# Data PER Model

Hristo Tujarov,Stefan Kalchev, Vassil Milev

***Abstract:*** *In this paper the model PER (Process Entity Relationships) for semantic data modeling is presented, based on process and architectural approach. Terms, symbols and conventions used are explained. The levels of the model were described, and some example models are given.*

***Key words:*** *Data Base, Data Modelling, Data Base Model, Information System, Enterprise Architecture.*

## INTRODUCTION

In the development of the information systems, the database project is that fundament upon which the system is build. Data modeling is an integral part of database design. The main objective of data modeling includes:

- Study of the semantics of the data.
- Simplify the procedures of data requirements description.
- Synchronizing the business processes and provided data.

Data modeling requires some questions about different entities (objects), relations and attributes to be answered.

This paper aims at presenting a semantic data model in which the concepts Chen's ER [4] and Codd's [6] TM/p model are further developed. This model is based on process approach and is named PER (Process, Entity and Relationships) a computer.

## 1.SEMANTIC DATA MODELLING [2,5,8]

The widespread usage and popularity of relational databases demonstrate that relational data model successfully deals with modeling in different problem areas. The necessity of more convenient and powerful tolls for modeling bring along semantic data modeling.

The main advantage of semantic data models is their ability to present the semantics of the data. They assist in defining important type of objects (entities), their properties (attributes) and relationships between entities in problem area.

Codd defines the purpose of semantic modeling in that way: "In reality, the task of supporting the semantics of is endless." The aim of semantic modeling is very important, because even a very small success can contribute for understanding and put some sort of system in the field of database design".

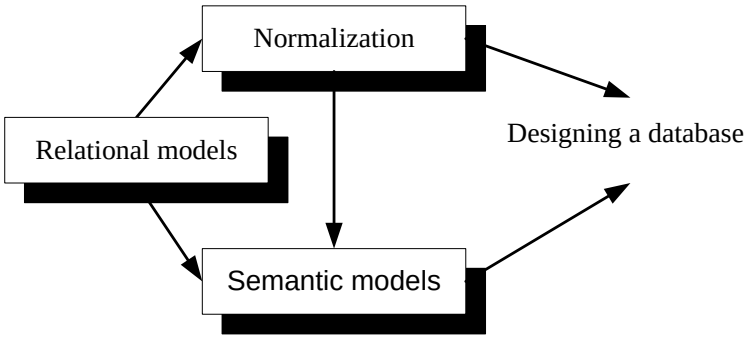



Figure.1 Data Modelling.

### The evolution of semantic data models[9,10,11]

Down the years semantic data models develop in two directions, outlined by Codd and Chen in the 70's. Every new model bring about:

- Better presented semantics of the data in the model.
- More easiness in study and suitability of the model for usage of CASE tools[3].
- More actual and detailed modeling of the object in computer world.

| | | |
|---|---|---|
| Data models | Entity Relationships diagrams (ERD) | CHEF, Martin, Bachman, IDEFXIX, Shlaer & Mellor, Merise, IEM |
| | Data model diagrams (DMD) | Martin, Bachman |
| | Data structure diagrams (DSD) | Jackson |
| | Logical Data structure diagrams (LDS) | SSADM |
| | Unified Modeling Language Diagrams (UML) | OOA&D |

Table 1. The evolution of semantic data models

## 2. THE PER MODEL

The **PER model** elaborate on semantic data models in several directions:

- Create new classification of the object.
- Add to model the process as a new object.
- Arrange particular type of objects in levels so to remove ambiguity in defining the type of relationships between objects.

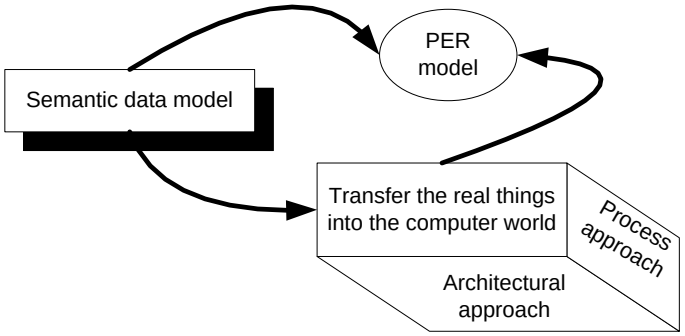



Figure 2. The PER Model.

In transferring the real objects into computer world the **PER model** is based on process and architectural approach [7].

**The process approach** – the activities of organization are chain of interconnected business processes – from marketing and planning to sell and services. It is natural to try to find relationships between the processes and data they provided in development of the information systems.The desired result can be achieved more effectively when the activities and respective recourses are managed as process [1,12].

