Issues in Information Systems
Volume 15, Issue I, pp. 285-291, 2014

DATA MODEL TRANSFORMATIONS: RELATIONAL TO DIMENSIONAL

Vladan Jovanovic, Georgia Southern University, vladan@georgiasouthern.edu
James Harris, Georgia Southern University, jkharris@georgiasouthern.edu

ABSTRACT

The paper extends the Conceptual Dimensional Fact Model to support the modeling of data marts by formalizing patterns of graph transformations from a relational data base schema to a data mart schema, and by standardizing all visual representations using the Idef1X standard notation and a single supporting case tool. The proposed modeling and automation approach has the potential to significantly speed up prototyping, improve quality of design documentation, and allow verification by developers and validation by users.

Keywords: Dimensional Fact Model, Data Model, Idef1X notation, Relational DB, Graph Transformations, Dimensional Model, and Data Mart.

INTRODUCTION

Data marts (DM's) are access layers to data warehouses that allow subunits of an organization to efficiently access data in a Data Warehouse. Golfarelli [5] contends that a lack of adequate conceptual modeling and analysis in designing data warehouses and particularly DM's is detrimental to quality, speed and utility. The field of database design not only recognizes the need for separating conceptual design from logical and physical design, but also provides a variety of different models and techniques [2,3,6,7,12]. The review of approaches to conceptual DM design reveals only one industry tested and viable method and notation, the Dimensional Fact Model (DFM) [5]. The benefits of using a DFM to model/analyze a DM requirements is that it allows a visual medium to elaborate dependencies within dimensions, creates a representation conducive to systematic formulation/prototyping and testing of queries, facilitates validation of requirements, provides an explicit basis for logical and physical design and above all, provides a conceptual representation for communication with developers and possibly users. Among the drawbacks are anecdotal complaints that modeling analysis is manual and too slow, that all the interim DFM models will endure significant changes, that it is not supported with design tools used in industry for logical/physical design, and that it is not well grounded in theory.

This paper presents a formalization of operations on DFM's as graph transformations over an attribute graph with a potential for automatic verification/expansion of conformant dimensions and the automation of DFM transformations into Dimensional star schema models for data marts. Moreover, by treating all attributes as entities in DFM the same notation Idef1X can be used throughout as well as the same case tool. This facilitates and automates the construction of DM's from relational schema and provides additional tools in designing DM's.

RELATIONAL SCHEMA TO DATA MART

We propose a method of creating DM's from relational schema by applying graph operations to a DFM and then de-normalizing and collapsing the DFM into a DM. Because of the fact that very few data models are represented as trees, there are almost always graph structures in a data model pertinent to a selected fact that need resolution in the process of creating a DFM. The following structural patterns of attribute to attribute relationships have been identified;

a) Switching 1:1 attributes, with possible reduction

For example figure 1. "Code" is a dimension of "FACT" and "description" is an attribute of "Code". Since the relationship is 1:1, "Code" and "description" can be switched in the graph and "Code" can then be removed (assuming it is no longer needed in the schema).

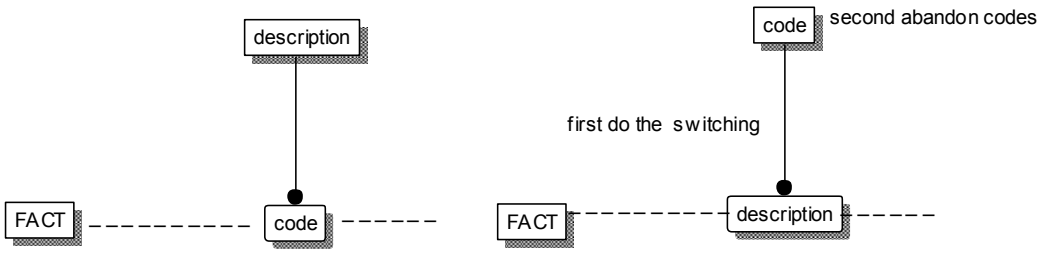

Figure 1. Switching and Reducing

b) Pruning, due to irrelevant attributes from an attribute tree (see figure 2).
c) Grafting: due to skipping irrelevant levels from an attribute tree (see figure 2).

