A Framework for Semi-Automated Implementation of Multidimensional Data Models

Ilona Mariana NAGY
Faculty of Economics and Business Administration, Babes-Bolyai University
Cluj-Napoca, ROMANIA
mariana.nagy@econ.ubbcluj.ro

Data warehousing solution development represents a challenging task which requires the employment of considerable resources on behalf of enterprises and sustained commitment from the stakeholders. Costs derive mostly from the amount of time invested in the design and physical implementation of these large projects, time that we consider, may be decreased through the automation of several processes. Thus, we present a framework for semi-automated implementation of multidimensional data models and introduce an automation prototype intended to reduce the time of data structures generation in the warehousing environment. Our research is focused on the design of an automation component and the development of a corresponding prototype from technical metadata.

Keywords: *Data Warehouse; Multidimensional Data Model; Automatic Data Structure Generation; Technical Metadata;*

1 Introduction

Data warehousing projects require the employment of considerable and varied resources on behalf of the enterprises, sustained commitment from the stakeholders and a long period of development time. In most cases, these projects are successful implementations, although there are several reasons that account for failure situations, such as expenditures exceeding the budget and inability to meet delivery deadlines. Companies that undertake data warehousing development need to stay competitive on the market with regard to implementation, maintenance and other similar activities. This competitiveness is ensured significantly by small costs of development and short delivery times; therefore, reducing some of the involved costs augments the chances of successful solution delivery and increases business satisfaction.

Given these circumstances, we introduce a framework for semi-automated implementation of data structures in the data warehousing environment and propose a prototype aimed to partially automate several processes in order to reduce the related costs. Regarding the implementation of the prototype, we make use of the technical metadata provided by the underlying operational systems, external source files and the metadata generated in the data warehousing environment. We also propose a distinction between structural and logical technical metadata, and employ both these types for automating the multidimensional model (i.e. data mart) schema implementation.

Due to the complex environment of the analytical systems and the nature of multidimensional models representation, the development of such a prototype is rather challenging. The first results of our research consist in the successful initial implementation of the prototype in the SAP Business Warehouse (SAP BW) environment. In this paper we present the theoretical and technical aspects of the implementation phase, discuss the arising issues and also propose future development directions. The paper is structured as follows: section 2 presents the overall data warehousing framework architecture with its components and the interaction between them; section 3 focuses on the specifics of the multidimensional model generation component, and also illustrates the process flow within the framework component; section 4 presents the implementation of the proposed prototype; section 5 is dedicated to the study of related work; and section 6 concludes the research and makes suggestions regarding future development.

2. Framework Presentations

The data warehouse architecture, which stands at the basis of this framework, may be decomposed into: storage elements, data handling procedures and the human factor (final users, technical staff) [1].

The proposed implementation framework addresses the semi-automated generation of the storage elements and partly data handling procedures, through corresponding ETL processes, in the data warehousing environment. The storage elements may be found in all architectural layers (data staging, data warehouse, and data mart layers). ETL processes are responsible for extracting data from source systems, transforming it in the staging area and loading the data into the storage elements for permanent storage.

Besides defining the means for automatically generating storage structures (i.e. the data warehouse and data marts schema), the framework also provides the definition metadata mapping between its components, and thus the "map" by which the automation may be achieved.

The framework provides guidance of implementing various storage structures, such as data warehouse and data marts schema, from technical metadata.

Fig. 1 depicts the proposed framework comprising the following components:

- *Data Staging Module*, which has the role of defining an integrated staging area where a single set of transformations (i.e. technical and business rules) are enforced for the entire data warehousing landscape;
- *Data Warehouse Schema Generation Module*, which enables automated generation of specific data storage metadata objects for the data warehouse layer from the technical metadata residing in the repository;
- *Data Mart Schema Generation Module*, which, similarly to the previous component, enables automated generation of the data mart schema from technical metadata;
- *Metadata Management Module*, which defines essential metadata storage and retrieval functions from the metadata repository, and handles the overall data warehousing structures generation process (e.g. replication, schema, scheduling, monitoring);
- *Data Management Module*, which facilitates the management of the data warehouse data.

