



Chapter XII

Comprehension of Hierarchical ER Diagrams Compared to Flat ER Diagrams

Revital Danoch, Ben-Gurion University of Negev, Israel

Peretz Shoal, Ben-Gurion University of Negev, Israel

Mira Balabaan, Ben-Gurion University of Negev, Israel

ABSTRACT

HERD is a semi-algorithmic method for creating hierarchical ER diagrams from bottom up. The method is based on packaging operations that are applied in several steps on a given flat ER diagram. The result is a hierarchy of simple and interrelated diagrams (namely ER structures) with external relationships to other such diagrams. This chapter describes the application of HERD method using an example from a hospital domain, and an experiment in which we compare the comprehension of HERD diagrams with flat ER diagrams.

INTRODUCTION

An ER diagram (ERD) models the data structure of some reality in terms of entities, relationships, and attributes. In case of a large-scale application, the ER diagram may become very big, and it may be very difficult to understand and

to manage it, especially by users or managers (Teory, Wei, Bolton, & Koenig, 1989; Campbell, Halpin, & Proper, 1996). A mechanism is needed to improve ERD comprehensibility and to simplify their maintainability, in particular if we want to effectively apply the ER model on large databases. Indeed, the common ER model includes some abstraction mechanisms that support comprehensibility, mainly generalization (sub-typing of entity-types) and aggregation (whole-part relationships). However, these abstraction mechanisms alone do not solve the problem of too much detail in too small a space (Moody, 1996).

A possible approach to handle the problem is to create hierarchical ER diagrams. Layering of diagrams has been a key tool of abstraction, thus removing complexity of large schemas generated by enterprise modeling. For example, layered diagrams are common in “traditional” system analysis, where functional decomposition is carried out with data flow diagrams (DeMarco, 1978). Hierarchical structuring is also common in novel software engineering methods. For example, UML class diagrams include the *package* construct, but only as an organizational tool (like a folder in file management), and not as a first-class construct.

Various abstraction methods have been introduced in the ER literature. See, for example, methods proposed by Teory et al. (1989), Jaeschke, Oberweis, and Stucky (1993), Gandhi, Robertson, and Gucht (1994), Campbell, Halpin, and Proper (1996), and Moody (1996, 1999). A brief review of some of the abstraction methods is provided in Danoch, Shoval, and Balaban (2001). Consequently, they proposed HERD, a new semi-algorithmic method for creating hierarchical ERDs from bottom-up, namely from a given “flat” ERD.

The HERD method utilizes well-known packaging (grouping) operations that were proposed in some of the earlier studies. The packaging operations are applied in steps, thus creating a hierarchy of simple and interrelated diagrams. To enable this, the common ER model is enriched by introducing new constructs: *structure* and *composite relationship*. A *structure* is a (partial) ER diagram that consists of entities and relationships; it may also contain other structures (hence, it is similar to package in UML), and is related to other structures by external relationships. An external relationship is a relationship of an entity within a structure with other entities that belong to another structure. A *composite relationship* is a relationship between structures or between entities and structures. In other words, it is an aggregation of one or more specific relationships among entities that do not belong to the same structure.

The packaging operations are applied in four steps. In the first step, they are applied on the flat (bottom-level) ERD, creating leaf-level structures and external relationships. In the second step, the packaging operations are applied recursively on the bottom-level ERD as well as on the already created structures, thus creating higher-level structures (with external relationships to other entities) and composite relationships (within subordinate structures and entities). In the

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/comprehension-hierarchical-diagrams-compared-flat/23017

Related Content

Blockchain Technology: Principles, Applications, and Advantages of Blockchain Technology in the Digital Era

Satveer Kaur, Neeru Jaswal and Harvinder Singh (2022). *Applications, Challenges, and Opportunities of Blockchain Technology in Banking and Insurance* (pp. 204-212). www.irma-international.org/chapter/blockchain-technology/306463

Measuring the Determining Factors of Financial Development of Commercial Banks in Selected SAARC Countries

Arodh Lal Karn, Girish Santosh Bagale, Bhavana Raj Kondamudi., Deepesh Kumar Srivastava, Ravi Kumar Gupta and Sudhakar Sengan (2022). *Journal of Database Management* (pp. 1-21). www.irma-international.org/article/measuring-the-determining-factors-of-financial-development-of-commercial-banks-in-selected-saarc-countries/311092

Online Data Mining

Héctor Oscar Nigro and Sandra Elizabeth González Císaro (2005). *Encyclopedia of Database Technologies and Applications* (pp. 427-432). www.irma-international.org/chapter/online-data-mining/11184

Relational Techniques for Storing and Querying RDF Data: An Overview

Sherif Sakrand Ghazi Al-Naymat (2011). *Advanced Database Query Systems: Techniques, Applications and Technologies* (pp. 269-285). www.irma-international.org/chapter/relational-techniques-storing-querying-rdf/52305

BROOD: Business Rules-driven Object Oriented Design

Pericles Loucopoulos and Wan Mohd Nasir Wan Kadir (2008). *Journal of Database Management* (pp. 41-73). www.irma-international.org/article/brood-business-rules-driven-object/3381