УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ" - ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ISSN 1857- 8691

ГОДИШЕН ЗБОРНИК
2012
YEARBOOK
2012

ГОДИНА 1 VOLUME I

GOCE DELCEV UNIVERSITY - STIP
FACULTY OF COMPUTER SCIENCE

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ГОДИШЕН ЗБОРНИК
2012
YEARBOOK
2012

ГОДИНА 1 **МАРТ, 2013** **VOLUME I**

GOCE DELCEV UNIVERSITY – STIP
FACULTY OF COMPUTER SCIENCE

ГОДИШЕН ЗБОРНИК
ФАКУЛТЕТ ЗА ИНФОРМАТИКА
YEARBOOK
FACULTY OF COMPUTER SCIENCE

За издавачот:
Проф д-р Владо Гичев

Издавачки совет	**Editorial board**
Проф. д-р Саша Митрев	Prof. Saša Mitrev, Ph.D
Проф. д-р Лилјана Колева - Гудева	Prof. Liljana Koleva - Gudeva, Ph.D.
Проф. д-р Владо Гичев	Prof. Vlado Gicev, Ph.D.
Проф. д-р Цвета Мартиновска	Prof. Cveta Martinovska, Ph.D.
Проф. д-р Татајана Атанасова - Пачемска	Prof. Tatjana Atanasova - Pacemska, Ph.D.
Доц. д-р Зоран Здравев	Ass. Prof. Zoran Zdravev, Ph.D.
Доц. д-р Александра Милева	Ass. Prof. Aleksandra Mileva, Ph.D.
Доц. д-р Сашо Коцески	Ass. Prof. Saso Koceski, Ph.D.
Доц. д-р Наташа Коцеска	Ass. Prof. Natasa Koceska, Ph.D.
Доц. д-р Зоран Утковски	Ass. Prof. Zoran Utkovski, Ph.D.
Доц. д-р Игор Стојановиќ	Ass. Prof. Igor Stojanovik, Ph.D.
Доц. д-р Благој Делипетров	Ass. Prof. Blagoj Delipetrov, Ph.D.
Редакциски одбор	**Editorial staff**
Проф. д-р Цвета Мартиновска	Prof. Cveta Martinovska, Ph.D.
Проф. д-р Татајана Атанасова - Пачемска	Prof. Tatjana Atanasova - Pacemska, Ph.D.
Доц. д-р Наташа Коцеска	Ass. Prof. Natasa Koceska, Ph.D.
Доц. д-р Зоран Утковски	Ass. Prof. Zoran Utkovski, Ph.D.
Доц. д-р Игор Стојановиќ	Ass. Prof. Igor Stojanovik, Ph.D.
Доц. д-р Александра Милева	Ass. Prof. Aleksandra Mileva, Ph.D.
Доц. д-р Зоран Здравев	Ass. Prof. Zoran Zdravev, Ph.D.
Главен и одговорен уредник	**Managing/ Editor in chief**
Доц. д-р Зоран Здравев	Ass. Prof. Zoran Zdravev, Ph.D.
Јазично уредување	**Language editor**
Даница Гавриловаска - Атанасовска	Danica Gavrilovska-Atanasovska
(македонски јазик)	(macedonian language)
Павлинка Павлова-Митева	Pavlinka Pavlova-Miteva
(англиски јазик)	(english language)
Техничко уредување	**Technical editor**
Славе Димитров	Slave Dimitrov
Благој Михов	Blagoj Mihov

Редакција и администрација	**Address of the editorial office**
Универзитет „Гоце Делчев“-Штип	Goce Delcev University – Stip
Факултет за информатика	Faculty of Computer Science
ул. „Крсте Мисирков“ 10-А	Krste Misirkov 10-A
п. фах 201, 2000 Штип	PO box 201, 2000 Štip,
Р. Македонија	R. of Macedonia

СОДРЖИНА
CONTENT

TOWARDS A GENERIC METADATA MODELING

Pavel Saratchev[1]

[1]*Technical University of Sofia, PhD Student, pavel.saratchev@triwdata.com*

Abstract: This document describes the creation of a generic metadata model prototype, combining the principles of Generic Modeling and Data Vault architectures.

