational database,* Datenbanksysteme in Buro, Technik und Wissenschaft, Darmstadt, West Germany, Springer-Verlag, pp. 161-179, in German.

These papers propose mechanisms for the incorporation of schema evolution into NF^2 databases. A number of types of schema changes are examined and implementation proposals suggested. See also Ariav (1991) for a discussion on this paper.

Marinos, L., Papazoglou, M. P. and Norrie, M. (1988) : *Towards the design of an integrated environment for distributed databases.* in *Parallel Processing and Applications.* E. Chiricozzi and A. D'Amico (ed.), Elsevier Science Publishers B.V. (North-Holland), pp. 283-288.

The primary aim of this paper is to present the design issues for the INTENT

distributed DBMS. The system is based around an *Extended Semantic Data Model* or *ESDM* which in turn is based on RM/T. While much of the paper deals with DDBMS issues, also discussed are questions relating to schema evolution within a distributed architecture.

McKenzie, E. and Snodgrass, R. (1987) : *Scheme evolution and the relational algebra.* Technical Report 87-003, Department of Computer Science, University of North Carolina, Chapel Hill, NC.

McKenzie, E. and Snodgrass, R. (1990) : *Schema evolution and the relational algebra,* Information Systems, Vol. 15, No. 2, pp. 207-232.

These papers extend the work done on temporal databases by the authors and presents a query and update algebra for databases with temporal and schema evolution support. The latter is regarded as incorporating the schema with transaction-time support. The position is taken that valid-time schema support is not required. The schema is considered as a set of attributes making up relations and a class indicating the level of temporal support applicable to the relation (snapshot, historical, rollback or temporal). An algebraic language is defined using denotational semantics which subsumes the power of the relational algebra or any other arbitrary historical algebra. See also Ariav (1991) for a fuller discussion on this work.

Narayanaswamy, K. and Bapa Rao, K.V. (1988) : *An incremental mechanism for schema evolution in engineering domains,* 4th International Conference on Data Engineering, Los Angeles, CA, IEEE Computer Society Press, pp. 294-301.

Discusses *instance inheritance* as a mechanism for allowing the evolution of a class into a family of related instances, ie. versions. A *has-version* attribute is added to an attribute to link it with all of its versions.

Orlowska, M. E. and Ewald, C. A. (1992) : *Schema evolution - the design and integration of fact-based schemata.* in *Research and Practical Issues in Databases, Proceedings of the 3rd Australian Database Conference.* B. Srinivasan and J. Zeleznikow (ed.), LaTrobe University, World Scientific, pp. 306-320.

Orlowska, M. E. and Ewald, C. A. (1991) : *Meta-level updates : the evolution of fact-based schemata.* Tech.Rep. 211, Key Centre for Software Technology, Department of Computer Science, University of Queensland.

These papers view schema integration as a schema evolution process, ie. the integration of two or more schemas is effected by choosing one and applying the facts held in the others. The first paper investigates the semantics of fact addition, the second those of fact update and deletion.

Roddick, J. F. (1991) : *Dynamically changing schemas within database models,* Australian Computer Journal, Vol. 23, No. 3, pp. 105-109.

In this paper schema evolution is considered the meta-database analogue of temporal support in relational databases. It is investigated with particular reference to the semantics of null values, its effect on integrity constraints and its impact on query languages.

Takahashi, J. (1990) : *Hybrid relations for database schema evolution,* 14th Annual International Computer Software and Applications Conference, Chicago, IL, IEEE Computer Society Press, pp. 465-470.

Schema Evolution within the Object-Oriented Paradigm

Banerjee, J., Chou, H.-T., Garza, J. F., Kim, W., Woelk, D. and Ballou, N. (1987) : *Data model issues for object-oriented applications,* ACM Transactions on Office Information Systems, Vol. 5, No. 1, pp. 3-26.

Banerjee, J., Chou, H.-T., Kim, H. J. and Korth, H. F. (1986) : *Schema evolution in object-oriented persistent databases,* 6th advanced database symposium, Tokyo, pp. 23-31.

