EVOLUTION OF INFORMATION SYSTEMS WITH DATA HIERARCHIES

Bogdan Denny Czejdo
Department of Mathematics and Computer Science, Fayetteville State University
Fayetteville, NC 28301, U.S.A.

Keywords: Information Systems (IS), System Evolution, Schema Evolution, Application Evolution.

Abstract: Recently, the research and practical efforts have intensified in the area of Information Systems (IS) supporting data and application evolution. The need to support IS evolution is caused by a variety of reasons including dynamicity of data sources, changing processing requirements, and using new technologies. In this paper we concentrate on evolution of IS data repositories caused by dynamicity of data sources. Our approach is to capture changes of various data hierarchies and use them as rules to implement evolution of IS data repository. Evolution of hierarchies can be categorized into hierarchy creation, hierarchy deletion, and hierarchy modification.

1 INTRODUCTION

The Information Systems (IS) currently support and underlie most, if not all, of our human activities. Throughout many years the importance of maintenance phase of development of IS was stressed by practitioners and by researchers. This effort, however, has not resulted yet in a complete and systematic solution for evolving IS. There is still a gap between relatively informal guidance for building maintainable IS and formal rules for IS evolution. It is very important to continue research and practical efforts in the area of Information Systems (IS) supporting data and application evolution. Such systems are sometimes referred to as Sustainable Information Systems (SIS).

The need to support IS evolution is caused by variety of reasons including dynamicity of data sources, changing processing requirements, and using new technologies (Rudensteiner, 2000) (Eder, 2001a) (Eder, 2001b) (Elder 2002). There are many aspects of IS evolution that need to be addressed. In this paper we concentrate on dynamicity of data sources causing evolution of IS data repositories. Our approach is to capture changes of various data hierarchies and use them as rules to implement evolution of IS data repository. Rather than describing atomic schema changes, our approach is based on changes of larger schema components referred to as hierarchies. Evolution of hierarchies can be categorized into: hierarchy creation, deletion, and modification for both schema hierarchy and instance hierarchy.

There are two approaches to IS evolution: logical or physical. In the logical evolution data repositories are integrated only at the logical level by referring to an integrated repository logical schema (no integration of old and new repository contents takes place, all data is stored only locally inside the repositories). User queries executed against the integrated repository logical schema are decomposed into queries for old repository and new repository. Queries issued for the old repository may need to be translated (Wiederhold, 1998). The advantage is that no central database is required to physically integrate old and new data. There are, however, serious disadvantages of such approach such as the need to maintain two or more data repositories and delays related with query transformations.

As opposed to the logical evolution, the physical evolution integrates both schemas and data. It requires extraction of data from old repository, checking consistency of old data against new data and updating the new repository. Queries submitted to the data repository are executed locally, without accessing the old repository, which considerably increases the query performance. It improves the availability of data. The SIS based on physical evolution can provide users with additional information such as aggregates, summaries or

Denny Czejdo B. (2008).
EVOLUTION OF INFORMATION SYSTEMS WITH DATA HIERARCHIES.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 351-356
DOI: 10.5220/0001716503510356

historical data. Physical evolution is closer to data warehousing approach and allows to use its popular technology for IS requiring high query performance and high data availability (Adamson, 1998) (Kimball, 1996) (Bischoff 1997) (Elmagarmid, 1999).. Therefore, physical evolution seems to be a better approach for IS unless evolution steps are small and happen very often.

Repository obtained by integrating various EDS often contains a variety of data hierarchies such as a product structure, a production organization. Modeling some of these hierarchies was discussed in the literature (Czejdo, 1996) (Kim, 1991). In this paper we describe a method for modeling evolution of various types of hierarchies.

This paper is organized as follows. In Section 2 we present the architecture of an Information System with EDS. In Section 3, we discuss the general framework for the evolution of such IS. Section 4 describes different types of hierarchies. Modeling of evolution of hierarchies is discussed in Section 5.

