An Approach to Data Mart Design from a Data Vault

Dragoljub Krneta
Lanaco
Information Technologies
Banja Luka, Bosnia and Herzegovina
dragoljub.krneta@lanaco.com

Vladan Jovanovic
Georgia Southern University
Allen E. Paulson College of Engineering and Information
Statesboro, Georgia, USA
vladan@georgiasouthern.edu

Zoran Marjanovic
University of Belgrade
Faculty of Organizational Sciences
Belgrade, Serbia
zoran.marjanovic@fon.rs

Abstract - **The paper presents an approach to the physical design of data marts. A simple direct algorithm is defined using data warehouse metadata model and rules. Our approach benefits from advantages that the Data Vault model brings to data warehousing, namely scalability, flexibility and increased performance on data loads not possible with traditional enterprise data warehouses (EDW). The motivation for our approach was to extend such benefits to the user-oriented data marts side by allowing design automation of initial set of star schema type data marts. The scope of the approach is limited to the Data Vault type of EDW.**

Keywords - **Data mart, Data Vault, design automation**

I. INTRODUCTION

Data warehouse (DW) is a subject oriented, nonvolatile, integrated, time variant collection of data in support of management's decisions [1]. At the core of most business intelligence applications, data warehousing systems are specialized in supporting decision making [2]. The Extract, Transform, Load (ETL) process involves fetching data from transactional systems, cleaning the data, transforming data into appropriate formats, and loading data into a warehouse [3]. A data mart contains summarized data at a certain level of the hierarchy. Unlike transactional data sources which are designed in a relational schema, a data mart is designed as dimensional schema for easier access to data [4]. Data warehousing includes enterprise data warehouses (EDW), data marts, and applications that extract, transform, and load data into the data warehouse or a data mart [5]. An important element of the DW is metadata, including definitions and rules for creating data [6]. The multidimensional model views data as consisting of facts linked to several dimensions. A fact represents a focus of analysis and typically includes attributes called measures. Measures are usually numeric values that allow quantitative evaluation of various aspects of an organization to be performed. Multidimensional data analysis or On-line Analytical Processing (OLAP) offers a single subject-oriented source for analyzing summary data based on various dimensions [7].

Data warehouses can be distinguished by the type of architecture. Bill Inmon [1] [10] proposed the Corporate Information Factory (CIF) as an integrated DW, i.e. database in the third normal form, from which multidimensional data marts are to be derived. The second option is bus architecture, defined by Ralph Kimball [11] where a DW is just a collection of data marts with conformant dimensions.

Data Warehousing 2.0 (DW 2.0) is a second-generation attempt to define a standard Data Warehouse architecture. One of the advantages introduced in DW 2.0 is its ability to support changes of data over time [16]. Data modeling techniques for the data warehouse include the need for an integrated, non-volatile, time-variant, subject-oriented, auditable, agile, and complete store of data. To address these needs, several new modeling approaches have been introduced. Among these are Anchor Modeling and Data Vault modeling [12]. Anchor Modeling is database modeling technique built on the premise that the environment surrounding a data warehouse is in a constant state of change. A large change on the outside of the model will result in a small change within [13]. The Data Vault approach [16] [17] addresses problems of flexibility and performance, enabling maintenance of a permanent system of records [18]. Data Vault (DV) model is recognized by the modeling style using Hub, Link, and Satellite entities [17]. Hubs represent a list of unique stable business keys unlikely to change over time (i.e., Suppliers or Parts) [19]. Link entities are a representation of foreign key references typically used to represent transactions between two or more business components (i.e. Hubs). A Satellite entity shows context information (i.e. attributes of a Hub or a Link) [29]. Data Vault and Anchor models are characterized by strong normalized data and insensitivity to changes in the business

environment as well as the modeling of small parts of the data model, without the need for redesign. Sixth Normal Form (6NF) is a term used in relational database theory by Christopher Date [15] to describe databases which decompose relational variables to irreducible elements. While this form may be unimportant for non-temporal data, it is certainly important when maintaining data containing temporal variables of a point-in-time or interval nature [14] [15]. The Data Vault model, in the case where a satellite table consists of one attribute, becomes a 6NF design. According to [17] and [20], a Data Vault makes most sense in the case of distributed data sources. The example of a Data Vault in Figure 1.