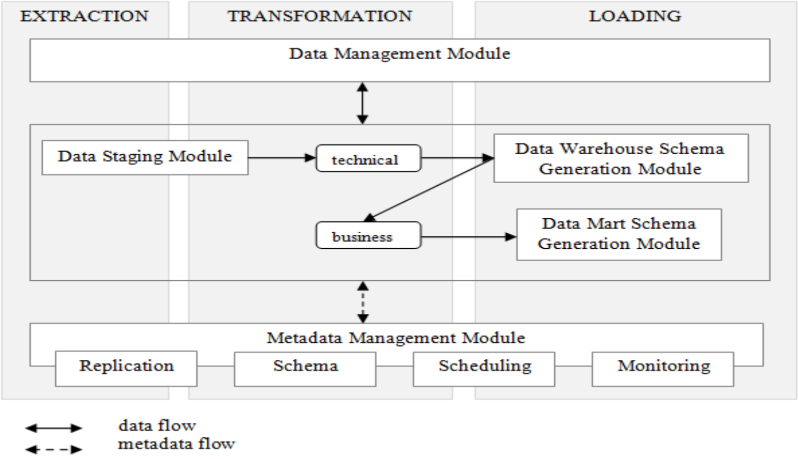

Fig. 1. Framework Architecture Overview

The components of the framework cover well-defined extraction, transformation and loading processes in the data warehousing environment. In order to facilitate a clear understanding of the way components interact, we present a brief description of these processes in the following paragraphs.

Extraction Processes

One of the first tasks performed within ETL processes is the extraction of relevant information that has to be propagated in the data warehousing environment [2]. Extraction processes are meant to handle various and often heterogeneous enterprise-wide sources of data, which may vary from relational databases and flat files, to legacy or hierarchical model structures. The diversity of source systems foremost determines the complex nature of these processes.

Moreover, there are two types of extraction methods defined at the logical level, namely full and incremental extractions. The full extraction method assumes that all the data existent in the source systems at one moment in time is extracted completely in the data warehousing environment, whereas the incremental extraction method employs a well-defined logic by which only most recently inserted or updated records are subject to extraction. The incremental extraction (also known as partial or delta extraction) requires the implementation of advanced change capture techniques. This is usually achieved in operational source systems by logging the changes in data through a timestamp mechanism and by using change tables to track the different insertion, deletion or update operations occurred since the last extraction process. Incremental extraction is normally used on large data sets and is not applicable to external source files, such as flat files. From a physical level perspective, there are two types of data extractions: online extraction (in which data is retrieved by directly accessing the source systems) and offline extraction (in which data is staged explicitly outside the original system in flat files, archive logs or transportable tablespaces, etc. 3]).

Generally, the extracted data is temporarily stored into a physical storage environment, known as data staging area, from where data is further processed and loaded into permanent storage structures.

Transformation Processes

Transformation processes are defined as part of the overall data cleaning approaches in the data warehousing environment, among data analysis, definition of transformation workflow and mapping rules, verification, etc. [4]. Their main goal is to produce a collection of consolidated and integrated data that conforms to well-defined standards, and may provide useful information for supporting business users' decision making.

Within the ETL processes, the transformations performed are reflected at the instance level and at the schema level. The instance level transformations concern data conversions and include various individual tasks, such as cleaning of misspellings or missing data values, resolution of conflicting representations of the same data element, elimination of duplicated data records, etc. At the schema level, transformations comprise technical standardizations of data types, field lengths, etc. and semantic standardizations, such as resolution of synonyms (two or more terms refer to the same concept) and homonyms (one term refers to different concepts) 5]. Other transformation processes may include combining multiple data records into a single data record, separating one data record into multiple ones, sorting, filtering and merging operations, applying complex conversion functions or formulas, assignment of surrogate keys, etc.