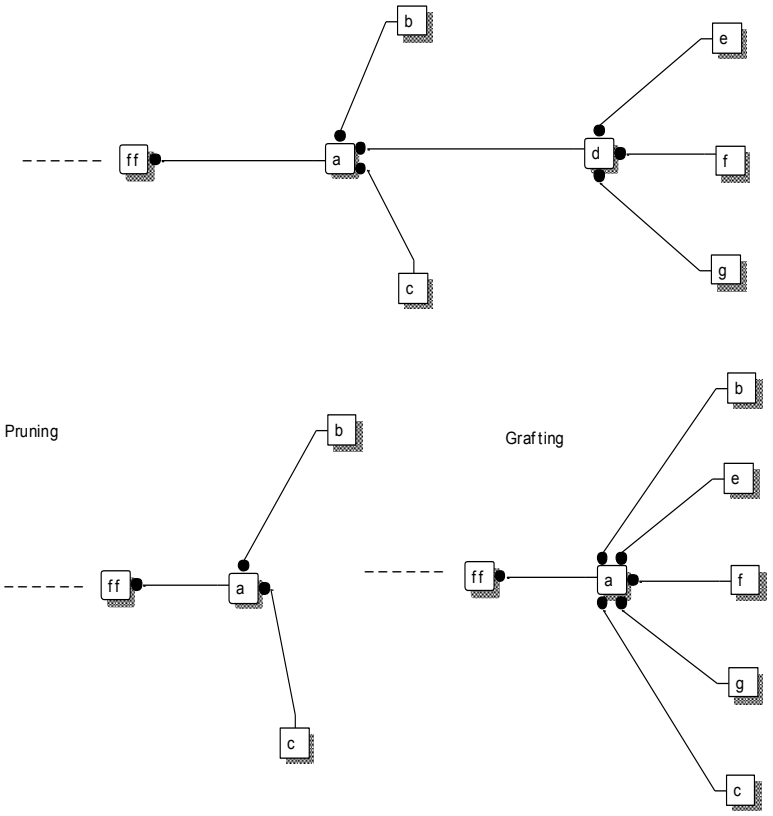

Figure 2. Grafting and Pruning

It is possible, in many cases (without addressing exceptions such as recursive/incomplete hierarchies or resolving bridge tables) using only the graph operations as illustrated above; to create DM's from a relational schema. The general steps are enumerated below:

a) An IDEF1X semantic data model is created from a relational schema.
b) From the IDEF1X model, a DFM is created by treating attributes as entities and selecting facts.
c) The graph operations (switching, pruning, and grafting) are selectively applied to the DFM.
d) All hierarchies are de-normalized and mini dimensions packaged into facts, creating a data mart
e) A prototype DM DDL for relational database engine implementation is generated (implementation is quite obvious for data models in Idef1X).

The following not fully normalized relational model (logical schema) describes an operational (OLTP) database (DB) source system for car rentals operation.

RENTAL_OFFICES (OfficeName, City, Area, State, Country)
CARS (LicensePlate, Category, Model, Brand, Fuel, RegistrationDate)
HAVE_OPTIONAL (LicensePlate(FK), Optional)
RENTALS (*LicensePlate(FK),* PickupDate, DropoffDate, *PickupPlace(FK), DropoffPlace(FK*), Miles)
DRIVERS (LicenseNumber, LicenseExpiration, DriverName, BirthDate)
DRIVE (LicenseNumber(FK), *LicensePlate+PickupDate(FK))*
INSURANCES (Risk, *LicensePlate+PickupDate(FK),* Cost)
PAYMENTS (*LicensePlate+PickupDate(FK),* Amount, Discount, PaymentMode)

Note that additional functional dependencies hold:

City → State →Country→ Area, and **Model → Brand**.

a) The Idef1X data model for the car rental system

To simplify the case (at least initially and as a consequence, an alternative model will be shown in discussing a final data mart design) we are dealing with one driver per rental (see Figure 3). Note that this model could have been obtained automatically from a DB schema by reverse engineering using a toll such as CA Erwin, Embarcadero, Power Designer or similar. Our preference for using IDef1X is its closeness to relational model.