Keywords: Generic, Metadata, Modeling.

27 Introduction

Metadata is essential component for the correct operation and further development of a BI/DWH-system. Despite this fact, most organizations do not have centralized processing and management of metadata. The main reasons for this fact are defined as:

- no established standards[13] - every software vendor has its approach to the preservation and management of metadata.
- additional resources and investments - resulting from a complex concept and implementation for centralized management of metadata.
- lack of clear understanding about the benefits which could bring a centralized management of the metadata.

For the reasons above, the goal of this paper is to design and implement a prototype for a generic metadata model which can be used for the creation of central repository for management of metadata with the following characteristics:

- universal model which can accept any type of metadata and so give a single point of view over all objects, business rules, processes and their dependencies throughout the systems in the organization.
- optimized and flexible structure with relative small amount of tables, which will facilitate the maintenance and reduce the costs of ownership.
- historization of the metadata with the ability to track changes and recover old versions of the data.

28 Conception

Meta data is often defined as “data about data”. This definition is rather abstract and can be interpreted differently in different context. But at this ambiguous aspect of the definition are expressed its accuracy and content. This is data that emerge from the modeling of certain objects and links between them in a given abstraction level and context. The model in every subsequent abstraction level is the meta model of the previous one. On fig. 1 is presented the dependence of models and meta data at various abstraction levels.

The real objects exist in the real world of abstract level 0. The model of the real object is abstract (simplified) image for a particular purpose[14] in a given context. With the modeling of real objects in the first abstraction level a

[13] Metadata standards: DDI, ISO 19115, ISO 19506, ISO 23081, MARC, CWM and many others.
[14] The purpose of modeling can be simulation, explanation, etc.

model 1 was created, which consists of metadata about the real object. A consequent modeling of the model 1 at the second level is a model 2, which in turn consists of a metadata from model 1. In the context of a real object, model 2 is its meta model and model 3 its meta meta model, which in turn is a model of a model 2 and meta model of model 1. Each model in fig. 1 has

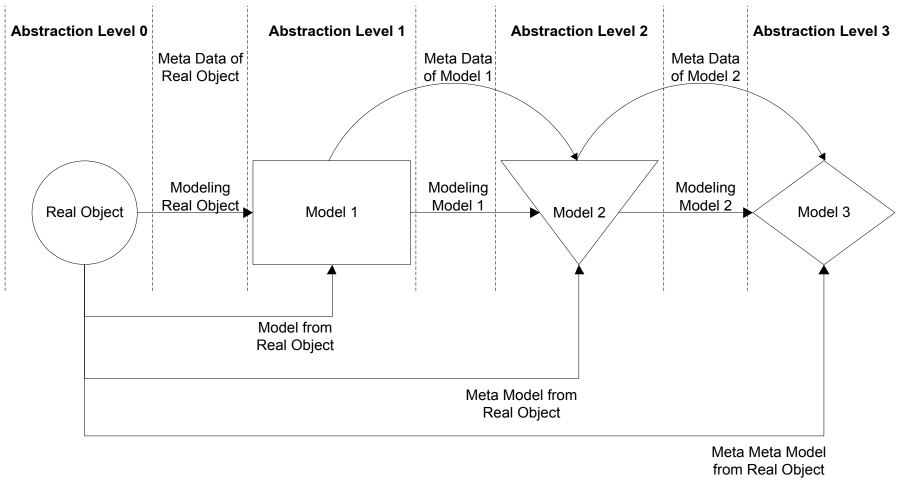

Figure 1 Models and Metadata

its own convention and language model[15]. By building a meta model on a higher abstraction level we can combine or consolidate different models having different conventions, semantics and meta data.