Metadata, both technical and business, has an essential role in guiding these transformation processes in the data warehousing environment. Metadata is used to define data characteristics, transformation mappings, workflow definitions, etc. [4] at the technical level, and to facilitate the understanding of converted data at the semantic level.

Loading Processes

The loading of data into the final permanent storage structures represents the last series of activities that comprise the ETL processes. Following the logical extraction methods, there are two distinct types of data loading: initial loading, normally performed the first time the data warehouse is deployed in production, and incremental loading, which represent on-going processes that occur daily, weekly, or monthly, as defined by business requirements. Once loaded into the data warehouse specific structures, the data is ready for analysis and reporting.

The presented ETL processes are part of data warehousing solution development. Their design and implementation requires extended business knowledge and correlation with business requirements. However, these processes may be to some extent automated by various generic tools available on the market, as well as by proprietary programs. In order to enable full support of various scenarios and deliver maximum value to business users, ETL tools should comply with a series of requirements, such as 6]: connectivity/adapter capabilities, data delivery and transformation, data and metadata modelling capabilities, data source and target support, data governance, design and development environment, operation and administration, architecture and integration, etc.

Having presented an overview of the overall framework architecture, as well as the various processes that stand at the basis of storage structures generation, we proceed to introducing our proposed multidimensional model schema generation component, and its implementation in the data warehousing environment.

3. Data Mart Schema Generation Module

The *Data Mart Schema Generation Module* is designed to enable faster implementation of the multidimensional data models in the data warehousing environment.

In order to achieve the desired degree of automation, we focus on capturing, storing and retrieving technical metadata within the framework's components. Moreover, as a prerequisite, we propose a differentiation of the technical metadata used and generated by the framework modules, into structural and logical metadata. Our reasoning is explained as follows:

- The *structural metadata* defines technical characteristics of the data structures, such as source systems, comprised fields, keys, data types and lengths, etc. These properties are stored in the data warehousing system independently of the way these data structures are used (e.g. as sources of data for other storage objects).
- The *logical metadata* describes the relationships (i.e. logical mappings) between the various metadata objects and storage structures of the data warehousing system, and is derived from the logical map of the frameworks components' physical model. It also comprises the logical multidimensional data model and all the other technical metadata provided by the data architect for automated generation purposes. Additionally, this type of technical metadata provides information about the data provenance and lineage (i.e. the source and path followed from source to target) [7].

When using technical metadata within the context of the defined framework, for automation purposes especially, we refer to the previously introduced two sub-types of metadata, namely structural and logical metadata.

Data Marts represent essential storage structures of departmental data in the enterprise's informational systems. They generally correspond to single business processes in the operational systems and are built as multidimensional data models in the data warehousing environment for supporting high performance querying and optimized on-line analytical processing. Due to their particular design (i.e. star schema layout with one central fact table surrounded by several dimension tables), the number of join operations required for data retrieval is reduced to a minimum, which makes these storage structures ideal for interrogation. Moreover, data marts store aggregated and highly indexed data, supplied by the integrated and consolidated data warehouse layer. This data goes through a process of business transformations, as depicted in Fig. 1, during which more complex formulas are applied for defining key performance indicators as well as other facts of interest for the analysis process.

Among the main reasons for building data marts in the analytical environment, we mention:

- data marts are created as part of a comprehensive data warehousing solution following the development of the underlying data warehouse layer, which provides them with a foundation of highly granular, historical, integrated and consolidated data; this improves consistency and accuracy in data representation across the enterprise;
- business users or department specific requirements may be easily accommodated in the analytical environment by means of various data marts built on the solid data foundation;
- data marts lift the burden of direct analytical processing on the data warehouse data;

Considering the undeniable importance of data marts in the data warehousing environment, we propose a schema generation module that speeds the development process by automating the implementation of multidimensional data models from technical metadata. Fig. illustrates the various components that interact with the *Data Mart Schema Generation Module*.