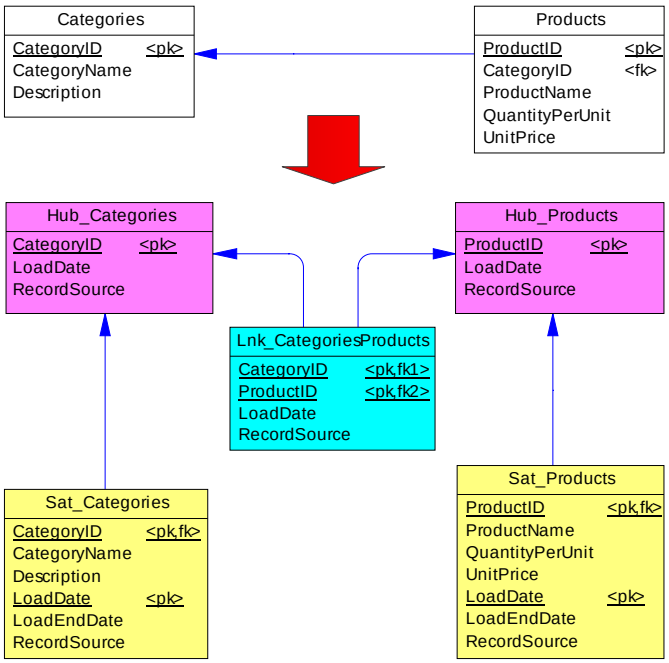

Fig.. 1. An example of mapping a relational to the Data Vault physical model

The Data Vault type DW is a solution for integrating, storing, and protecting data. However, it is neither intended nor suitable for intensive queries or reports. That is why the DW architecture with a Data Vault layer also contains a data mart layer using the star schemas for reporting data access [21]. According to [22], the dimensions of the star schema are resulting from the Hub and their Satellite tables, and the fact tables from a Satellite and corresponding Link tables. Organizations are building DW using different methodological approaches. One practical problem is that the process of designing a DW and data marts is not sufficiently automated.

The rest of the paper is structured as follows: section 2 reviews related work, section 3 presents a novel approach to physical data mart design automation from a Data Vault, and section 4 is composed of a conclusion and outline of future work.

II. RELATED WORKS

The papers focused on design automation for data warehouse content are selected for relevancy, taking into consideration perceived contribution to the theory and practice of design, and indirectly verified by the number of citations (www.harzing.com).

Golfarelli, Maio and Rizzi [8] suggested that a conceptual modeling of the database and a semi-automatic sequence hybrid approach of the warehouse design start by using E/R documentation. The Dimensional Fact Model (DFM) overcomes the gap between the facts and the scheme of the data warehouse. Logical design should be based on the evaluation of the expected load and the amount of data. The paper adduces the possibility of combining special dimensions that are not present in E/R diagram. The selection of aggregated facts should be implemented in a constellation of the scheme of separated DW and integrated data marts [23]. In the cases where E/R documentation is not available but the data exists, suggestion is to begin working from the relation database scheme. After the scheme analysis and the creation of a conceptual or a logical model, identification of the facts is suggested. After this step, semi-automatic forming of the attribute tree in line with demands will follow. The next step is the identification of dimensions, measures, and aggregate functions. The last step in this process is the derivation of a logical and physical data warehouse scheme. Revision and improvement of this procedure is given in [23].

The illustration of the algorithm for automatic generation of a conceptual scheme with the use of operational database scheme is given in [24] (Phipps and Davis). A data-driven approach is a starting point, and later on a demand-driven approach is used (making it a sequence hybrid approach). The initial conceptual scheme is made using the Multidimensional E/R model. An algorithm is illustrated using TPC-H (Transaction Processing Performance Council - ad Hoc) Benchmark schemes and queries. This paper brings an algorithm for deriving a conceptual scheme from OLTP scheme. Algorithm use numerical fields and relationships between entities as the base for creation of a Multidimensional E/R scheme. The paper represents a significant improvement in automation of data warehouse design.