Meta data that arise from the modeling of objects and related processes are defined as *structural* metadata. Depending on the area of its origin, the data is classified as technical and business metadata. For example, technical metadata can serve as a description of a table that consists of columns of certain specific format. Example of business metadata could be the definition of a customer - which is a natural or legal person with certain attributes that has purchased a given product or service. The structural metadata gives the user the ability to navigate through its structure and in this way to get an understanding of the real objects without ever having touched them.

Structural metadata is a relatively static data, dependent on how often the underlying object or its model changes. Related objects however undergo changes mainly in the long run. In contrast, metadata derived from the interaction[16] of the objects is dynamic in nature and is known as *operational* metadata. Operational metadata could be statistics that describe events and

[15] The various conventions and language modeling are represented by the different forms of fig. 1.
[16] In this case the interaction between the objects is modeled.

processes related to the real objects. Depending on the area of its origin, metadata is also classified as technical or business. On fig. 2 is presented a classification of technical/business and structural/operational metadata.

In order to manage metadata with different origin and classification type the corresponding model has to be created in a high abstraction level with generic architecture. This will allow the creation of one single generic model for every type of metadata.

Business Metadata	• Structure and Definitions of business Objects and Measures (Customer, ROI, etc.) • Business Rules (what to do if Customer XYZ does or doesn't do smth.) • Etc.	• Statistics of the Business Objects and Measures • Patterns and Frequencies of Business Usage • Data Quality Statistics • Etc.
Technical Metadata	• Data Models • Source- and Target-Systems • Domain Values • Dependencies • Etc.	• Runtime Data • Storage Data • Patterns and Frequencies of Data Access • Etc.

Figure 2 Classification of Metadata[17]

The purpose of the Generic Modeling [2] is that some parts of the model or the whole model could be reused in other models without any or with minor local changes. In addition to this requirement changes of the data model should be minimized or avoided if possible, even with changing business rules. To achieve this goal the business model is divided to business rules and independent from them generic data model. In order for the model to be generic, the architect should follow the principles of generic design: [3]

1. *Candidate attributes should be treated as representing relationships to other entity types.*
2. *Entities should have a local identifier within a database or exchange file. These should be artificial and managed to be unique. Relationships should not be used as part of the local identifier.*
3. *Activities, associations and event-effects should be represented by entity types (not relationships or attributes).*
4. *Relationships (in the entity/relationship sense) should only be used to express the involvement of entity types with activities or associations.*

[17] For other examples of metadata ref. [1].

5. *Entity types should represent, and be named after, the underlying nature of an object, not the role it plays in a particular context.*
6. *Entity types should be part of a subtype/super type hierarchy of generic entity types in order to define a universal context for the model.*

The model produced with the Generic Modeling architecture is capable to adapt changes of business rules without any or with minor local changes. Its standardization [4, 5] allows it to be combined with other generic models and to be exchanged [6] between different organizations. Data is presented very consistently without any denormalization, which leads to the elimination of any anomalies.

Despite the advantages of this approach there are also some drawbacks. Mostly, they are reflected in the complexity of the data model. Large number of entity classes[18] and their relations leads to extremely large and complex data models that are practically incomprehensible in scope and can be managed only with a specialized software. The extensive normalization of the generic models leads to significantly lower performance of the queries.

To avoid these disadvantages we'll apply the Generic Modeling principles to the Data Vault (DV) [7, 8] architecture. The DV-architecture is optimized for storage and handling of the so called *raw data* and consists of three main types of entities: *Hubs*, *Satellites* and *Links*.

Hubs consist of lists of unique business keys. Example of hub can be the table of the entity **Product** which will contain all **ProductNr** (product numbers) of products that are known in the system as well as several technical attributes which are recorded by the ETL-processes and can be used for control such as: **SQN**, **LOAD_DTS**, **EXTRACT_DTS** and **REC_SRC** - sequence, load date timestamp, extract date timestamp and record source.

Satellites consist of descriptive data for business keys and/or their associations. They are built in the form of slowly changing dimensions (SCD 2) [9] and contain all historized information about the hubs or connections including the technical attributes listed above and **LOAD_END_DTS** - load end date timestamp, which contains the date on which the data set got a new version in its history.

