Meeting the Need for ETL Documentation: A Model-driven Framework for Customizable Documentation Generation

Frieder Jacobi
Department of Economics and Business Administration, Chemnitz University of Technology, Chemnitz, Saxony, Germany., frieder.jacobi@wirtschaft.tu-chemnitz.de

Robert Krawatzeck
Department of Economics and Business Administration, Chemnitz University of Technology, Chemnitz, Saxony, Germany., robert.krawatzeck@wirtschaft.tu-chemnitz.de

Marcus Hofmann
Department of Economics and Business Administration, Chemnitz University of Technology, Chemnitz, Saxony, Germany., marcus.hofmann@wirtschaft.tu-chemnitz.de

Meeting the Need for ETL Documentation: A Model-driven Framework for Customizable Documentation Generation

Frieder Jacobi
Chemnitz University of Technology
frieder.jacobi@wirtschaft.tu-chemnitz.de

Robert Krawatzeck
Chemnitz University of Technology
robert.krawatzeck@wirtschaft.tu-chemnitz.de

Marcus Hofmann
Chemnitz University of Technology
marcus.hofmann@wirtschaft.tu-chemnitz.de

ABSTRACT

Within Business Intelligence systems (BI systems), ETL (extract, transform and load) processes move numerous data from heterogeneous sources to a data warehouse and become more complex with growing enterprise size. To keep costs and expenditure of time for maintenance and evolution of those systems slight, ETL processes should be documented. A well-documented system also leads to higher transparency regarding the origin and processing of data, which increases the system's acceptance by business users. However, the preparation of high-quality software documentation is sophisticated and therefore it usually only takes place in the design or development phase of BI systems. To ensure that the documentation is always updated, an automatic generation is advantageous. The paper at hand presents a conceptual framework for automated configurable ETL documentation generation. The presented framework creates benefits for BI systems developers as well as business users.

Keywords

ETL documentation, documentation framework, automatically generated documentation, user-specific documentation, data warehouse system, business intelligence, knowledge management, Model Driven Architecture.

THE NEED FOR ETL DOCUMENTATION

In the field of Business Intelligence (BI), extract, transform, load (ETL) refers to three separate functions combined in a single software component. First, the extract function reads data from a specified source system and extracts a desired subset of data. Next, the transform function works with the acquired data – using rules or lookup tables, or creating combinations with other data – to convert them into the desired state. Finally, the load function is used to write the resulting data to a target system, which may or may not previously exist (Vercellis, 2009).

ETL tools are pieces of software which support users in the modeling and execution of ETL processes (Vercellis, 2009). Such tools are usually present in businesses maintaining an own data warehouse. With an increasing enterprise size, an increasing number of operational systems must be integrated and advanced reports must be created to meet the business user's requirements. Thus, ETL processes grow larger and become more complex. At the same time, end users want to be informed about how facts are determined, so they feel certain that they base their decisions on correct information. With growing complexity of ETL processes, providing this information becomes more difficult. This is enhanced by the fact that most end users do only need a part of the data embedded in ETL process descriptions. To meet the needs of different user roles and organizational units, the automatic process for creating documentation has to be configurable.

The general task of documentation is to describe how informational systems operate from a technical perspective as well as from an end user's perspective (Laudon and Laudon, 2010). As for other software products, the benefit of BI systems is decreasing without sufficient and high-quality documentation, because under certain circumstances it is hard or not possible at all to determine if the software meets the requirements (Wallmüller, 2001).

The requirements on BI systems are predetermined by a complex corporate environment and therefore underlie changes over time. This leads to a permanent process of adaption, which reaches from the data sources to the central data warehouse to the data marts through to the reports. As a result, the documentation of ETL processes which has been defined in the design and concept phase respectively in the development process becomes obsolete over time and is finally no longer usable. An

obsolete software documentation – which is not updated after software changes – is less valuable (Forward and Lethbridge, 2002; Laudon et al., 2010).