Banerjee, J., Chou, H.-T., Kim, H. J. and Korth, H. F. (1987) : *Semantics and implementation of schema evolution in object-oriented databases,* ACM SIGMOD conference, SIGMOD Record, Vol. 16, No. 3, pp. 311-322.

These papers and others, Chou and Kim (1988), Kim and Chou (1988), Kim, Garza, Ballou and Woelk (1990), Kim, Banerjee, Chou and Garza (1990), discuss schema evolution within the ORION prototype OODBS to varying extents; Banerjee, Chou, Kim and Korth (1987) being the most extensive. In these papers they define first a set of constraints (invariants) to maintain consistency under schema modification in the same way as integrity constraints are specified for data. Where this does not specify exactly what should happen in a given circumstance a set of rules are invoked to choose between choices. The allowable modifications are then presented in a *schema change taxonomy* and examples of the implementation are shown.

Beech, D. and Mahbod, B. (1988) : *Generalised version control in an Object-oriented database,* 4th IEEE International Conference on Data Engineering, Los Angeles, CA, IEEE Computer Society Press, pp. 14-22.

This paper discusses versioning within an Engineering domain and presents a flexible method of version control through the introduction of generic instances as a representative of all of the versions of that object. An application can reference either a specific instance of an object, to get a particular version, or its generic instance to obtain the latest version.

Bjornerstedt, A. and Hulten, C. (1989) : *Version control in an object-oriented architecture.* in *Object-Oriented Concepts, Databases and Applications.* W. Kim and F. Lochovsky (ed.), Addison-Wesley/ACM Press, pp. 451-485.

Discusses version control in general terms and within the context of the AVANCE object management system. The discussion is divided into versioning at the application level, to provide historical support, and at system level to provide transaction and concurrency control.

Chou, H. and Kim, W. (1988) : *Versions and change notification in an object-oriented database system,* 25th ACM/IEEE Design Automation Conference.

See notes under Banerjee, Chou, Kim and Korth (1987).

Christodoulakis, D., Soupos, P. and Goutas, S. (1989) : *Adaptive DB schema evolution via constrained relationships,* IEEE International Workshop on Tools for Artificial Intelligence. Architectures, Languages and Algorithms, Fairfax, VA, IEEE Computer Science Press, pp. 393-398.

Gibbs, S.J., Tsichritzis, D., Casais, E., Nierstrasz, O.M. and Pintado, X. (1990) : *Class management for software communities,* Communications of the ACM, Vol. 33, No. 9, pp. 90-103.

This is primarily a review paper which discusses issues in collaborative object-oriented software development. It presents (among other things) considerations relating to class evolution including some of the difficulties and design decisions concerning class versioning.

Kim, W. and Chou, H.-T. (1988) : *Versions of schema for object-oriented databases,* ACM SIGMOD Int. Conf. Very Large DataBases, Los Angeles, CA, pp. 148-59.

Kim, W., Garza, J.F., Ballou, N. and Woelk, D. (1990) : *Architecture of the ORION next-generation database system,* IEEE Transactions on Knowledge Engineering, Vol. 2, No. 1, pp. 109-124.

Kim, W., Banerjee, J., Chou, H.-T. and Garza, J.F. (1990) : *Object-oriented database support for CAD*, Computer Aided Design, Vol. 22, No. 8, pp. 469-479.

See notes under Banerjee, Chou, Kim and Korth (1987).

Lerner, B. S. and Habermann, A. N. (1990) : *Beyond schema evolution to database reorganisation*, SIGPLAN Notices, Vol. 25, No. 10, pp. 67-76.

The OTGen environment explained in this paper aims to take schema evolution a step further and presents an approach capable of reorganising the database. This is done by maintaining a class table which is used to map between versions of a database.

Nguyen, G. T. and Rieu, D. (1989) : *Schema change propagation in object-oriented databases*, Information Proceedings 89. Proc. IFIP 11th World Computer Conference, San Francisco, CA, North-Holland, pp. 815-820.