Links in their turn consist of unique lists of associations which are representing the relations of two or more business keys. Basically they define the interactions between business objects which are represented by the hubs.

[18] For ISO 15926 there are more than 10.000 predefined entity classes.

On fig. 3 are listed the basic elements of the DV-architecture, which in this case consist of three hubs **Product**, **Customer** and **Order** with their satellites related with a link which records the history of the their relations.

With the separation of the attributes in hubs, satellites an links, DV present a highly flexible architecture with the following characteristics:

- changes in the data model is mostly carried out by adding additional links and satellites.
- different models can be just put together with creating of additional links and satellites.
- historization is done only within the highly denormalized satellites, which allows effective management of their life cycle.

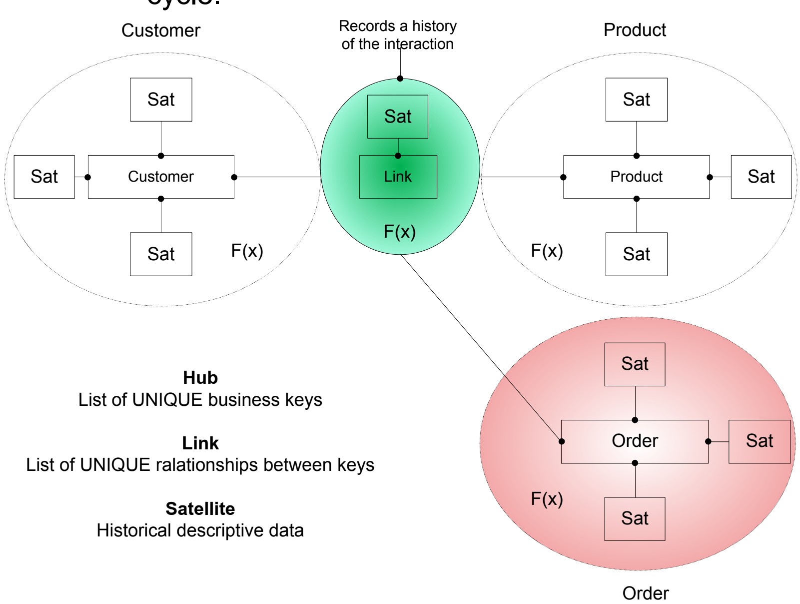

Figure 3 Example Data Vault ([8] p. 20)

- classification of the tables makes easier the navigation and orientation through the model and facilitates the standardization of the ETL-processes.
- model is optimized for storing the data but not for query the data. However, the advantages of this architecture outweighed this disadvantage.

By combining the principles of Generic Modeling with the Data Vault structure we can create a model which is generic enough to contain all kind of metadata and still be manageable.

29 Implementation

To achieve our goal we'll start the modeling in a high abstraction level. Every single object, which can exist alone will be presented as an object within the hub **H_OBJECT** and will have certain attributes respectively in the hub **H_ATTRIBUTE**. Objects and attributes will have certain types **H_OBJECT_TYPE** and **H_ATTRIBUTE_TYPE**. On fig. 4 is presented the first draft of the model.

Every object can have any number of attributes, and each attribute can be owned by any number of objects. Relationship between the entities objects and attributes will have cardinality **(n:m)** and be implemented through an additional entity **L_OBJECT_ATTRIBUTE**. In the same way will be expressed the relationships between all other entities. Thus will observe the third and fourth principle of Generic Modeling that require activities and associations to be modeled with separate entities which are the links. Relations between the attribute type and attribute **L_ATTRIBUTE_ATTRIBUTE_TYPE** and object with object type **L_OBJECT_OBJECT_TYPE** are modeled on the same principle. Object types can also have attributes, which is modeled by their relation on the

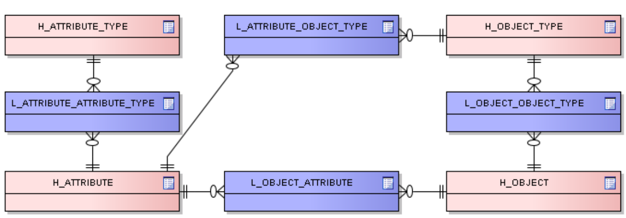

Figure 4 First Draft of the Generic Metadata Model

same principle. Object types can also have attributes, with the link **L_ATTRIBUTE_OBJECT_TYPE**.