The problem of data warehouse design, with improvement of the logical design process as the main goal, is given in [25] (Peralta et al.). A mechanism is suggested for gaining a logical scheme of the DW through earlier defined transformation of source logical scheme which can be used to supplement existing design methodologies. The transformation enables better logical design of DW and enables tracking design and source mapping, and logical structure of the DW. This track is very important, because it enables mapping between the sources and DW elements. Mapping is relevant for the identification of loading data and debugging problems.

Romero and Abello in [26], represent a new approach in the automation of multidimensional data warehouse design. The paper suggests a semi-automatic method with the goal to find multidimensional business concepts from ontology domain, which represents different and potentially heterogeneous data. This approach is capable of integrating the data from

heterogeneous data sources, which describes their domains through ontologies. One of the most prospective areas where this method can be applied is the semantic web, which contributes the integration of external data from the web in the data warehouse.

Krneta, Jovanovic, and Marjanovic [27] presents a novel agile approach to large scale design of enterprise data warehouses based on a Data Vault model. An original, simple, and direct algorithm is defined for the incremental design of physical Data Vault type enterprise data warehouses using source data meta-model and rules, and used in developing a prototype case tool for Data Vault design. This approach solves primary requirements for a system of record, that is, preservation of all source information, and fully addresses flexibility and scalability expectations.

Our direct approach to automatic generation of the Data Mart physical models is based on metadata schemas of a Data Vault DW through the use of rules. It should be noted that the majority of listed approaches are academic, and that only [23] and the Data Vault in [16] [17] and [28], have a tradition of industrial use. A direct physical design for Data Mart is practical only in case of Data Vault separation of storage of identities (permanent keys) from evolving relationships and characteristics (attributes).

III. DIRECT PHYSICAL DATA MART DESIGN

This work is part of a larger research program dealing with the next generation DW design technology [30] where Data Vault is a baseline approach to DW design and most of the requirements are derived from a developer's standpoint.

We propose a direct approach to the design of physical data mart based on the Enterprise Data Warehouse (EDW) Data Vault metadata model. The phases of initial Data Mart design are shown in Figure 2.

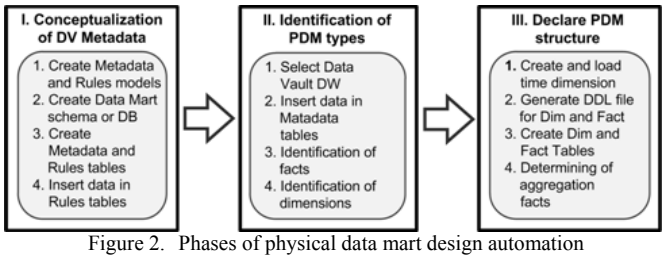

Figure 2. Phases of physical data mart design automation

The name direct approach reflects that the approach directly leads to data mart models based on a physical model of a Data Vault. This approach is feasible because the Data Vault model follows a restricted architecture that only uses Hubs, Links, and Satellites. The entities in the Data Vault have fixed structures that cannot change, and only changes to a Data Vault are additions. A simplified PDM design process is shown in Figure 3.

Data mart design, determination of business measures and dimensions, and designing fact tables and dimension tables are complex processes that largely demand the participation of users. Fact tables contain business measures. The measure is a numerical value that is of central interest for analysis (for example, the quantity of products sold, transportation costs, total revenue, total expenditure, etc). On the basis of these measures, some measures that are not among the original data can be recognized and derived. The literature dealing with the automation of design data warehouse or data marts from the relational model uses several techniques to identify the measures and dimensions.