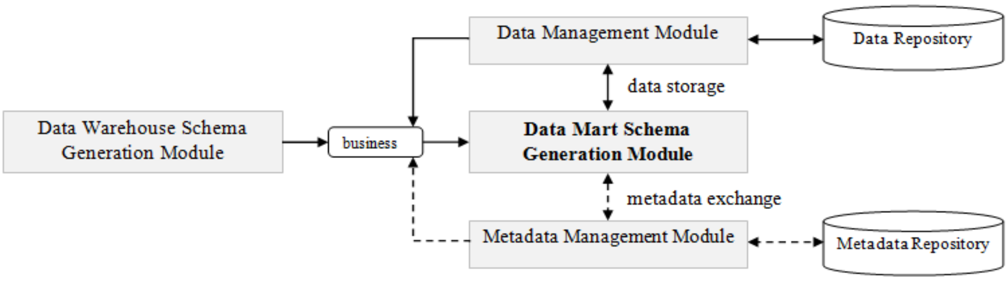
Data Management Module
Data Repository
data storage
Data Warehouse Schema Generation Module
business
Data Mart Schema Generation Module
metadata exchange
Metadata Management Module
Metadata Repository

Fig. 2. Framework Architecture: Data Mart Schema Generation Module

In order to accomplish the schema generation, the proposed module exchanges technical (i.e. structural and logical) metadata with the Metadata Management Module; this consequently retrieves and stores metadata into the metadata repository. The module uses technical metadata to generate empty schemas of the multidimensional model. In the context of data mart schema generation, structural metadata is represented by the tables, columns and the data type definitions of the corresponding structures, whereas the logical metadata is represented by the multidimensional data model (i.e. fact table, dimension tables, relationships between them).

The *Data Mart Schema Generation Module* is supplied with highly granular integrated data from the data warehouse repository. However, the data is subject to comprehensive transformations based on the business logic defined by user requirements and stored as business metadata in the repository.

Fig. 2 depicts the process flow within the Data Mart Schema Generation Module comprising six steps:

- The first step covers the loading of the XML file providing structural and logical metadata into the system. This operation is facilitated by the specifics of each implementation platform.
- In the second step, the XML file is processed and useful information is sorted out. This means that the XML file is parsed and analysed so that relevant information regarding the structure of the multidimensional data model is extracted from its content (e.g. the fact and dimension tables, the relationships between them, the fields and corresponding data types, etc.). Subsequently, the technical metadata is stored in the repository, from where it is further retrieved by the *Metadata Management Module.*
- Within the third step, the technical metadata (i.e. structural and logical) is retrieved and processed by the *Data Mart Schema Generation Module* through its interaction with the *Metadata Management Module.*
- The fourth step defines the actual schema generation process from the processed structural metadata, as indicated by the logical metadata.
- In the fifth step, the generated multidimensional schema is checked and validated against inconsistencies of technical nature specific to the implementation process (e.g. valid technical names for the fact and dimension tables, maximum number of dimensions accepted, appropriate definition of relationships based on primary-foreign key definition, etc.).
- The final step represents the communication of the schema generation results (e.g. logging of processing steps in the system, user interface, etc.).

Having defined the architectural aspects of the proposed framework, along with its main components and processes, we proceed to presenting briefly the platform - dependent prototype implementation. The technologies used for the implementation processes are: SAP R/3 as operational source systems, SAP Business Information Warehouse as analytical system, and the ABAP programming language.