All of the entities are named after their main character, which is the requirement of the fifth generic principle. The prefixes **H**-HUB and **L**-LINK are implying their use which is a hub and a link. Every entity model will have an artificial primary key (technical key), which is the second principle of the Generic Modeling. Because of the fact that technical keys do not change or change extremely seldom, modeling with hub and links results in one extremely stable and yet flexible architecture. Changes in the structure of the model are implemented mainly by adding new hubs which interact with other with the help of new links without changing the consistency in the rest of the model.

The sixth principle of the Generic Modeling demands that the entities have to be a part of sub/super type hierarchy to determine the universal context of their design. To satisfy this principle and extend the meaning of the relation in the model will be added the entities H_RALATION and type of relation hub H_RELATION_TYPE, as well as the links L_OBJECT_RELATION_TYPE, L_OBJECT_RELATION and L_OBJECT_RELATION_TYPE representing the relations, shown on fig. 5:

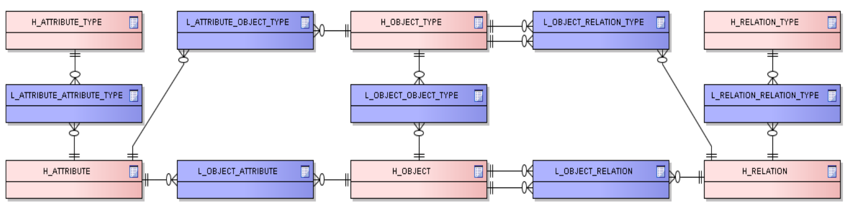

Figure 5 Second Draft of the Generic Metadata Model

The hubs **H_RELATION_TYPE** and **H_RELATION** are used to classify all possible kinds of references - hierarchies, associations, relations, connections, etc. between the entities **H_OBJECT** and **H_OBJECT_TYPE**. The relations will be provided by the hub **H_RELATION** and the links **L_OBJECT_RELATION** and **L_OBJECT_RELATION_TYPE**, which will be done by double referencing the hub with **H_OBJECT** and **H_OBJECT_TYPE**. The double reference will be done with two foreign keys in the link, both of which will referencing the primary key of the corresponding hub. With the use of hub **H_RELATION** the reference will be performed in a certain context, which satisfies the requirement of universal context of design.

A detailed presentation of the link L_OBJECT_RELATION with the hubs **H_OBJECT** and **H_RELATION** is shown on fig. 6:[19]

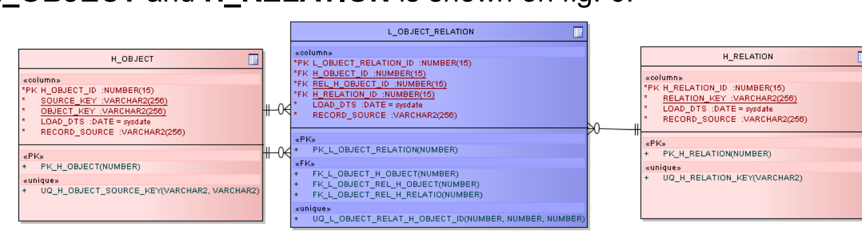

Figure 6 Second Draft of the Generic Metadata Model

Hub **H_OBJECT** has an artificial primary key **H_OBJECT_ID** and one composite business key[20] consisting of **SOURCE_KEY** and **OBJECT_KEY**. **SOURCE_KEY** is the key of the source from which the object keys have been

[19] The link **L_OBJECT_RELATION_TYPE** is modeled in the same way.
[20] The combination of **SOURCE_KEY** and **OBJECT_KEY** must be unique.

imported. The key is composite because the business keys of different objects from different sources may have the same values. **H_OBJECT** is the only hub with a composite business key. All other hubs have one artificial primary key, one business key and two technical attributes **LOAD_DTS** and **RECORD_SOURCE**, which are common attributes [21] in the Data Vault modeling representing the date on load and source of the data.