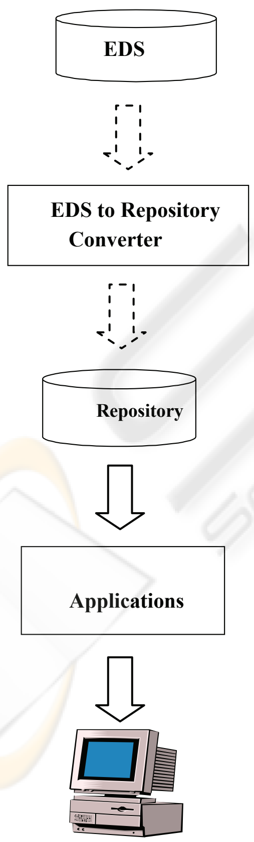

Figure 1: An Information System with External Data Sources.

2 ARCHITECTURE OF AN INFORMATION SYSTEM WITH EDS

Our discussion concentrates on evolution of an Information System with external data sources (EDS). Architecture of such system is shown in Figure 1. External data is processed by an EDS-to-Repository converter. This converter monitors changes to EDS, extracts data from EDS, cleans them and transform them into the common stored repository model. Such a converter is responsible for discovering inconsistencies in the source data, integration and transformation of data, data loading and refreshment, ensuring data quality, etc.

The stored repository is designed to integrate all EDS. The EDS may vary from proprietary applications and legacy systems to modern relational, object or object-relational database systems. They may include flat files, spreadsheets, XML documents, news wires or multimedia contents. All EDS usually differ in data model, require different user interfaces, and present different functionality.

3 ARCHITECTURE OF A SUSTAINABLE INFORMATION SYSTEM WITH EDS

Most IS assume, that data sources' schemas are static and that only the data changes. However, this assumption doesn't hold in the real world applications. Changes occur frequently in EDS. Most often those changes concern data instance or classification hierarchy (e.g., assigning an object to another subclass, merging two subclasses, etc.). After such change, queries involving data affected by the change begin to yield incorrect results.

Contemporary, most IS are unable to handle such changes, which hinders their functionality. In this paper we discuss the sustainable IS that guarantees the basic requirement to have access to all new data items and to maximize access to old data items. There are many approaches to build such a system. The physical evolution approach is described in this paper. The system architecture of such system is shown in Figure 2. We will concentrate in this paper on one of the most important tasks, namely, how to consolidate old data repository and the new data repository.