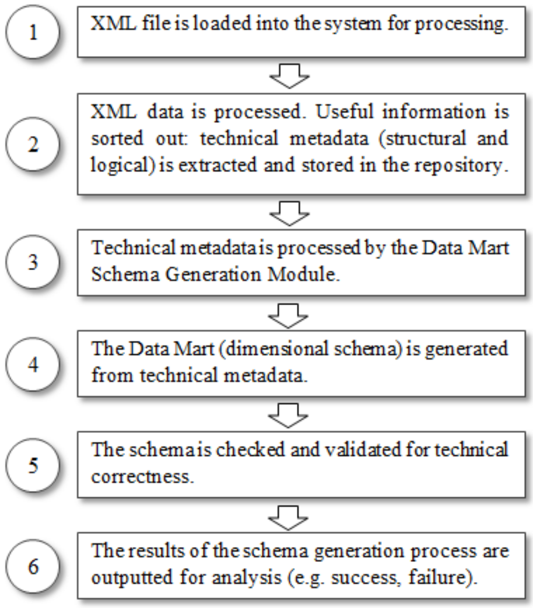
1
XML file is loaded into the system for processing.
2
XML data is processed. Useful information is sorted out: technical metadata (structural and logical) is extracted and stored in the repository.
3
Technical metadata is processed by the Data Mart Schema Generation Module.
4
The Data Mart (dimensional schema) is generated from technical metadata.
5
The schema is checked and validated for technical correctness.
6
The results of the schema generation process are outputted for analysis (e.g. success, failure).

Fig. 2. Data Mart Schema Generation Process Flow

4. Framework-based Prototype Implementation

As prerequisite for prototype implementation, a Unified Modelling Language (UML) Class Diagram portraying the dimensional data model to be implemented as data mart storage structure is required. We chose the modelling of the data model schema with UML as this represents a standardized general-purpose modelling language extensively used in software engineering (aspects of dimensional data modelling with UML are presented in [8]). Moreover, numerous CASE (Computer Aided Software Engineering) tools enable the exporting of class diagrams as Extensible Markup

Language (XML) files, a commonly-used standard defined by a set of rules for encoding document.

Considering the sub-components of the automation framework, presented in Fig. 2, and their interdependence, the multidimensional data model implementation begins with the interpretation of the exported XML format class diagram from the CASE tool.

An XML parser/analyser retrieves the entities with corresponding attributes and data types, as well as the relationships between them (the multidimensional schema design), and maps the extracted information to the logical metadata represented in the repository as relational metadata tables. These represent flat relational database tables whose structure contain relevant fields, such as: entity type (fact table or dimension), automatically generated technical name, descriptive name, corresponding attributes, attributes data type and length, position in the entity, etc. The result of the XML analyser component execution is the physical storage of the multidimensional data model in relational form in the metadata repository of the analytical environment.

A subsequent action in the multidimensional generation process is represented by metadata processing, as depicted in the third step of the Fig. 2 process flow. The prototype retrieves the technical metadata from the repository (i.e. the multidimensional data model stored previously) and passes it over to the following step, namely to the multidimensional structure generation.

This data mart schema generation is implemented with by number programs, functions and system classes. Its goal is to enable the creation of platform-specific metadata objects that provide structural and physical representation of the multidimensional logical data model in the data warehouse environment.

The specifics of the implementation are listed below:

- The multidimensional model is implemented in the data mart layer of the SAP Business Warehouse, through an InfoCube structure. The InfoCube consists of several types of atomic components (InfoObjects), which form the fact table and the dimensions of the model. Internally, the cube is comprised of relational tables arranged together in a star schema, where the fact table contains key figures and the dimension tables are linked to master data tables. InfoCubes are normally supplied with data from data structures of the lower architectural levels, such as DataSources, InfoSources, master data InfoObjects, and DataStore Objects (the existence of data supply structures is ensures by the implementation of the extended version of the prototype).
- The information required for automated implementation of the multidimensional model is read from the logical metadata stored in the repository. Corresponding InfoObjects, which form the structure of the InfoCube are retrieved from the metadata repository, through a string and data type matching algorithm, as explained below (in order to facilitate this option, the data modeler should consider using the same or similar names and data types for the dimensions' attributes and a high granularity level as derived from the operational database tables).