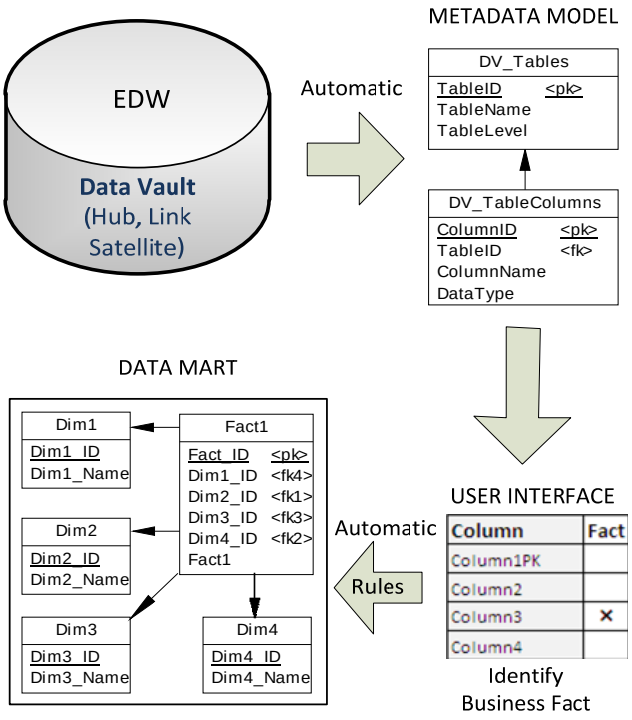

Figure 3. The process of generating data mart from a Data Vault

Fact tables usually have a connection relationship of many-to-one to the related dimension tables [26]. According to [11], the performance of fact tables from relational model is based on the selection of a many-to-many table from ER model which contains numeric facts that are not keys. Golfarelli, in [8], saw good candidates for measures in the entities that are most often changed, and numerical columns that are aggregated.

According to [24], numerical attributes may represent measures associated with business events. Therefore, entities with the largest number of numerical attributes are potential fact tables, while other tables are potential dimensions. However, some numerical attributes are not suitable for measures, such as a phone number or zip code [29]. The solution to this problem is the use of knowledge stored in the WordNet semantic lexicon as a knowledge base for the identification of numerical data [31], which may be a business measure.

The rule in [8], states that good candidates for measures are entities that are commonly changed and can be applied only to

the transaction type fact tables. Such entities, for example, are invoice items or order items. According to [32], entities in a transactional database can be divided into transactions, components, and classifications. Transaction entities describe business events and represent the best candidates for facts. Entities from components describe the situation associated with the transaction, such as product, time, location, etc. and represent good candidates for dimensions [31]. The choice of dimensions is crucial for the design of data warehouse and the ETL process. It is widely known that time is one of the critical dimensions for the ETL process. When designing a data warehouse, time is explicitly presented as a relational attribute and therefore is an obvious candidate for dimension [9].

In the literature relating to design of Data Vault systems ([16] [17] [20] [21] [28]), the rule for transforming data to a data mart dimensional model is: the dimensions of the star scheme are resulting from Hub tables and their Satellites, and measures (fact) are resulting from the Link tables and their Satellites. Recommendations for the reverse process, transform Data Mart from Star Scheme to Data Vault model is given in [33]. In this case, it is necessary to locate one or more columns in the dimensions that constitute the business keys. If the observed columns are independent, it is necessary to make a separate table Hub.

The direct approach to Physical Data Mart (PDM) design automation presented in this paper includes use of rules for data mart design. To understand the detailed rules we first outline the general algorithm for recognition of Fact and Dimension tables relations based on the mapped Data Vault model. Taking into account that the time dimension is the only dimension that is almost always there in every data mart, the algorithm consists of the following steps:

(i) For each Data Vault table
  (a) For each Link table
    - For each Satellite of Link: user confirms fact column by selected business fact
  (b) For each Hub table
    - For each busines key column of Hub: system confirms dimension by selected business key
    - For each Satellite of Hub: user confirms aditional dimension by selected other Satellite attribute(optional)

(ii) Creating and filling time dimension table

(iii) For each confirmed fact
  (a) User determined aggregation of facts

A. *Conceptualization of metadata – phase 1*

The main phase is the Conceptualization of Metadata resulting in a physical data model that can be used in designing individual Data Mart. When making the meta model and rules, we distinguish between the part of the model that is independent of the data source (source-independent design) and part of the model which depends on the source (source-specific design). Independent part refers to the metadata tables and rules, and dependent part concerns procedures related to different platform of data warehouse. The RuleTypes table contains information about the types of policies such as rules for different Data Vault table, rules for identifying Fact and Dim tables, and rules for creating Fact and Dim tables. The Rules Table contains information about a particular type of rules and actions to be performed when a certain condition is satisfied. Rules in column Rules allow easier execution of commands that are stored in the column RuleAction.

The first step is building a model of the metadata and of the modeling rules (Figure 4).

