Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1996 Proceedings

Americas Conference on Information Systems (AMCIS)

8-16-1996

A Methodology for Dynamic Schema Evolution in a Heterogeneous Database System

Sudha Ram
Department of MIS, University of Arizona, ram@bpa.arizona.edu

Shankarnarayan G.
Department of MIS, University of Arizona, gshankar@bpa.arizona.edu

A Methodology for Dynamic Schema Evolution in a Heterogeneous Database System

Sudha Ram, Shankarnarayan G., Dept. of MIS, Univ. of Arizona,
{ram,gshankar}@bpa.arizona.edu

Changes that occur in the real world over time affect the database that captures the real world. The extension of the database changes as the real world changes over time. The schema of the database should reflect these changes to be consistent with the real world. The implication of such changes is that the database schema needs to evolve over time as well. The ability of the database schema to change itself in response to changes in the real world is defined as schema evolution. Changes to database schema may also be triggered by changes in the database content. Schema evolution must ensure that the data or information that is currently stored in the database is preserved and application programs that access the database must not be made obsolete. In an earlier paper, we presented a graph theoretic methodology coupled with semantic modeling that helps manage and support the dynamic evolution of the database schema [18]. In this paper, we present a framework for a system that supports dynamic schema evolution in a heterogeneous database environment.

An important area of research in atmospheric sciences is the study of global climate change. The data studied here, referred to as Global Climate Change data, contain facts recorded by different measuring instruments that may be mounted on satellites, balloons and aircraft. These instruments record facts on cloud characteristics such as optical thickness, water content and cloud types as well as surface properties like vegetation index, surface radiation and surface flux. The data may be stored in a set of heterogeneous databases depending on the source (surface data recorded by earth observatories versus satellite data collected on cloud properties). The schema of such a set of heterogeneous databases changes often because of one or more of the following reasons:- (a) Changes to the real world. (b) Changes in user perceptions of the world and (c) Changes to existing data in the database. While this describes a scientific database, schema changes also occur in business [10] and engineering databases[4]. Given that the trend in database environment is towards heterogeneous databases, changes as described above emphasize the importance of providing support for dynamic schema evolution across heterogeneous databases.

Past Approaches to Schema Evolution

Research on schema evolution has mainly focused on using the object-oriented model. Kim et.al. have presented a taxonomy of schema changes in describing the implementation of ORION [2][3][9]. Other object-oriented approaches for schema evolution include constrained relationships [5], class evolution and versioning [6]. Osborn presents an algebra exploiting polymorphism in object-oriented databases [14]. Li and Mcleod describe an intelligent system based on object flavor evolution and kernal models [10]. Schema evolution using the relational data model has focused on temporal definitions and versioning. Temporal relational databases have been extended by extending relational algebra to provide temporal and evolutionary support [13],[1]. ER (and EER) models have also been used to describe schema evolution. Object oriented models tend to be restrictive in their applicability to heterogeneous database environments. Class relationships, attribute and class constraints remain hidden in the definition of objects and methods. The ER and EER models do not capture all the constraints of the real world such as complex objects and attribute/class constraints. The Unifying Semantic Model (USM) is an extension of the ER model and includes constructs that explicitly capture complex objects, attribute constraints and class constraints [15,16].The USM thus offers acceptable solutions to the above limitations and hence our schema evolution methodology using the USM can model many more of the evolutionary changes which the other approaches cannot, owing to the deficiencies in their underlying models. More importantly, the USM is a conceptual model that is independent of the underlying database(s) and hence is ideally suited for schema evolution in a heterogeneous environment. We feel that it is essential to have a schema evolution framework that is general enough to be applicable to all types of databases and which adequately represents all the required constructs.