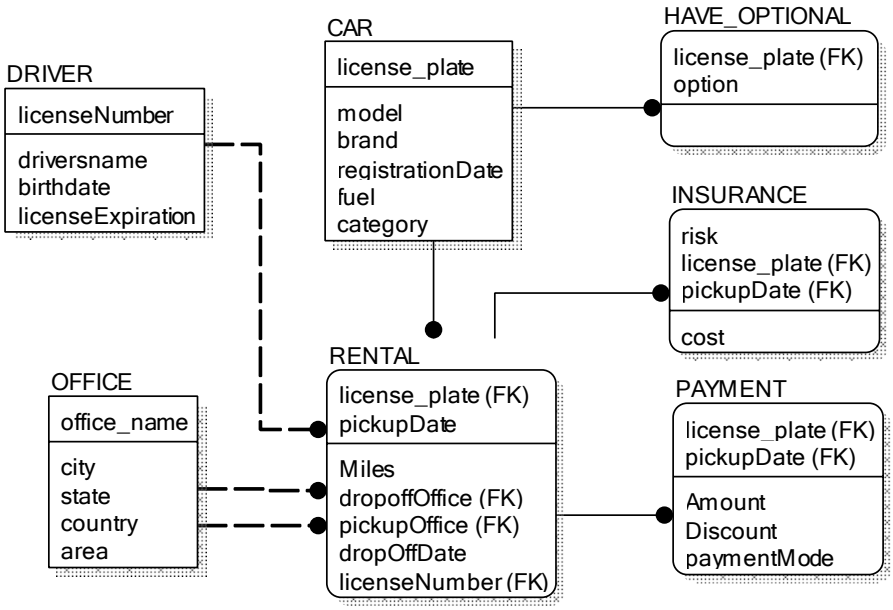

Figure 3. Initial Idef1X data model representation of Relational Schema

b) Creating the DFM schema for the car rental system:

There are two good candidates for facts: RENTAL and PAYMENT. In order to simplify the drawing, PAYMENT is not related to an attribute trees if RENTAL is. A standardized date dimension is added following a date dimension template (blueprint for a conformant date dimension).

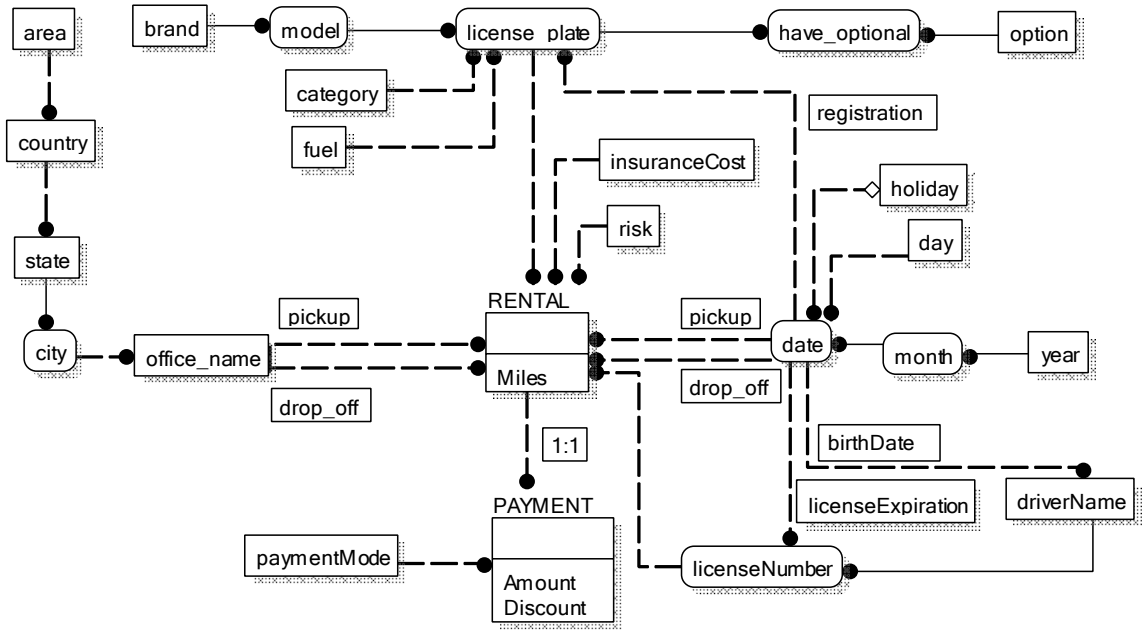

Figure 4. Initial DFM

Choosing either RENTAL or PAYMENT as a baseline fact, as the two are in a one to one relationship, is the prudent thing to do. Selecting RENTAL seems more natural and will be used in this example, although in a deeper analysis we could use two sequential events with overlap. In the edited attribute tree, the drop-off date is pruned and replaced by a Duration attribute computed as a number of days between the drop-off and the pick-up dates. All measures from PAYMENT are grafted, as well as payment mode, to the RENTAL. Insurance costs are added to the selected fact and the risk pruned. All the options on a car (license plate) are pruned to simplify analysis. The resulting interim DFM is shown below.