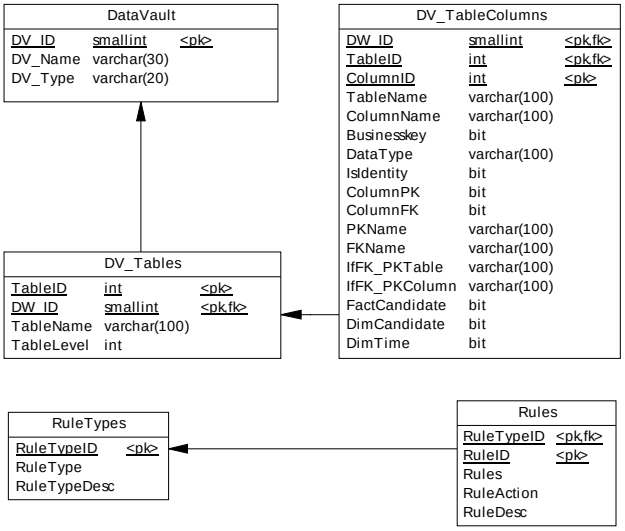

Figure 4. Physical metadata Data Vault model and model of rules

The following example is given to show the SQL rule to describe a set of constraints that are applied to identify dimension candidate based on the value of the column BusinessKey:

```
UPDATE DV_TableColumns SET DimCandidate=1
WHERE BusinessKey = 1
```

The second step involves creating a data mart schema (or data mart database). The third step creates metadata, a rules table and procedures on appropriate data mart schema or database. The last step of this phase involves loading data in rules tables. After the first phase and creation of tables and appropriate procedures, conditions are created for the development of an application that allows one to automate the design of the physical model of data mart with minimal user interaction.

B. Identification of physical data mart types – phase 2

Taking into account that the time dimension is the only dimension that is always present in every data mart, this phase consists of the following steps:

The first step involves the identification of a Data Vault DW, by a user selecting specific database (through the user interface). The second step of this phase (in figure 2) inserts metadata in metadata tables of the selected Data Vault DW.

The third step involves determining business fact (fact table). The user needs to determine business facts (through appropriate user interface). Based on certain business measures filled in FactCandidate column with the appropriate value (0 or 1). Loading these columns can be automatically based on rules stored in the table Rules (Figure 4). Specifying the business measure identifies the fact table. The measure is mainly numerical or data types: int, money, real, smallint, decimal [8] [9] [10] [11] from Sat_ tables that are not PK or FK [22] [33]. Mentioned potential measures are mainly found in the tables at the lowest hierarchical level (orders item, invoice items, etc.) in the transactional database. For ease of identification of measures, in the approach outlined in this paper, a user is initially offered only data from DV_TableColumns table that have this specified data type, and the data is sorted by the hierarchical level of the table. In addition, in this section system offers to a user the name of the corresponding fact table.

The fourth step of this phase involves determining business dimensions and dimension table, including the time dimension. After measures have been set up, the system, based on business keys from hub tables, defines table dimension, setting the value of DimCanditate to 1 for appropriate columns from hub tables and their satellites. Beside the business key, tables of the dimensions should include description data from specific satellite tables. Through a suitable user interface, the user will be offered suggested dimensions or option to choose different dimensions from specific satellite tables.

C. Declare physical data mart structure – phase 3

The last phase in the process of automation of the physical design of data mart is the initial declaration of the PDV structure. This phase include following steps:

(i) *Create and load time dimension***.** The first step in this phase is to create and load the time dimension table. Start date and end date of the time period are based on minimum and maximum date from DimTime column on table and used for the time dimension (for example InvoiceDate).