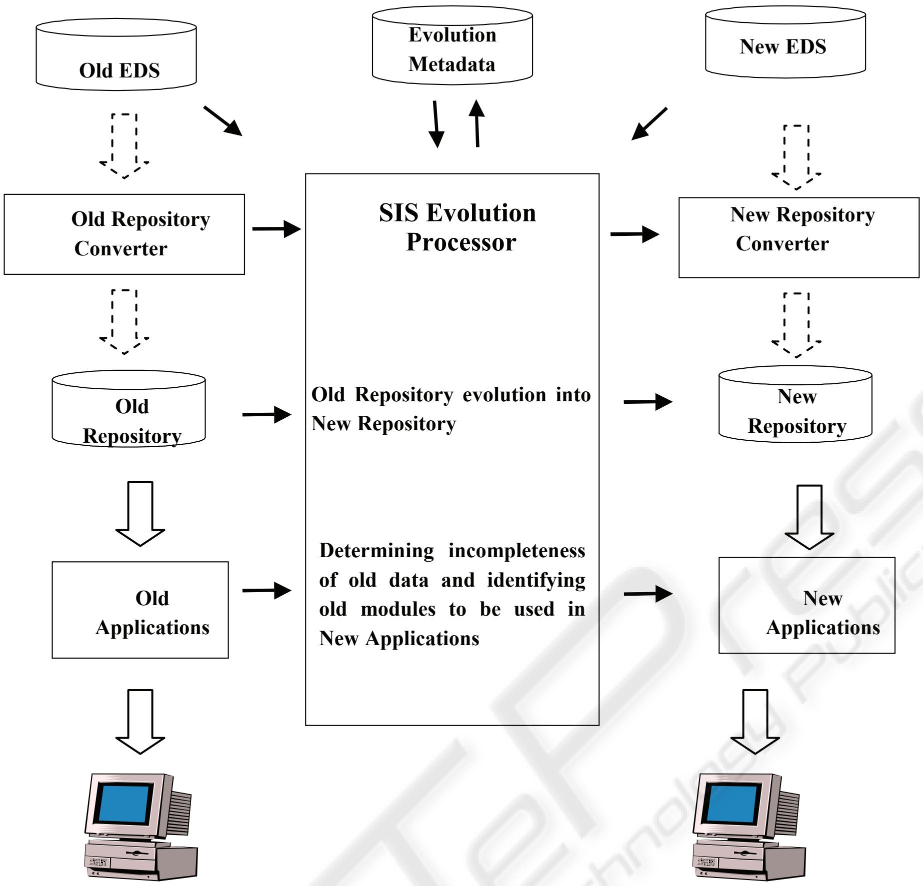

Figure 2: A Sustainable Information System with External Data Source.

4 HIERARCHY MODELING

The IS repository can contain many different data hierarchies. These hierarchies describe various relationships between data such as a production structure, organizational units (divisions, departments, branches etc.), a structure of products, classification of products, etc.

The data hierarchy can be component hierarchy or classification hierarchy. The example of component hierarchy graph is shown in Figure 3. This hierarchy is typically based on part-of/consists-of relationship between entity sets. Within component hierarchy we distinguish two subtypes of hierarchies. The first subtype of component hierarchy is used to describe structure in which we can identify and name all levels. We call this component hierarchy with well-established levels. In generally, this hierarchy_describes an organizational and/or production structure in which we can identify and name all levels.

Each level corresponds to homogeneous enterprise objects (objects with identical properties) whereas different levels can contain heterogeneous enterprise objects (objects with different properties).

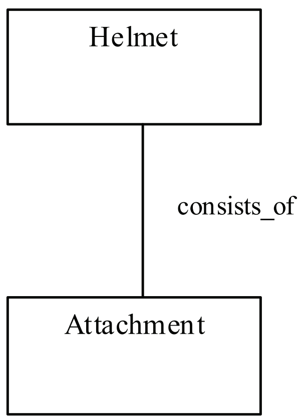

Figure 3. Example of Component Hierarchy in Casualty Information System.

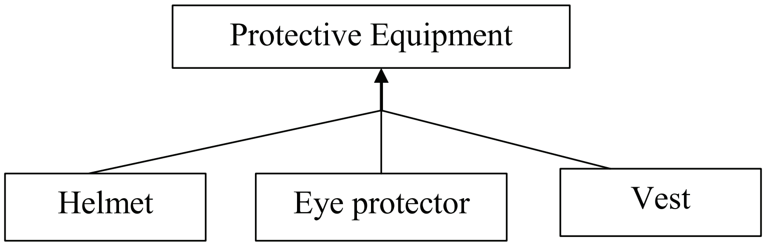

Figure 4: Example of Classification Hierarchy in Casualty Information System.

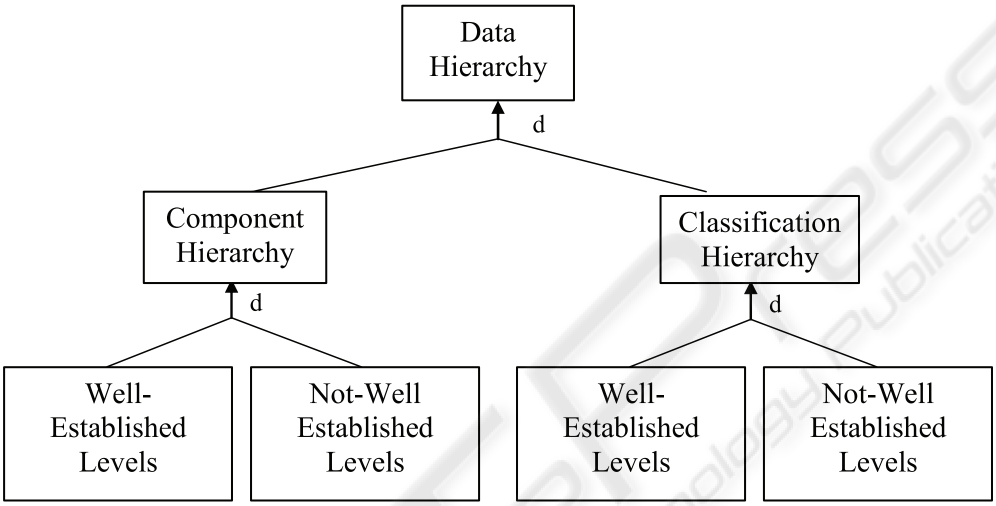

Figure 5: Meta-model for Data Hierarchies.