InfoObjects represent the smallest unit of information in the SAP BW environment [9] and they are the core building blocks in the informational model (all the other data warehousing-related objects are built with InfoObjects). They are used for modelling business entities, attributes, hierarchies, key figures, or key performance indicators, as well as time, unit and currency information 10], and to create structures and tables in order to enable information modelling in a structured form within the data warehouse environment. The multidimensional generation component requires the retrieval of existing/previously generated InfoObjects in the warehousing repository, and their use in the modelling of the InfoCube. The process is accomplished through a similarity ranking algorithm for approximating string matching [11], validated by the data type and length conditions. Once the InfoObjects have been identified and retrieved, the prototype generates the InfoCube's physical implementation.

The final step of the prototype's execution is represented by the data load and storage component's implementation. Loading processes are generated based on the logical mappings (i.e. transformations) between the InfoObjects of the data structures which represent the source of data and the target InfoCube.

5. Related Work

The automation issues in the data warehousing environment have been previously addressed in the industry and academic world, most of them referring to ETL processes design, such as data extraction and population [12], data cleaning and integration [13], [14] etc. Some research works [15], [16] propose an automation of the conceptual or logical data warehouse design. In [15] the authors propose two algorithms: one for automating DW conceptual schema derivation from OLTP schemas and another one for the evaluation of candidate conceptual schemas with user queries; in [16] the authors introduce a rule-based mechanism, which automatically generates the relational data warehouse schema by applying existing design knowledge (a conceptual schema, non-functional requirements and mappings between the conceptual schema and the source database); in [17] the an automatic generation tool for conceptual design model implementation from conceptual graphical models, which comes closest to our idea. Automation of logical and physical mappings aspects in the data warehousing environment, as treated in [18], [19], [20] have also been considered in our approach.

However, to the best of our knowledge, automating the creation of data warehouse-relevant data structures from technical metadata, with the aim of defining a comprehensive enterprise wide solution has not yet been covered. We argue that given the existence of external metadata files provided by data modellers, our proposed prototype shortens considerably the manual repetitive work of developers and enables the implementation of an initial enterprise data warehouse model.

5 Conclusion

The framework architecture for semi-automated implementation of multidimensional models and the corresponding prototype, presented in this paper are designed for providing faster development of data warehousing projects, and reducing related costs and delivery times. Our main contributions consist in the design of a framework that defines several automation components and the boundaries of the systems for its implementation. The main goal achieved by the corresponding prototype implementation is the interpretation of UML class diagrams exported in XML format and the semi-automated generation of the multidimensional model in the data warehousing environment. The proposed prototype implements the physical data model from existing and generated technical metadata and logical mappings of the underlying metadata repository.

Several enhancements are to be made in the development of the prototype; therefore, we continue our research in order to improve the functionalities, enable integration of additional data sources types, and optimize the execution processes.

6. Acknowledgment

Ph.D. scholarship, Project co-financed by the SECTORAL OPERATIONAL PROGRAM FOR HUMAN RESOURCES DEVELOPMENT 2007 – 2013;

Priority Axis 1. "Education and training in support for growth and development of a knowledge based society"; Key area of intervention 1.5: Doctoral and post-doctoral programs in support of research. Contract nr: POSDRU/88/1.5/S/60185 – “Innovative doctoral studies in a Knowledge Based Society” Babeş-Bolyai University, Cluj-Napoca, Romania.

References

[1] J.A. Rodero, J.A. Toval, and M.G. Piattini, "The audit of the Data Warehouse Framework*," in *Proceedings of the International Workshop on Design and Management of Data Warehouses*, Heidelberg, Germany, 1999, pp. 14-1:14-12.