(ii) *Create DDL (Data Definition Language) file for fact and dim tables***.**

(a) When we have the business fact, appropriate tables can be identified (by setting the value 1 in column FactCandidate). It is possible to generate a script to create a fact table in a data mart, based on the rules in the Rules table. This step accomplishes the following:

- Form a cursor to go through the DV_TableColumns Where FactCandidate=1
- Retrieve data and assign variables
- In each iteration use dynamic SQL to supplement sql_statement for create table DDL

(b) Based on a business key the column DimCandidate will be filled. Filling in this section can be automatically based on rules stored in the table Rules, according to [17] and [33]. We identified the appropriate dimension tables containing value 1 in column DimCandidate which enables generation script to automatically generate dimension table, based on the rules in the Rules table. This step provides the following:

- Form a cursor to go through the table DV_TableColumns Where DimCandidate=1
- Retrieve data and assign variables
- In each iteration use dynamic SQL to supplement sql_statement for create table DDL

(iii) *Create dimension and fact tables***.** After creating DDL file for fact and dimension table, it needs to execute scripts. First, execute the script to create the dimensions tables. After that, execute the script to create the fact table:

(a) Execute a sql_statement, creating a dimension table.

(b) Execute a sql_statement, creating a fact table.

(iv) *Determining aggregation facts***.** At the end of this phase, the user needs to choose the measures which will be aggregated and the method of the aggregation of the measures. For example, if we have a table with the details of the order in which the amount of sold products and unit prices are listed, it is necessary to define the amount to be added. The amount of sold goods would be calculated as quantity times the unit price.

Procedures described in the third phase make it possible to automatically create a complete data mart based on the Data Vault concepts. Loading schema and transforming it following pre-programmed rules certainly supports design performance, scalability and agility (user intervention is minimal but necessary, and is mainly focused on recognizing major business facts). The second and third phases of Figure 2 need to be done for each specific system.

IV. CONCLUSION

The Data Vault type data warehouse is a modern solution for integrating, storing, and protecting data. However, it is

neither intended nor suitable for intensive queries or reports. That is why the data warehouse architecture with a Data Vault layer (persistent system of records with full history of changes) also contains a Data Mart layer using the star schemas for data access, see a detail example in [34].

In this paper a simple algorithm for the design of physical data mart is presented. The algorithm is predicated on a Data Vault type data warehouse as the source. Our approach is based on using the metadata model and rules. Relationships between entities in the Data Vault data warehouse and rules for the development of a data warehouse are necessary for physical design automation i.e. derivation of a data mart model. The conceptualization of metadata presented a physical model that can be used in the design of an individual data mart. An important innovation is realization of a data mart schema directly from Data Vault schemas. Such a direct approach was possible thanks to the feature of the Data Vault model i.e. separation of unchangeable identities of entities in real systems (Hubs) from time variant relationships among such entities (represented by Links) and the characteristics of such entities and their relationships (represented by Satellites).

The Data Vault provided a valuable capability to integrate data by adding links while preserving all source data essentially in its original form. Traditional approaches to data integration in data warehouse and data mart design frequently used [23] pruning of data from the original data source(s) and additional typically irreversible transformations i.e. consolidation that requires complex intervention by experts (due to creative and semantically rich transformations) and very clearly leads to a loss of auditing capability.

This paper also demonstrated scalability, flexibility and utility of the algorithm for designing data marts based on Data Vault type data warehouses. The current direction of our research is further automation of ETL processes into and from Data Vault to feed a data mart.

REFERENCES

[1] H. W. Inmon, Building the Data Warehouse,Wiley Computer Publishing, New York, 1992.

[2] M. Golfarelli, S., Rizzi, A Survey on Temporal Data Warehousing, International Journal of Data Warehouse and Mining, 5(1), 2009, 1-17.

[3] B. Larson, Delivering Business Intelligence with MS SQL Server 2008, McGraw Hill, 2009.

[4] D. Krneta, D. Radosav, B. Radulovic, Realization Business Intelligence in Commerce using Microsoft Business Intelligence, 6th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, 2008, pp. 1–6.

[5] H. J. Watson, Recent developments in data warehousing, Communications of the AIS, 8, 2001, 1-25.

[6] S. Adelman, L. Moss, M., M. Abai, Data Strategy, Addison Wesley, New York, 2001.

[7] T. Niemi, L. Hirvonen, K. Järvelin, Multidimensional Data Model and Query Language for Informetrics, Journal of the American Society for Information Science, 54, 2003, 939–951.

[8] M. Golfarelli, D. Maio and S. Rizzi, The Dimensional Fact Model: a Conceptual Model for Data Warehouses, International Journal of Cooperative Information Systems, vol. 7, 1998, n.2&3.