Link **L_OBJECT_RALATION** has one attribute **L_OBJECT_RELATION_ID** which plays the role of an artificial primary key and a combination of three foreign keys, one to the primary key of hub **H_RELATION** and two to the primary key of hub H_OBJECT (**H_OBJECT_ID** and **REL_H_OBJECT_ID**).[22] This modeling technique could be applied to model a hierarchy with unlimited depth and structure. To demonstrate this approach let's model an organization with the following characteristics:

- the organization O1 is divided into two main departments D1 and D2.
- the business activity of department D2 take place in two regions, which are managed by the units U1 and U2.
- with expanding the organization in the future an emergence of new departments and units is expected.
- both new and old departments and units may also have their subdivisions.
- the organization's structure should remain flexible, allowing reformation of departments, units, divisions and subdivisions.
-

The model of the described above structure is a hierarchy with levels of certain elements that can be represented as follows:

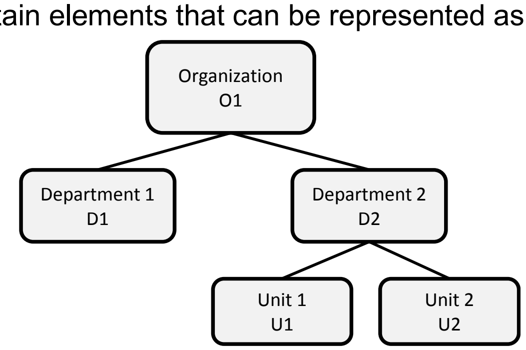

Figure 7 Organization's Structure: Organization, Department, Unit

[21] **LOAD_DTS** and **RECORD_SOURCE** are used in all hubs, links and satellites in the model.
[22] The other links in the model have the same standardized structure similar to **L_OBJECT_RALATION**.

The metadata for the model shown on fig. 7, will be inserted in link **L_OBJECT_RALATION** and hubs **H_OBJECT** and **H_RALATION** is presented on fig. 8:

H_OBJECT

H_OBJECT_ID	SOURCE_KEY	OBJECT_KEY	LOAD_DTS	RECORD_SOURCE
1	XYZ	O1	01.01.2012	FINANCE
2	XYZ	D1	01.01.2012	FINANCE
3	XYZ	D2	01.01.2012	FINANCE
4	XYZ	U1	01.01.2012	FINANCE
5	XYZ	U2	01.01.2012	FINANCE
...	...	...	...	...

L_OBJECT_RELATION

L_OBJECT_RELATION_ID	H_OBJECT_ID	REL_H_OBJECT_ID	H_RELATION_ID	LOAD_DTS	RECORD_SOURCE
1	1	2	1	01.01.2012	FINANCE
2	1	3	1	01.01.2012	FINANCE
3	3	4	1	01.01.2012	FINANCE
...	3	5	1	01.01.2012	FINANCE
...	...	...	...	...	...

H_RELATION

H_RELATION_ID	RELATION_KEY	LOAD_DTS	RECORD_SOURCE
1	R1	40909	INTERNAL
...	...	...	...

Figure 8 Organization's Structure: Organization, Department, Unit

In hub **H_OBJECT** is inserted the metadata of organization, departments and units. The primary keys are doubled referenced in the link **L_OBJECT_RALATION** as foreign keys, modeling a hierarchy between them. In **H_OBJECT_ID** are placed the keys of the parent element, while the **REL_H_OBJECT_ID** these of their direct descendants. A reference to **H_RALATION** classifies the relationship between the object and its related object as “parent-child”. By expanding the organization's new departments and units will be added at the appropriate level and their metadata will be loaded accordingly.