Nguyen, G.T. and Rieu, D. (1989) : *Schema evolution in object-oriented database systems*, Data and Knowledge Engineering, Vol. 4, No. 1, pp. 43-67.

A review of the support for schema evolution within the *Cadb*, *Encore*, *GemStone*, *Orion* and *Sherpa* systems is given and a *dynamic classification* proposal for propagating schema changes is discussed based around the idea of *relevant classes* (classes of semantically connected instances (see also other work by the authors on *Cadb*)).

Osborn, S. L. (1989) : *The role of polymorphism in schema evolution in an object-oriented database*, IEEE Transactions on Knowledge and Data Engineering, Vol. 1, No. 3, pp. 310-317.

This work accommodates schema evolution by developing an algebra which exploits polymorphism. In particular the algebra supports restructuring caused by the decomposition (aggregation) of an attribute and the addition of subclasses to an aggregate class in a generalisation hierarchy.

Penney, D.J. and Stein, J. (1987) : *Class modification in the GemStone object-oriented DBMS*, SIGPLAN Notices (Proc OOPSLA '87), Vol. 22, No. 12, pp. 111-117.

The GemStone approach to schema evolution explained in this paper adopts the approach of converting existing instances to the new class version rather than the late conversion (convert data when required) method used by Banerjee *et al.* (1986, 1987) and Skarra and Zdonik (1986). The advantage of simplicity for this method is compared with the flexibility of the other methods.

Skarra, A.H. and Zdonik, S.B. (1986) : *The management of changing types in an object-oriented database*, SIGPLAN Notices (Proc OOPSLA '86), Vol. 21, No. 11, pp. 483-495.

Skarra, A.H. and Zdonik, S.B. (1987) : *Type evolution in an object-oriented database.* in *Research direction in object-oriented programming*. B. Shriver (ed.), Cambridge, MA, MIT Press, pp. 393-416.

These papers look at the evolution of types as a mechanism for allowing multiple version viewing of data. Programs are able to use a view of the data that is applicable to the version of the program and data is able to be held in the form that was extant when it was last updated. This is achieved by employing filters between the old instances and the methods that expect new versions of the class.

Tan, L. and Katayama, T. (1989) : *Meta operations for type management in object-oriented databases - a lazy mechanism for schema evolution*, First International Conference on Deductive and Object-Oriented Databases, DOOD89, Kyoto, Japan, North-Holland, pp. 241-258.

Under the GemStone system (Penney and Stein (1987) all instances are converted to the new version immediately. Under the Orion system, Banerjee *et al.* (1986, 1987) and that of Skarra and Zdonik (1986) the data remains in the form extant at the time of last update. In this paper a mechanism of *lazy evaluation* (of a type similar to that of functional languages) is employed to alleviate the problems caused by a type

change by delaying the physical modification of the instance until it is accessed.

Miscellaneous Schema Evolution Research

Laine, H., Maanavilja, O. and Peltola, E. (1979) : *Grammatical database model*, Information Systems, Vol. 4, pp. 257-267.

This paper presents a database model based around a temporally sensitive *Assertion Expression Language*. The temporal aspect of the model is extended to the schema also and thus the model possesses a schema capable of modification over time.

Roddick, J. F. (1992) : *SQL/SE - A Query Language Extension for Databases Supporting Schema Evolution*, SIGMOD Record, Vol. 21, No. 3.

This work presents an extension to SQL to handle some of the functionality provided by schema evolution in relational databases. In particular the paper looks at:

i. *completed relations* and the problems of retrieving all data over all time;
ii. the problems presented by null values;
iii. dual time line support (ie. both historical and rollback support);
iv. the modifications that may be necessary to query language output.

Ventrone, V. and Heiler, S. (1991) : *Semantic heterogeneity as a result of domain evolution*, SIGMOD Record, Vol. 20, No. 4, pp. 16-20.

The paper discusses the problems specifically caused by changes in the domain of an attribute and its consequences for the interpretation of the data. Examples of situations where domain changes necessitate application program amendments are given and the characteristics of a possible solution are proposed.