[9] M. Golfarelli, D. Maio, S. Rizzi, Conceptual Design of Data Warehouses from E/R Schemes, In Proceedings of the 31st Hawaii International Conference on System Sciences, Kona, Hawaii, 1998, pp. 215-227.

[10] W. H. Inmon, Building the Data Warehouse, 3rd Edition, Wiley Computer Publishing, 2002.

[11] R. Kimball, The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses, Wiley Computer Publishing, 1996.

[12] L. Rönnbäck, H. Hultgren, Comparing Anchor Modeling with Data Vault Modeling, Modeling for the modern Data Warehouse, White paper, 2013. http://hanshultgren.files.wordpress.com/2013/06/modeling_compare_05_larshans.pdf

[13] L. Rönnbäck, O. Regardt, M. Bergholtz, P. Johannesson, P. Wohed, Anchor Modeling - Agile Information Modeling in Evolving Data Environments, Data & Knowledge Engineering, 2010, pp 1229–1253.

[14] C. Knowles, 6NF Conceptual Models and Data Warehouses 2.0, Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, March 2012.

[15] C.J. Date, H. Darwen, N. Lorentzos, Temporal data and the relational model: A detailed investigation into the application of interval and relation theory to the problem of temporal database management, Morgan Kaufmann Publishers, Amsterdam, 2002.

[16] D. Linstedt, Super Charge your Data Warehouse, Kindle Edition, 2010.

[17] D. Linstedt, Data Vault Model & Methodology (2011), http://www.learndatavault.com.

[18] V. Jovanović, I. Bojičić, Conceptual Data Vault Model, Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA, 2012, 131-136.

[19] C. Knowles, V. Jovanović, Extensible Markup Language (XML) Schemas for Data Vault Models, Journal of Computer Information Systems, Vol.53, Issue 4, 2013.

[20] K. Graziano, Introduction to Data Vault, 2011, http://www.slideshare.net/kgraziano/why-data-vault

[21] M. Casters, P. Bouman, J. van Dongen, Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration, Wiley Publishing, 2010.

[22] S. Govindarajan, Data Vault Modeling The Next Generation DW Approach, 2010, http://www.globytes.com.

[23] M. Golfarelli, S. Rizzi, Data Warehouse Design: Modern Principles and Methodologies, McGraw-Hill, New York, 2009.

[24] C. Phipps, K. Davis, Automating Data Warehouse Conceptual Schema Design and Evaluation. 4th International Workshop on Design and Management of DW, Toronto, Canada, 2002.

[25] V. Peralta, A. Marotta, R. Ruggia, Towards the automation of data warehouse design, InCo, Universidad de la República, Montevideo, Uruguay, 2003.

[26] O. Romero, A. Abello, Automating Multidimensional Design from Ontologies, Proceedings of the ACM tenth international workshop on Data warehousing and OLAP, Lisbon, Portugal, 2007.

[27] D. Krneta, V. Jovanović, Z. Marjanović, A Direct Approach to Physical Data Vault Design. Computer Science and Information Systems, Vol. 11, No. 2, 2014, 569–599.

[28] R. Damhof, The next generation EDW, Database Magazine, Netherlands, 2008.

[29] A. Battaglia, M. Golfarelli, S. Rizzi, QBX: A CASE tool for Data Mart design, In Proceedings 30th International Conference on Conceptual Modeling, Brussels, Belgium, 2011, pp. 358-363.

[30] V. Jovanovic, Z. Marjanovic, DW21 Research Program-Expectations (White paper), FON/Breza software engineering, Belgrade,Serbia, 2013.

[31] M.N.M Nazri, S.A. Noah and Z. Hamid, Using lexical ontology for semi-automatic logical data warehouse design, 5th international conference on Rough set and knowledge technology, Beijing, China, 2010.

[32] D.L. Moody, M.R.A. Kortink, From Enterprise Models to Dimensional Models: A Methodology for DW and Data Mart Design, International Workshop on Design and Management of Data Warehouse, Stockholm, Sweden, 2000.

[33] D. Linstedt, From Star Schema to Data Vault, 2010, http://danlinstedt.com/datavaultcat/from-star-schema-to-data-vault/

[34] D. Linstedt, M. Olshimke, Building a Scalable Data Warehouse Architecture with DV 2.0, MK 2016