An empirical study conducted in 2011 (Gluchowski, Hofmann, Jacobi, Krawatzeck and Müller, 2011; Hofmann, Müller, Jacobi and Krawatzeck, 2012) shows that the automatic documentation of ETL processes is not properly supported by methods and tools. The results show further that in practice ETL processes are the least documented architectural components in modern BI systems. In the area of scientific research, deficits are recognizable, either. There are few relevant scientific works to find which follow up with the problem of (automated) documentation of BI systems in general and ETL processes in special. On the other hand, there is the fact of ETL processes having a high percentage in the development expenses of BI systems because of their complexity and the quantity of involved heterogeneous systems (Inmon, 1997).

The means for automated documentation of ETL processes differ between the various ETL modeling tools. In the area of commercial and open source solutions they can be differentiated into three classes: (1) tools with a built-in documentation component (e.g. IBM DataStage and Talend Open Studio), (2) tools without an integrated component but with currently available third-party applications that add documentation capabilities to the specific ETL tool (e.g. SSIS Documenter[1] and BI Documenter[2] for Microsoft SQL Server Integration Services (MS SSIS)), and (3) tools without any support for automated ETL documentation generation (e.g. Pentaho Data Integration[3]). The currently provided means for automated ETL documentation generation – regardless of being commercial or open source – create structured plain HTML files, including all available ETL process metadata, a graphical representation or a screenshot of the ETL processes (using the tool-specific graphical notation), and a navigation bar. Therefore, they reduce the manual effort of ETL documentation creation. However, in contrast to common formats usually used in business situations like Portable Document Format (PDF) or word processing formats (e.g. Office Open XML (DOCX)), the supported output formats are limited to plain HTML (except for the BI Documenter for MS SSIS, which also allows the creation of Compiled HTML Help files (CHM)). Furthermore, there are hardly any configuration capabilities provided, thus users have no choice except documenting the ETL process metadata as a whole. This is especially problematic since the documented information primarily addresses developers rather than business users. The biggest disadvantage of current ETL process documentation means is the fact that only explicit information within ETL process metadata gets documented. This ignores the additional value provided by making implicit information explicit by further metadata analysis – like a full lineage or impact analysis of specific table fields, including variables and control flow statements.

By means of a structured building of relations between data being embedded in ETL process definitions, information is allocated. This information may in turn be used by employees in different business units to generate knowledge in the organization. Thus a periodical, automated documentation of ETL processes in BI systems gives advantages from multiple perspectives. From the technical perspective the further development and maintenance of the system becomes easier, because the developers always have the latest version of the ETL process documentation. From the economic perspective this may save time and decrease development costs. Thus it becomes possible to store knowledge independently of individual persons. The benefits are shorter training periods for new employees and fewer loss of knowledge caused by fluctuation of employees. From the end user perspective the acceptance and the trust in the system may be increased by the higher transparency regarding the origin and processing of the data.

Therefore, the paper at hand presents a vendor-independent framework which satisfies the need for support for an automated, user-specific ETL process documentation and thereby closes the scientific gap in the field of automatic ETL documentation generation.

After this brief discussion of the problem and the motivation for an automated ETL documentation, requirements on high-quality ETL documentation are defined. Subsequently, we present a conceptual framework which solves the problem pointed out in a flexible way. Concluding, we summarize the findings and give an outlook to further research activities.

---

[1] http://www.ssisdocumenter.com/

[2] http://pragmaticworks.com/Products/Business-Intelligence/BIDocumenter/

[3] In the currently latest version 4.2.0, PDI provides a special *„Automatic Documentation Output"*-step, which creates documents (different file formats are supported) containing only screenshots of and marginal information (like creator name and creation data) about the ETL processes as a whole. Information about individual transformation steps is not provided. Thus, PDI does not fulfill the requirements for ETL process documentation.