**The architectural approach** – the idea of effective interaction between business and IT become a base of the architectural approach in which these two concepts are viewed as a whole – enterprise architecture. The data architecture stands as a base for synchronization of business with IT [12], which of its part is based on the data models [14].

### 2.1. TERMS, SYMBOLS AND CONVENTIONS USED

**PER object model –** a tool for modeling and structuring the data in the real world, integrate all objects in problem area with their relationships. The objects are arranged in hierarchical tree structure according to specific for the model rules.

**Object diagram** – a visual representation of PER model using defined symbols and conventions.

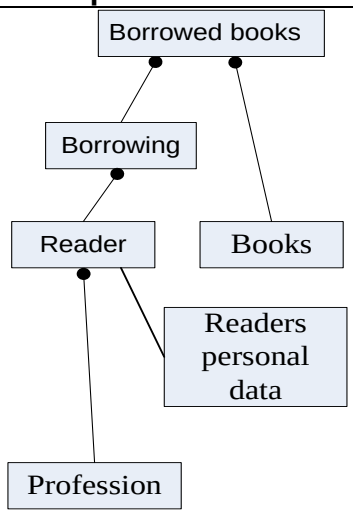
| Symbol | Meaning | Example | Description |
|---|---|---|---|
| AO | Associative object |  | Logical connection between two or more objects. Used in relations of type "many to many". Presented by expression – sale of products, students exam, borrowing of a book etc. |
| PO | Process object | | Include data of the process, subject to modeling. Presented by a verb – sell, buy, order |
| BO | Basic object | | Include basic data of the objects that take part in the process – direct or indirect. Presented by a noon – clients, products, books, authors… |
| SO | Subordinate object | | Include detailed data for the basic object. Presented by the name of the basic object and explanatory suffix – clientFD, clientCOM, clientID. |
| OL | Object fixed list | | Include relatively unchangeable data for an attribute of the basic object. Presented by a noon – city, profession, education, department… |
|  | attribute | | Property of the object of the real world. Every attribute has a value that identify a particular object – number, name, price,date.. |
|  1:N<br>  1: 1 | | | Type relationships between objects |

Table 2. Terms, symbols and conventions used.

## 2.2. LEVELS OF THE MODEL

The model is built of four interacted levels.

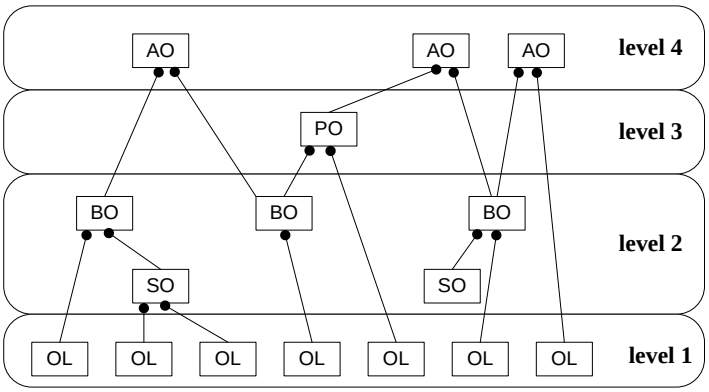



Figure. 3 Levels of the PER model.

**First level**

This is the level with relatively unchangeable data. It represents a catalogue of data that describe and explain the attributes from the upper level. In the application the objects in this level will appear as combo or list boxes from which the user can choose characteristic for particular instance of object. Once entered, this data can be use repeatedly in the upper levels. Examples: Cities, units of measure, color of eyes etc.

**Second level**

This is a level of objects that participate in the process. They are basic because of the roles they play in the process and are described in detail. Usually they are divided as basic, secondary and subordinate objects. This is done for making tables more simple and application more effective.

**Basic objects** – objects which attributes participate in transactions. Examples: client, books, products etc.

**Subordinate objects** – objects that are in relations of type 1:1 with basic object, and whose attributes are used for displaying variety of reports for the basic object. Example: Client's data for communication, Client's individual data, photos etc.

**Secondary objects** – kind of subordinate objects that exist relatively independently within the framework of the application and don't participate in the process object.They could be fully independent or bound with basic objects with N:M type of relation. Example: authors of the books, events etc.

**Third level**

This is the objects process level. All remaining objects and their roles in certain level depend on them. All basic objects that participate in the process are included. In some cases when the application are a type of box-index (for students, for books etc.) the objects process could be missing. For defining the attributes of this kind of object it is necessary some questions to be answered – who, when, where, what kind of document etc. Example: order, supply, borrowing, exam.