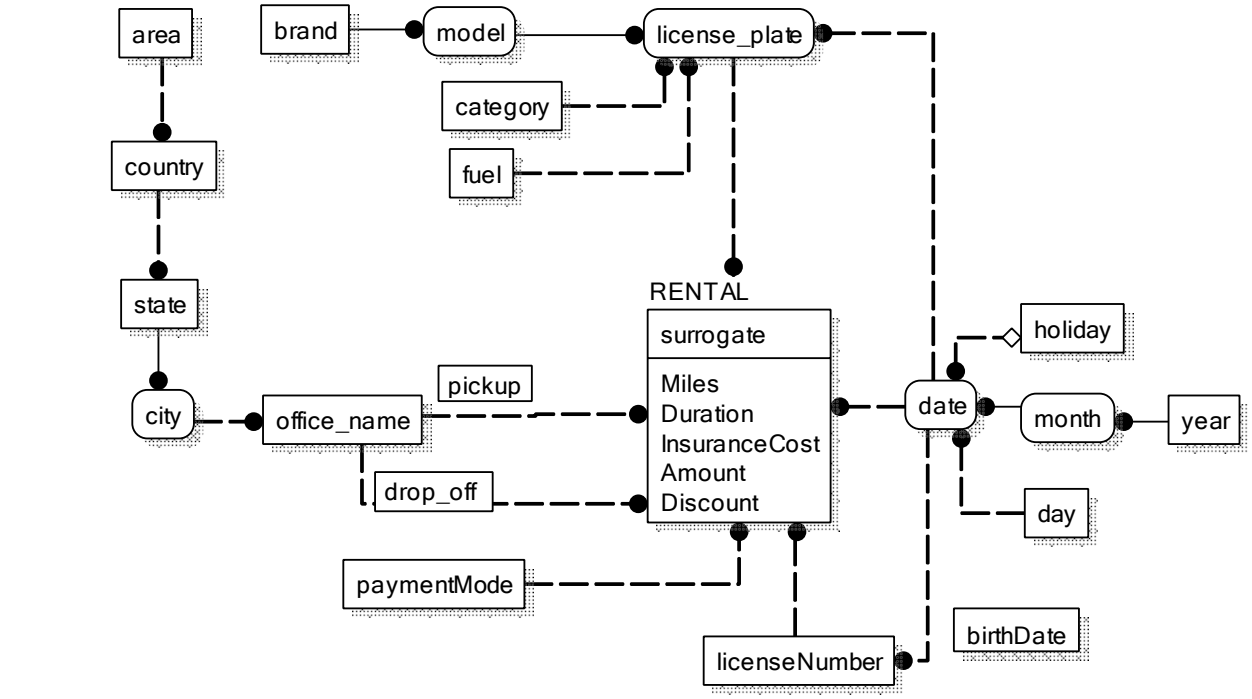

Figure 5. Modified DFM

Drivers name can be pruned, as well as license number and expiration date. Birthdate can be transformed to age. Names changed to office and car. A surrogate PK is selected for RENTAL to avoid identifying relationship from a car dimension (license_plate). The final DFM is shown as Figure 6.