The paper at hand is a contribution to Research-in-Progress. To ensure scientific rigor, it is adhered to the Design Science process by (Peffers, Tuunanen, Rothenberger and Chatterjee, 2007) in combination with the instructions for single scientific activities by (Hevner, March, Park and Ram, 2004).

REQUIREMENTS ON HIGH-QUALITY ETL DOCUMENTATION

Inside a company there are different user roles, which have different requirements on the structure and content of the documentation of ETL processes. The business user needs a view that enables him to verify the sources and the quality of the provided information. On the other side, a developer needs a more detailed and technical oriented documentation of the system. Different gradations of granularity, like system level (Online Transaction Processing Systems, OLTP), database level, table level and attribute level thereby increase the clarity. An additional categorization of the content according to further aspects, for example the organizational structure (marketing, sales, etc.) seems to be reasonable to provide a purposeful supply of information.

In addition to the ETL process description those two aspects should be considered at the creation of user-specific documentation. Figure 1 summarizes the presented facts and illustrates schematically their influence on the documentation to be created.

High-quality documentation is defined by eight characteristics (Wallmüller, 2001): changeability, consistency, completeness, comprehensibility, definiteness, identifiability, standard-compliance and up-to-dateness. Those characteristics should be taken into account for an automated ETL documentation creation framework.

The currently available solutions for automated ETL documentation for commercial and open source modeling tools do not fulfill the requirements pointed out above completely; especially the capability of creating user-specific documentation is hardly supported.

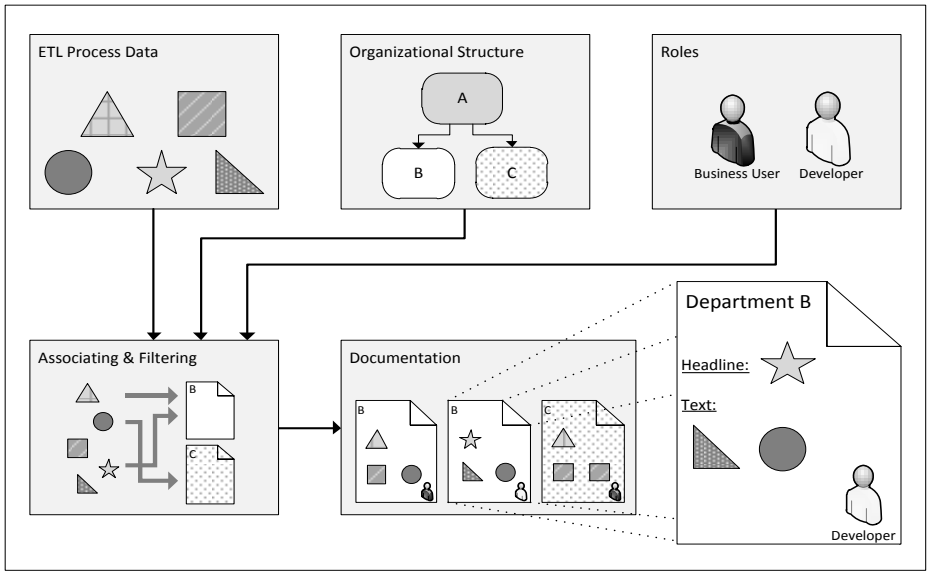

Figure 1. Creation of user-specific ETL documentation.

CONCEPTUAL FRAMEWORK FOR AUTOMATED ETL DOCUMENTATION GENERATION

In this section we present a conceptual framework for configurable ETL documentation creation based on the framework for user-specific IT documentation (Krawatzeck, Jacobi, Müller and Hofmann, 2011). Additionally, the adapted framework fulfills the eight requirements on high-quality documentation (Krawatzeck et al., 2011) and offers a high degree of flexibility – especially concerning the integration of multiple platforms – by using the advantages of the Model Driven Architecture (MDA) (Czarnecki and Helsen, 2006; Miller and Mukerji, 2003).