**Fourth level**

This level is for associative objects. They realized relationships of type N:M between two or more objects from lower levels. The associative object could be of two kinds – with or without participation of the object process. They are identified by composite primary key from the primary keys of the objects in relationship. If needed, some characterizing the relationship attribute could be included. More often they are quantity, price, distance etc. Example of associative object with process object and basic object: Order of products, Supply of inventory, borrowing of books, exam of students etc. Example of associative object without process object: authors of the books, communication with the client, materials for recipe etc.

### 2.3. RULES OF PER MODEL

- **The primary key of a type object** from lower level appears as foreign key of a type object from upper level.
- **The type of relationships** from the lower level to the upper level is 1:N.
- **The associative object** has a composite key from the primary keys of two or more objects from lower levels.

### 2.4. BASIC PRINCIPLES OF THE PER MODEL

**Uniqueness** – every attribute from the type object should represent unique information.

**Identification** – every object should have an identifier with one or more attributes.

**Functional dependency** – attributes included in the type object should fully describe them and depend on the identifier only.

**Data independence** – the changes of data for one object (except identifier) should not affect the other objects.

### 2.5. SEQUENCE IN CREATING PER MODEL

Generally the **PER model** could be used for data modeling in information systems that have one or more processes (as in the case of "Exam" bellow) and in index-card kind of systems (example "Personal").

In the first case the modeling begins with defining the process/processes and in the second with defining the main entity in the domain.

The realization of the **PER model** includes of 8 consecutive steps:

1. Defining process objects in problem area (domain).
2. Defining the basic objects.
3. Creating subordinate objects.
4. Defining the relations between basic objects and process object.
5. Creating associative objects with type of relationships M:N.
6. Defining attributes for created objects.
7. Creating reference objects for fixed list objects.
8. Creating data dictionary for all objects and their attributes. Key attribute are marked.

The attributes of the objects can be defined by answering certain number of questions, specific for every type of objects and some other properties (identifier, individual and common for some type properties).

In Table 3 an example shows specifying the attributes of the main objects used in **PER model.**

## 3. SPECIFYING THE ATTRIBUTES OF OBJECTS

| **Question** | **Example attribute** | |
|---|---|---|
| **PROCESS OBJECT – SELL OF PRODUCTS** | | |
| Who take part and how? | 1:n Seller<br>1:n Client<br>n:m Products | |
| When? | Date,time | |
| Where? | Shop, Store | |
| What kind of document? | Invoice, check | |
| Other characteristics | Transport included | |
| **ASSOCIATE OBJECT – PRODUCT SOLD** | | |
| Which objects are linked? | Process, Products | |
| What measure? | Units of measure | |
| How many? | Numbers indicating quantity | |
| **BASIC OBJECT – SELLER, CUSTOMER, PRODUCTS** | | |
| Individual properties | Name, address, notes | |
| Properties of type fixed list | Product type, settlement | |
| Object fixed list – product type, settlement | | |
| Individual properties | Name | |

Table3. Specifying the attributes of objects.

## 4. SAMPLE DIAGRAMS

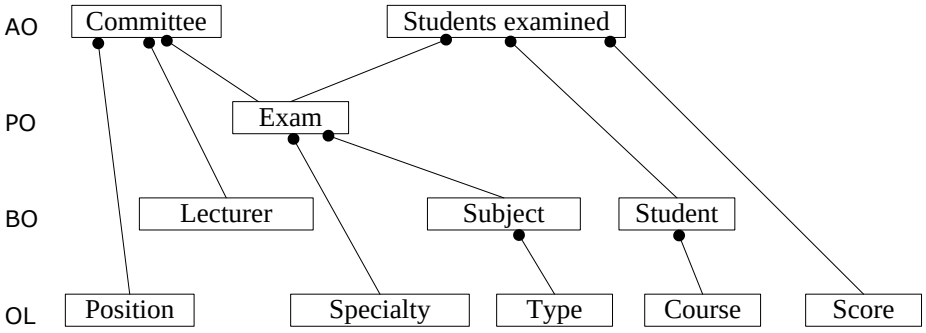



Figure. 6 Exam.