Presented at this stage entity types are the hub and the link that contain only business keys of real objects and the relations between them. Metadata however comprises not only business keys, but many additional attributes. Typical of these attributes is that they describe the real objects and their references and change over time. In the Data Vault modeling all attributes of this type are placed in the satellites, which are referencing a hub or connection. On fig. 9 are shown the corresponding satellites for the hub **H_OBJECT** and link **L_OBJECT_RALATION**:

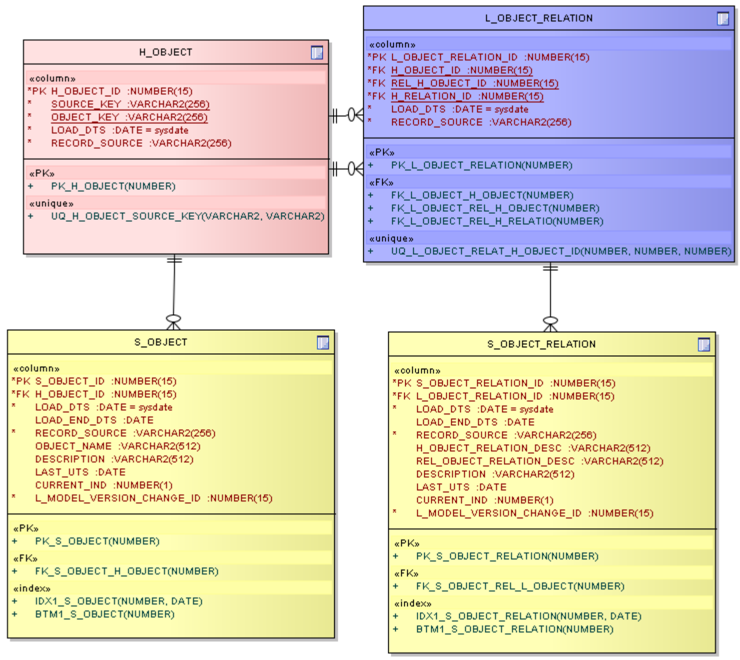

Figure 9 H_OBJECT, L_OBJECT_RALATION, S_OBJECT and S_OBJECT_RALATION

The cardinality of hubs and links to their satellites is **(1:n)**, every primary key has at least one reference key in its satellite. The satellites are built on the principle of slowly changing dimension (SCD 2) and contains the entire history of the hub or connection. With the satellites to their corresponding hubs and links the model can be presented in its final shape, which is shown on fig. 10

The presented satellite's attributes can be extended for the different requirements and needs. In addition there could be more satellites for one hub or link, containing completely different business attributes. This approach is often applied for the data which is coming from different sources.