Graph Based Representation of the USM

In this section we summarize a formal framework that maps any USM onto a graph [18]. A graph is defined as a representation that includes a set of nodes and links that connect these nodes - described by the set of associated nodes (N) and the set of links (L).The USM model lends itself to a graph-based representation due to its graph like structure. A graph theoretic representation of the USM is first formally defined on a graph G. In the USM, elementary classes can be treated as nodes in a graph and the relationships between them as links in the graph. Hence any USM can be represented as a graph G=(N, L) where N is a set of nodes on the graph that identify the different classes and L the set of edges that depict the relationship between the classes. An USM graph, G (N, L) where N={N_E , N_W , N_I , N_C , N_A , N_D , N_{Ct} ,N_{At} , N_M } is a disjoint union of a set of nine nodes and L={L_I , L_W , L_S , L_C , L_A , L_D , L_{Ct} , L_{At} , L_M } is a disjoint union of set of nine links or edges. As edges in G have directionality, the USM graph is a directed graph. The corresponding USM description of each member of N and L is presented in Table 1.

Node	Describes	Link	Directionality	Describes
N_E	Simple Strong Entity Class	L_I	Bi-directional	Interaction Relationship
N_W	Simple Weak Entity Class	L_W	Uni-directional to base class	Weak Class Relationship
N_I	Interaction Weak Entity Class	L_S	Uni-directional to base class	Subclass Relationship
N_C	Complex Class	L_C	Uni-directional to base class	Complex Class Relationship
N_A	Attributes	L_A	Bi-directional	Attributes property Relationship
N_D	Domains	L_D	Bi-directional	Domain Association
N_{Ct}	Class Constraint	L_{Ct}	Uni-directional to class	Class Constraint Relationship
N_{At}	Attribute Constraint	L_{At}	Uni-directional to attribute	Attribute constraint Relationship
N_M	Composite class Member	L_M	Uni-directional to comp. class	Member Relationship

Table 1: Description of nodes and links in the USM graph [18].

The graph based representation of any USM schema provides a mathematical foundation for representing changes that cause schema evolution. Changes to the USM can be modeled as changes to the USM graph. It also enables us to define the rules to maintain consistency of the schema before and after the change. As the edges have directionality, the USM graph is a directed graph. The in-degree of a node N in the graph G ($id_G(N)$) is the cardinality of the edges that point to the node and out-degree ($od_G(N)$) is the cardinality of edges that point out from the node N.

Schema Evolution Rules: Based on the above representation, we define the underlying set of rules for constructing and modifying the USM in terms of the nodes and links. The set of rules defined form the grid that defines how the changes to the schema will be captured and mapped onto the USM graph and therefore onto the semantic model (USM). Any changes that violate these rules will be rejected. For example, by definition of the USM, no weak class (N_W), subclass (N_E,N_W) or a complex class (N_C) can exist without a base strong/weak entity class (N_E). This can be stated as :

$od_G(N)$ 1, N {N_W, N_I, N_C}and a link L | L{ L_W, L_S, L_C } that contributes to od_G (N).

Schema Evolution Taxonomy: Having defined a graph G, based on the description of the USM model and the set of rules that govern the construction of G we can now describe the changes to the schema in terms of changes to the nodes and links of the graph. As an example, the set of possible changes to a simple strong entity class and the implications of each change is given below. For brevity we restrict ourselves to this example. For more details the readers are referred to [17][18].

Schema Evolution Operators. Based on the schema evolution taxonomy and rules described above, a set of schema change operators can be defined. A complete list of schema change operations, can be found in [17]. The schema change operators can be classified into three groups**:- operators that add new nodes or links** to the graph, **operators that delete existing nodes / links** from the graph and **operators that modify nodes /links** which can be described as adding and/or deleting links to/from the graph in any order.

Automating Dynamic Schema Evolution

Schema evolution process can at best be semi-automated. We have developed a prototype software system to provide semi-automated support to the schema evolution process. The front end of this system is a graphical for modeling the USM developed on a IBM RS/6000 platform running the AIX operating system. The repository for the meta data of the USM has been implemented using a RDBMS viz. SYBASE System 10. The schema evolution operators have been implemented using the C programming language. The tool is being tested using heterogeneous databases for studying global climate change.