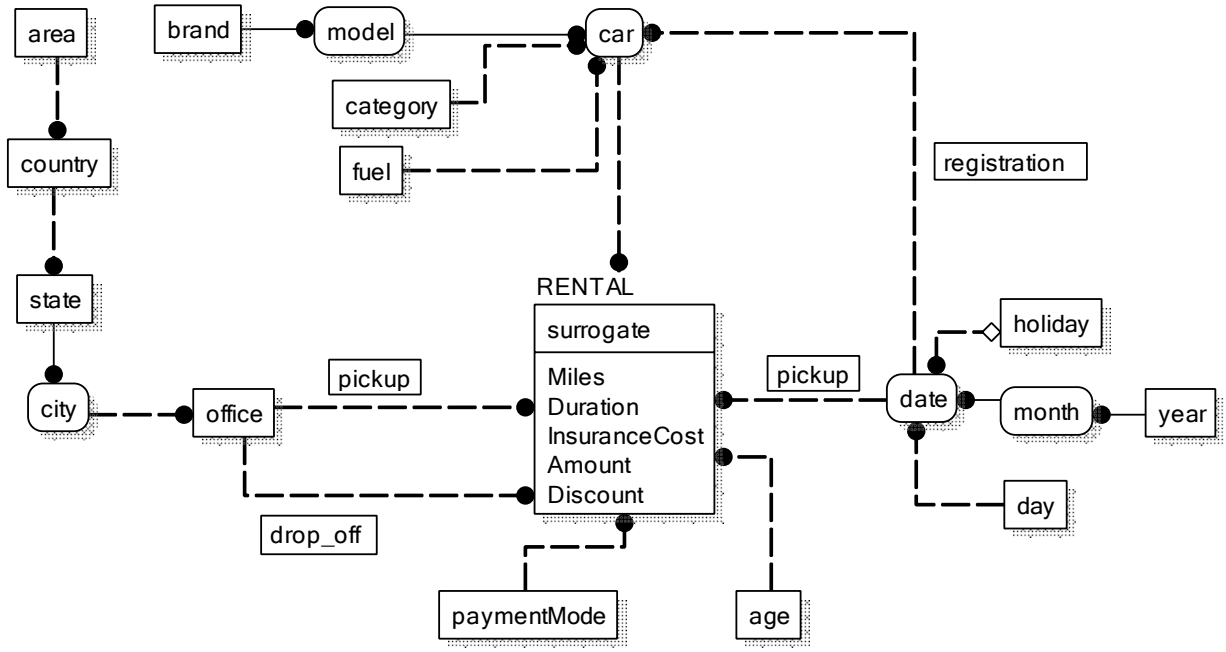

Figure 6. Final DFM

Collapsing (de-normalizing) all dimensional hierarchies and packaging mini dimensions into facts, a very simple data mart design emerges, see Figure 7. Comparing this result with an alternative, shown in Figure 8, the alternative model can handle a more complex source data set, however, there is a loss of efficiency, primarily due to more joins and more complex algorithms regarding the ages of drivers.