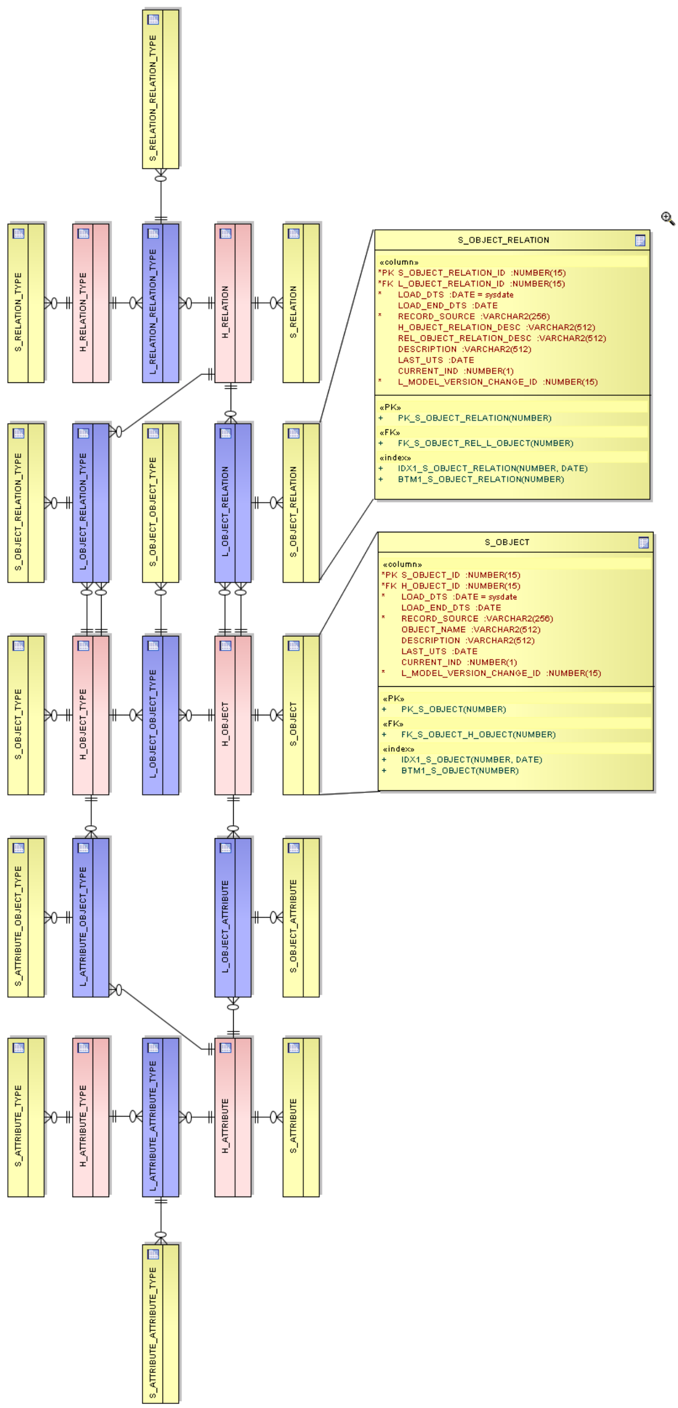

Figure 10 Generic Model for Metadata Repository

11 Concluding Remarks

With adopting of the presented generic metadata repository the organizations may lay the foundations of a centralized and efficient management of all types of models and metadata. Thus it is possible cataloging of technical and business processes in the organization, monitoring the structural dependence and changes, restoring their old versions and changes. The presented model has been designed generic and can be extended or adapted to the requirements of the organization. Standardized structure and typing of the entities contribute to rapid orientation in the model and allow the establishment of standard methods and processes for their loading and query. This in turn reduces the cost of implementation and administration and contributes to optimal use of available resources, which are often quite limited in the organizations. Generic concept of modeling is suitable not only for metadata repository, but also for models that will be created for transactional and master data.

References

Book:

[1] D. Marco. *Building and Managing the Metadata Repository: A Full Lifecycle Guide*. John Wiley & Sons, Inc., 2000.

Technical report:

[2] J. Fowler. *Step for data management, exchange and sharing*. Technical report, Technology Appraisals, 1995.

[3] M. West. *Developing high quality data models.* Technical report, Shell International Limited, 1996.

Web page:

[4] ISO 15926-4:2007 *Industrial automation systems and integration -- Integration of life-cycle data for process plants including oil and gas production facilities -- Part 4.* 12.2012. http://www.iso.org/iso/catalogue_detail.htm?csnumber=41329

[5] ISO 10303-233 *Industrial automation systems and integration -- Product Data Representation and Exchange*. 11.2012. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=55257

[6] ISO 10303-11:2004 *Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual* 12.2012.

http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38047

Book:

[7] D. Linstedt, K. Graziano, and H. Hultgren. *The New Business Supermodel. The Business of Data Vault Modeling*. John Wiley & Sons, Inc., 2000.

[8] D. Linstedt. *Super Charge Your Data Warehouse*. Lulu, second edition, August 2009.

[9] R. Kimball. *The Data Warehouse Toolkit*. John Wiley & Sons, Inc., 2000.