MDA is an approach to model-driven software development that has been standardized by the Object Management Group (Miller et al., 2003). MDA essentially is based on a strict separation of the system specification and implementation, which

leads to a high degree of reusability. The implementation is realized by models at different levels of abstraction – Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM) and code – and automated model transformations.

An overview of the framework is shown in figure 2. In the following, the components are described in detail, starting with ETL tools which are used to define ETL processes (represented in the bottom left corner of the figure), following the data flow via the model representation of ETL processes and the transformation into a document model through to the generation of concrete documenting artifacts.

ETL Process Modeling Tools

As mentioned above, ETL tools are pieces of software which support users in the modeling and execution of ETL processes (Vercellis, 2009).

ETL Tool Meta-models

ETL tools usually provide a graphical user interface which allows a graphical representation of the modeled ETL processes. Additionally, modeled processes may be saved and loaded. In most cases, the output format does not conform to an open standard – like the Common Warehouse Model (CWM) (Object Management Group, 2003) –; instead a proprietary one is used. For utilizing the data as input for a documentation framework, the format must be read and semantically understood. The problem is that proprietary formats are not designed for being understood and reused by other tools, thus an interface is necessary. The retrieved information subsequently may be stored in platform-specific models which conform to a tool's specific meta-model formalizing the ETL tool's specific modeling capabilities.