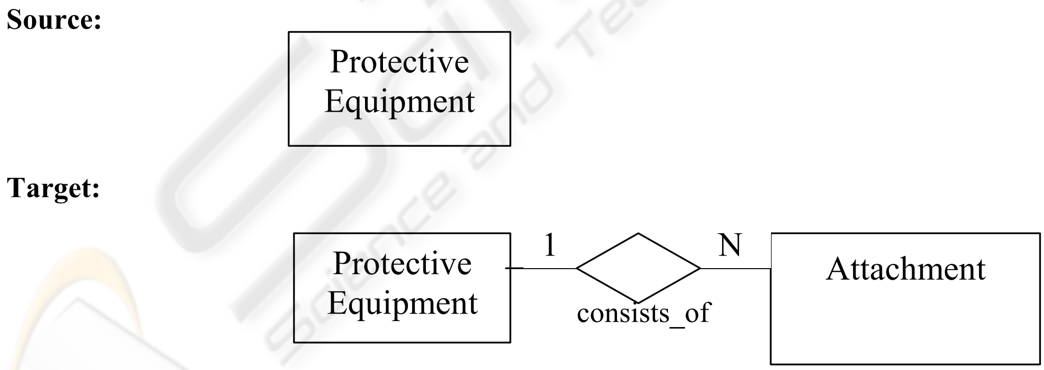

Figure 6a: A Schema Rule for Evolution of an Entity Type into Data Hierarchy with Well-Established Levels

Source: `Protective_Equipment = {A, B, C, D, E, F}`

Target:
```
Protective_Equipment = {A, B, C, D}
Attachement = {E, F}
Consists_Of = {(A,E), (B, E), (D, F)}
```

Figure 6b: An example of Instance Rule for Evolution of an Entity Type into Component Hierarchy With Well-Established Levels.

The second subtype of component hierarchy is used to describe the hierarchy where the level names are of no importance and the objects are homogenous (they have the same attributes). We call this component hierarchy with not-well-established levels. In general, component hierarchy with not-well-established levels is typically used to model parts that are built from other parts.

The data hierarchy can be also classification hierarchy. The example of classification hierarchy graph is shown in Figure 4. It is based on is_a or subclass relationships. This hierarchy is used to describe classifications of entities into types and subtypes and therefore it is referred to as classification hierarchy. When this hierarchy describes groups with different properties, then we call it classification hierarchy with well-established levels. When this hierarchy describes groups with identical properties or that can change often, then we call it classification hierarchy with not-well-established levels.

Hierarchy modeling can be described by a meta-model shown in Figure 5.

5 EVOLUTION OF HIERARCHIES

The general objective of schema evolution is to provide a new schema that will integrate old and new data and allow a user to view data using a uniform environment. The issue of schema evolution is difficult due to the semantic heterogeneity between old and new schema that appears in the form of schematic and data conflicts among component databases. Using our approach of hierarchy evolution within schema evolution can alleviate some problems.

The evolution of hierarchies can be described by transformation rules. Generally, there are schema and instance transformation rules. The example of schema transformation is shown in Figure 6a. It consists of source and target schemas. The example of instance transformation rule is shown in Figure 6b and consists of source and target set definitions. The transformation rules can be graphical or symbolic. For conciseness of the presentation we limit ourselves to symbolic rules based on individual instances in this paper.

Different categories of evolution of hierarchy were identified: hierarchy creation, deletion, and modification for both component hierarchy and classification hierarchy. Let us concentrate ourselves on evolution of an Entity Set into various data hierarchies. Let us start from discussing creation of a component hierarchy, when in the process of evolution, the data hierarchical structure is created from the simple set of instances as shown in Figure 6. Let us assume that the component hierarchy with well-established levels. In our case we create two levels: Protective Equipment and Attachment as illustrated in Figure 6a.