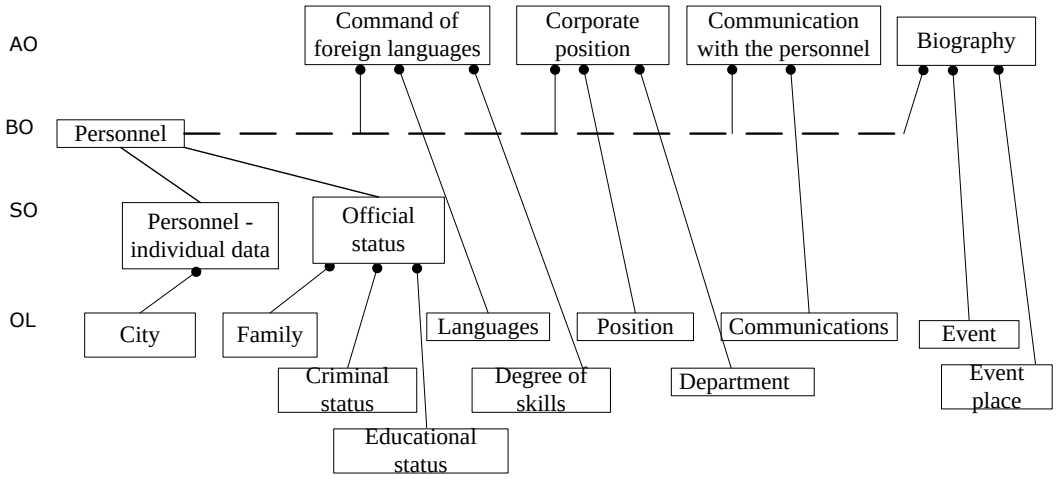



Figure. 7 Personell.

## CONCLUSIONS AND FUTURE WORK

The **PER model** is a suitable tool for data modeling by which the problem areas could be represented fast and effectively.

The model can be used in successfully in relational database design, because has the advantage of providing natural relationships with processes in the information system
In general the following advantages of **PER model** could be indicated:

- Including the processes as a main object allows better comprehension of the semantics in the problem area.
- The data could be easily identified with decomposition of the processes from different levels.

Hierarchical arrangement of the objects presumes unidirectional defining of the relationships between objects – from low to upper level always of type of 1:N.

## REFERENCES

[1] Bordens K., B. Barrington, Research design and methods: a process approach , 2010 | ISBN-10: 0073532029
[2] C. Chaomei., I.l Song, W. Zhu, Trends in Conceptual Modeling: Citation Analysis of the ER Conference Papers (1979-2005) http://cluster.cis.drexel.edu/~cchen/papers/confs/issi2007.pdf
[3] CASE Tools, http://www.umsl.edu/~sauterv/analysis/F08papers/View.html
[4] Chen, P.P., "The Entity-Relationship Model: Toward a Unified View of Data," ACM Trans. On Database Systems, Vol.1, No.1, March 1976, pp. 1-36.
[5] Date C. J., The Birth of the Relation Model, Intelligent Enterprise, Volume1,Number1, October 1998
[6] E.F. Codd. Extending the Database Relational Model to Capture More Meaning. ACM Transactions on Database Systems, Vol. 4, # 4, December 1979, http://citforum.ru/database/classics/codd_2/
[7] Greefhorst, D. , E. Proper ,Architecture Principles: The Cornerstones of Enterprise Architecture (The Enterprise Engineering Series) , 2011,| ISBN-10: 3642202780
[8] N. Rishe. Database Design: The Semantic Modeling Approach. McGraw-Hill, 1992, 528 pp, ISBN 0-07-052955-8, http://hpdrc.cs.fiu.edu/library/books/datades-book/
[9] Peter P. Chen ,.Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned http://www.csc.lsu.edu/~chen/pdf/Chen_Pioneers.pdf
[10] Roland K., L. Delcambre, The Evolution of Conceptual Modeling ,Springer,2011, ISBN 978-3-642-17505-3
[11] Virtual Exhibitions in Informatics, The history of conceptual modeling, http://cs-exhibitions.uni-klu.ac.at/index.php?id=185
[12] Милев В., С. Калчев, М. Стефанова. Architectural approach, based synchronization business and information technology, Научна конференция с международно участие „ Mathtech2010"- Шумен, 2010
[13] Тужаров Хр. Шест сигма, Процесен подход ,2008, http://tuj.asenevtsi.com/6S/6S01.htm
[14] Тужаров Хр.,В.. Милев, Ст. Калчев Архитектура на предприятието, Архитектурен подход ,2010, http://tuj.asenevtsi.com/EA/EA05.htm

## ABOUT THE AUTHOR

Prof. Hristo Tujarov, Department of Computer Systems and Technologies, University of Veliko Turnovo, Phone: ********** E-mail: hrituj@abv.bg

Assist. Prof. Stefan Kalchev, Department of Computer Systems and Technologies, University of Veliko Turnovo, Phone: *********, E-mail: stkalchev@mail.bg

Vassil Milev, PhD student, Department of Computer Systems and Technologies, University of Veliko Turnovo, Phone: +359 883 340 613, E-mail: milev.vassil@gmail.com