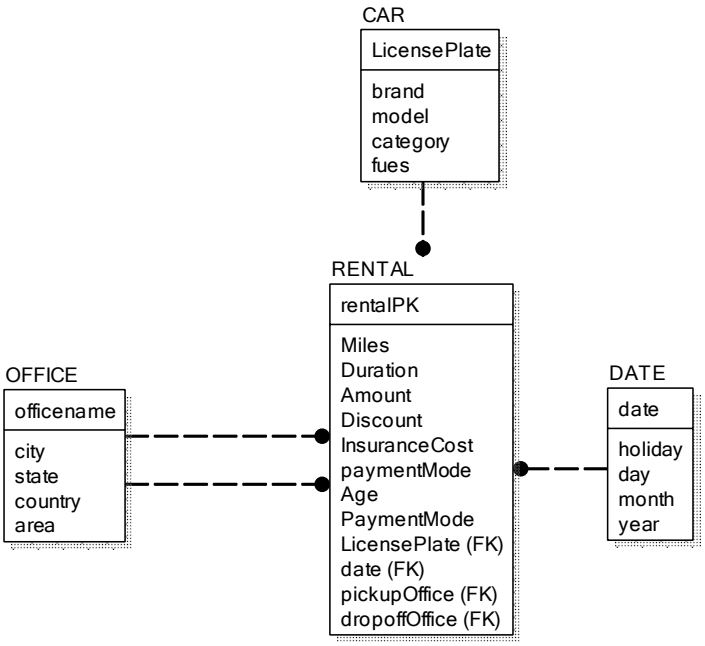

Figure 7. DFM de-normalized into a DM

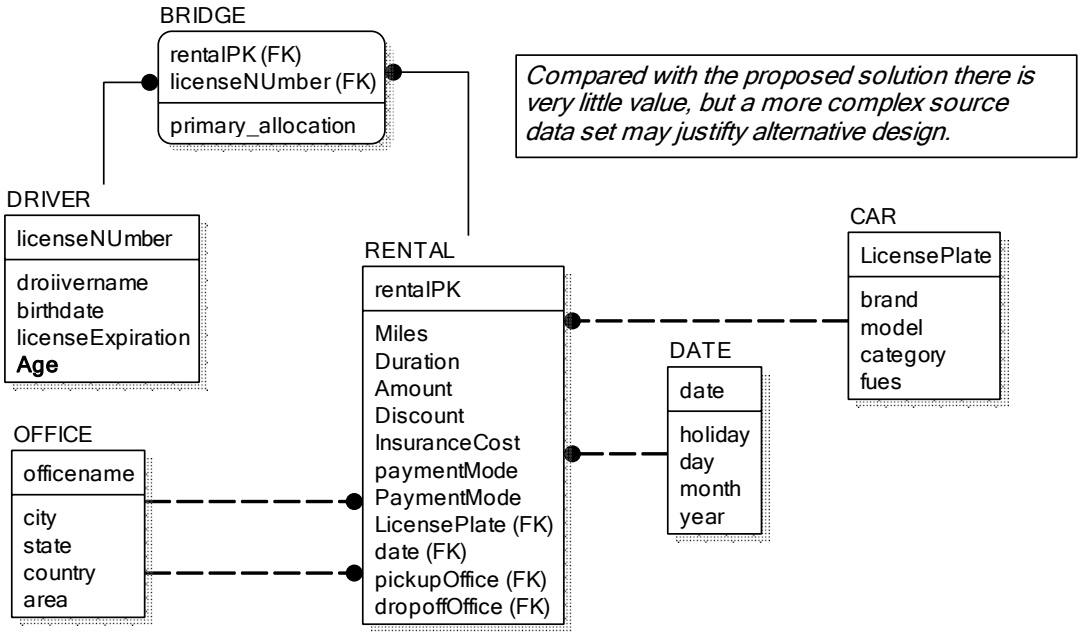

Figure 8. Data Mart alternative with multiple drivers

Either model can be directly prototyped with one click i.e. create table relational implementation schema created.

CONCLUSIONS

We have illustrated a method of using graph operations to generate a DM from a relational schema using DFM interim representation. The methods shown facilitate the design of DM's from relational schema and are a first step in automating the design. Future research will focus on exploiting matrix representations of directed graphs (incident/precedence matrices, see Langefors [11]) in manipulating interim DFM representations, and verifying conformance with already standardized dimensions. Furthermore, work on specifying and generating the ETL processes of transformations from RDB to DM is well underway, and includes research on metadata standardization. Future work includes working on the harder problem of schema evolution.

REFERENCES

1. Adamson, C. (2010). *Star schema- the complete reference*. McGraw Hill.
2. Batini, C., Cheri, S., and Navathe, S. (1992). *Conceptual database design.* Addison-Wesley.
3. Bruce, T. (1992). *Designing Quality Databases with Idef1X Information Models.* Prentice Hall.
4. Golfarelli, M., and Rizzi, S. (2009). *Data Warehouse Design*. McGraw Hill.
5. Golfarelli, M., and Rizzi, S. (1998). The Dimensional Fact Model: A Conceptual Model for Data Warehouses. *Int. journal of Cooperative Information Systems*, 7, 215-247.
6. Halpin, T. and Morgan, T. (2008). *Information Modeling and Relational Databases (2nd edition).* Morgan Kaufman.
7. Hay, D. (1996). *Data model patterns- conventions of thought*. Dorset House.
   Imhoff C., Galemmo N., and Gregor J. (2003). *Mastering data warehouse design- relational and multidimensional techniques*. John Wiley.
8. Jovanovic, V. (2008). Teaching Agile Data Model Validation. *Proceedings of the 9th ACM SIGITE conference on Information technology education, 139-146*.
9. Jovanovic, V., Subotic D., and Mrdalj S. (2014). Data Modeling Styles in Data Warehousing. *Proceeding of the 2014 international convention on information and communication technology, electronics and microelectronics (MIRPO)* [online]. http://mipro-proceedings.com/en/miprobis.
10. Kimball, R., and Ross, M. (2013). *The Data Warehouse Toolkit (3rd edition*). John Wiley.
11. Langefors, B. (1966). *Theoretical Analysis of Information Systems*. Studentliteratur, Lund.
12. Simsion, G. and Witt, G. (2005). *Data Modeling Essentials* (*3rd edition*). Morgan Kaufmann.