NODE TYPE	CHANGES TO NODE	IMPLICATIONS OF CHANGES TO NODE
SIMPLE	Add Entity Class	Add Interaction Relationship, Add attributes
STRONG ENTITY	Delete Entity Class	Delete Int. Relationship, Attributes, Subclasses, Complex classes
CLASS	Change Name of Class	Add Attributes / Delete Attributes

Table 2: A Sample Taxonomy of Schema Evolution

Using the framework described above, we can identify the causes of schema changes and provide a mechanism that implements the changes to the database schema. The rules presented ensure correctness and consistency of the database schema during and after incorporating the changes. Moreover, changes need to be represented back in the underlying set of heterogeneous databases. In mapping the changes, new data would have to be added on to the database. The database administrator (and/or the users) involvement is essential in populating this data, which is why the evolution process cannot be completely automated. Consider a global schema that represents the current state of the heterogeneous databases. A Mapping Dictionary (MD) maps the federated /global schema to the underlying set of heterogeneous databases (HDB). Using the methodology above, we can convert this schema to an USM graph to which changes are applied. The resulting (modified) USM graph can then be converted back to another USM schema that has now evolved. A translator that lies between the USM and the database would translate the changes into data definition statements that can be accepted by the databases, using the MD and the evolved schema. It must also facilitate the capture of additional data based on the implemented definitions. The MD is then updated to reflect these changes so that the MD now maps the evolved schema to the set of HDB below. A schematic of this system is shown in figure 1 below. While there are several related research topics that need to be explored, our current focus is at providing a mechanism for mapping schema changes back to the set of heterogeneous databases. Such a mechanism would eventually obviate the need for major database re-organizations by providing a semi-automated support for schema evolution. Since USM provides a generic method of mapping without any consideration for the type of the underlying database, we feel that our research provides a good foundation for ongoing research on managing schema evolution.

.

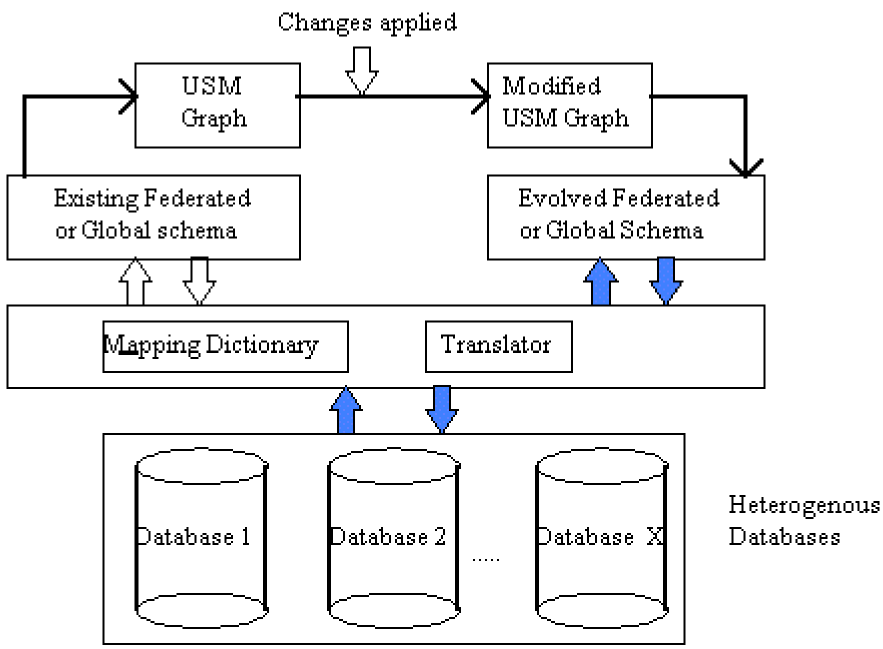

Figure 1: A System for Dynamic Schema Evolution

References

Available upon request from Dr. Sudha Ram, Dept. of MIS, College of Business and Public Admin.,

430, McClelland Hall, University of Arizona, Tucson, AZ 85721. (Phone: (520) 621-2748; Fax (520) 621-2433)