[2] D. Theodoratos, S. Ligoudistianos, and T. Sellis, "View selection for designing the global data warehouse," *Data Knowledge Engineering*, vol. 39, no. 3, pp. 219 - 240, December 2001.

[3] P. Lane, *Oracle Database Data Warehousing Guide, 10g Release 2 (10.2)*. Redwood City, CA: Oracle, Inc., 2005.

[4] E. Rahm and H.H. Do, "Data Cleaning: Problems and Current Approaches," *IEEE Data Engineering Bulletin*, vol. 23, no. 4, pp. 3 - 13, 2000, http://sites.computer.org/debull/A00DEC-CD.pdf.

[5] P. Ponniah, *Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals*. New-York: John Wiley & Sons, Inc., 2001.

[6] T. Friedman, M. A. Beyer, and E. Thoo, "Magic Quadrant for Data Integration Tools," Research Report ID Number: G00207435, 2010.

[7] M.H. Bracket, *The Data Warehouse Challenge: Taming Data Chaos*, 1st ed.: John Wiley & Sons, 1996.

[8] M. Muntean, "Implementation of the Multidimensional Modeling Concepts into Object-Relational Databases," *Revista Informatica Economică*, vol. 3, no. 43, 2007.

[9] SAP AG. (2009) Sap Library Document. [Online]. HYPERLINK "http://help.sap.com/SAPhelp_nw04/helpdata/en/" http://help.sap.com/SAPhelp_nw04/helpdata/en/

[10] K. McDonald, W.H. Inmon, A. Wilmsmeier, and D.C. Dixon, *Mastering the SAP Business Information Warehouse*. Indianapolis, Indiana: Wiley Publishing, Inc., 2002.

[11] S. White. (2004, February) Catalysoft. [Online]. HYPERLINK "http://www.catalysoft.com/articles/StrikeAMatch.html" http://www.catalysoft.com/articles/StrikeAMatch.html

[12] J. Adzic, V. Fiore, and S. Spelta, "Data Warehouse Population Platform," in *Proceedings of the VLDB 2001 International Workshop on Databases in Telecommunications II* , London, UK, 2001.

[13] M. Jarke et al., "Improving OLTP data quality using data warehouse mechanisms," in *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, Philadelphia, PA, 1999.

[14] V. Tziovara, P. Vassiliadis, and A. Simitsis, "Deciding the Physical Implementation of ETL Workflows," in *Proceedings of the ACM tenth international workshop on Data warehousing and OLAP*, New York, USA, 2007.

[15] C. Phipps and K. Davis, "Automating Data Warehouse," in *Proceedings of the 4th International Workshop on Design and Management of Data Warehouses*, London, UK, 2002, pp. 23-32.

[16] V. Peralta, A. Illarze, and R. Ruggia,

"Towards the Automation of Data Warehouse Logical Design: a Rule-Based Approach," in *Proceedings of CAiSE Short Paper Proceedings*, 2003.

[17] K. Hahn, C. Sapia, and M. Blaschka, "Automatically Generating OLAP Schemata from Conceptual Graphical Models," in *Proceedings of the ACM DOLAP 2000*, 2000, pp. 9-16.

[18] S. Benkley, J. Fandozzi, E. Housman, and G. Woodhouse, "Data element tool-based analysis (DELTA)," MITRE Technical Report 1995.

[19] M. Castellanos, A. Simitsis, K. Wilkinson, and U. Dayal, "Automating the loading of business process data warehouses," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology* , 2009, pp. 612-623.

[20] E. Rahm and P.A. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal — The International Journal on Very Large Data Bases*, vol. 10, no. 4, 2001.

Ilona-Mariana NAGY is currently a PhD Candidate in Cybernetics and Statistics at the Faculty of Economics and Business Administration of the Babes-Bolyai University. She has a Bachelor's degree in Business Informatics and a Master's degree in Business Informatics and the Informational Society. Her main scientific fields of interest include: databases, Data Warehousing, Business Intelligence and SAP technologies.