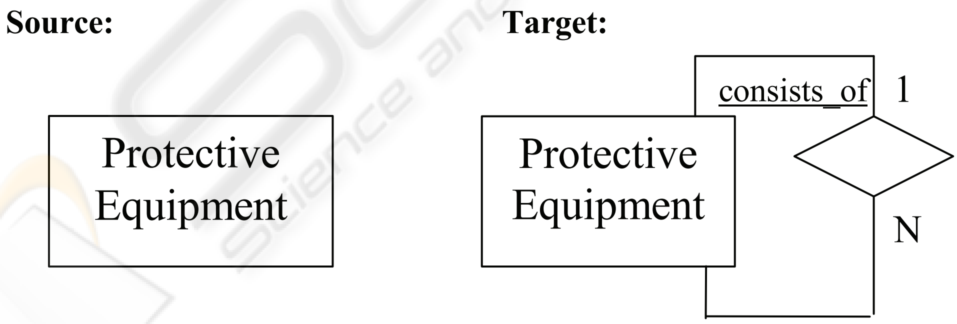

Figure. 7a: A Schema Rule for Evolution of an Entity Type into Component Hierarchy with Not-Well-Established Levels.

Source: `Protective_Equipment = {A, B, C, D, E}`

Target: `Protective_Equipment = {A, B, C, D, E}`
`Consists_Of = {(A,B), (A, C), (A, D), (B, E)}`

Figure. 7b: An Instance Rule for Evolution of an Entity Type into Component Hierarchy with Not-Well-Established Levels.

Let us now discuss evolution of an entity type into a component hierarchy with not-well-established levels. This evolution, takes place when in the process of changes the data hierarchical structure is created from the simple set of instances as shown in Figure 7. In our case of protective equipment it means that one instance of protective equipment can consist of other protective equipments, and they in turn can have their own components, etc. as illustrated in Figure 7.

6 CONCLUSIONS

In this paper, we discussed the problem of the data hierarchy evolution resulting from the changes in the underlying external data sources. We showed how an old data repository for IS can be transformed to new repository by applying appropriate transformation rules. We limited ourselves to individual instance-based rules. These rules can be easily expanded to include SQL-like set expressions and can be specified graphically.

REFERENCES

Adamson, C. and Venerable, M., 1998. Data Warehouse Design Solutions, John Wiley & Sons, Inc.

Bischoff, J., Alexander T.,1997. Data Warehouse: Practical Advice from the Experts. Prentice-Hall, Inc.

Czejdo, B., Morzy, T., Matysiak, M., 1996. Hierarchy and Version Modeling. In Proceedings of Symposium on Expert Systems and AI, ESDA '96, Montpellier.

Eder J., Koncilla C., 2001a. Changes of Dimensions Data in Temporal Data Warehouses. In Proceedings of the DaWak'2001.

Eder, J., Koncilla, C., Morzy, T., 2001b. A Model for a Temporal Data Warehouse. In Proceedings of the Intl. OESSEO'2001 Conference. Rome, Italy.

Eder, J., Koncilla, C., Morzy, T., 2002. The COMET Metamodel for Temporal Data Warehouse. In Proceedings of the CAISE'2002.

Elmagarmid, A., Rusinkiewicz, M., Sheth, A., eds, 1999. Management of Heterogeneous and Autonomous Database Systems. Morgan Kaufmann Publishers, Inc.

Kim, W., Seo, J., 1991. Classifying Schematic and Data Heterogenity in Multidatabase System. In IEEE Computer 24(12), 12-18.

Kimball, R., 1996. The Data Warehouse Toolkit, John Wiley & Sons, Inc.

Rudensteiner, E., Koeller, A., Zhang, X., 2000. Maintaining Data Warehouses over Changing Information Sources. In Communications of the ACM, vol. 43, No. 6.

Wiederhold, G., 1998. Mediators in the architecture of future information systems, In IEEE Computer C-25, 1.