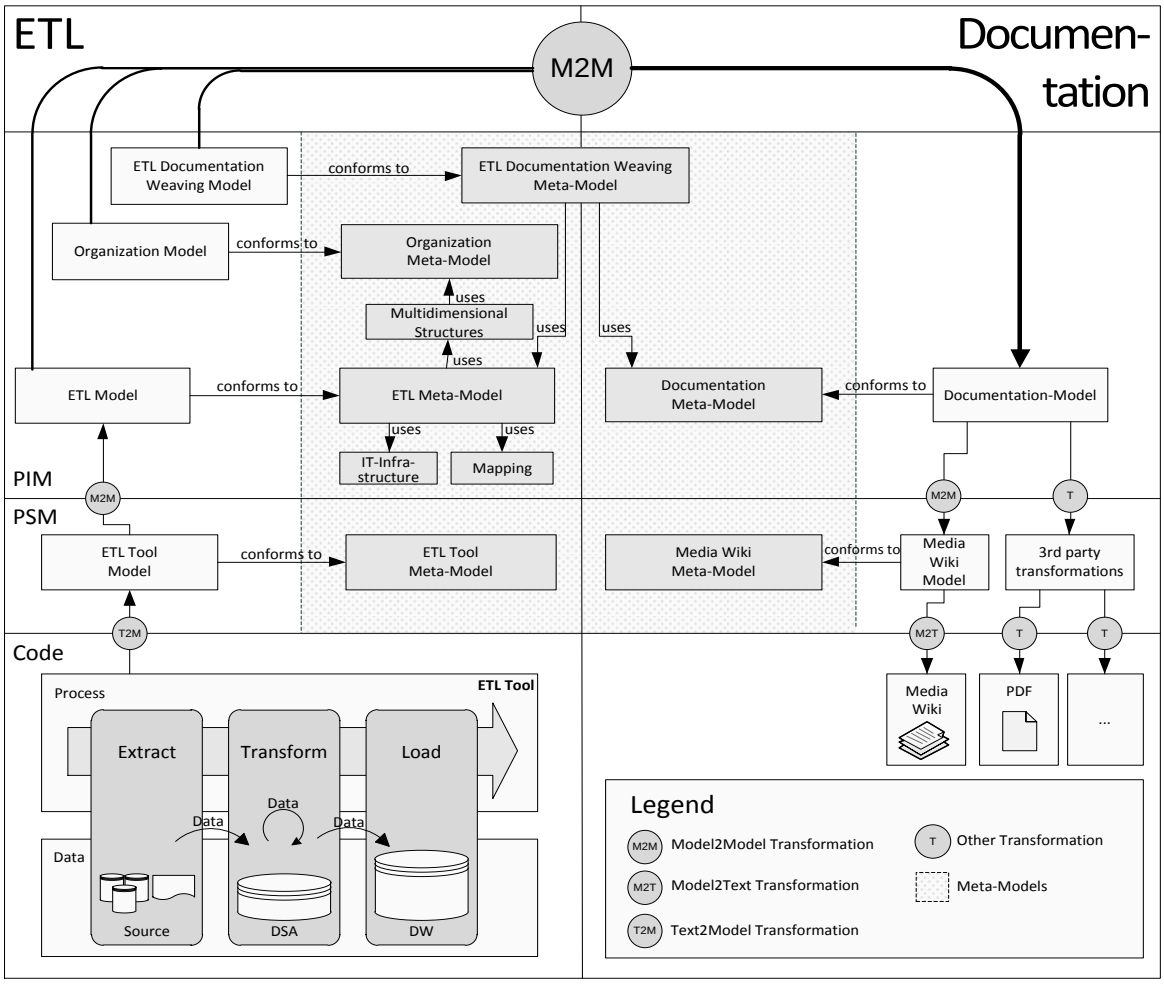

Figure 2. Model-driven framework for ETL documentation generation.

Platform-Independent ETL Meta-model

Through the existence of multiple ETL tool providers, several platform-specific meta-models must be developed, each being specific for the tool it covers. Those meta-models are normally not compatible to each other. Transforming those platform-specific models directly into documenting artifacts leads to a large number of similar transformations. This results in high maintenance efforts and error-proneness in the case of new or changed requirements for documentation.

To avoid this problem a platform-independent ETL meta-model is required, which describes the ETL processes in a conceptual way. Platform-specific ETL models can be transformed into a platform-independent model using model-to-model transformations (M2M). The platform-independent model forms the central point for further transformations. The implementation of such an ETL meta-model has been investigated, among others, by (Muñoz, Mazón, Pardillo and Trujillo, 2010) and (Vassiliadis, Simitsis and Skiadopoulos, 2002). Muñoz et al. describe a data warehouse modeling tool based on a model-driven approach, including a functional ETL meta-model. They suggest using the platform-independent models to document ETL processes, but an explanation of how to come to detailed documentation adjusted to the user's specific requirements is missing. Conceptual work on ETL processes has been undertaken by (Vassiliadis, Simitsis and Baikousi, 2009), who presented a promising taxonomy of ETL activities.

The MDA approach allows the definition of a number of additional meta-models covering several aspects of the domain of data warehousing (Kurze and Gluchowski, 2010), like multidimensional and relational data structures, IT infrastructure which describes the involved devices and data mapping which contains calculation and derivation rules. Those meta-models would cover a large range of elements needed for the specification of ETL processes, such as the involved devices and the calculation of derived measures. Concerning the realization of an ETL activity meta-model, the work of (Vassiliadis et al., 2009) proposes a promising starting point. Presumed this ETL activity meta-model has been implemented, the presented framework is capable to model ETL processes platform-independently.

Transformation from ETL to Documentation

The platform-independent ETL model contains the complete definition of the ETL processes in a conceptual form, but for the creation of documentation being valuable for specific end-users, further processing is needed. Depending on the users' department affiliations and professions, ETL process data has to be filtered and structured differently. The data needed for this purpose are not provided within the ETL process description, thus other data sources are required.

An organization model contains data about the organizational structure, including departments and their relations. In combination with the description of multidimensional structures which relates – amongst others – data marts and departments, the nonexclusive, noncovering assignment of ETL data to organizational structure is possible.

For structuring the ETL documentation dependent on the users' professions, the ETL-Documentation-Weaving meta-model has to be implemented. An ETL-Documentation-Weaving model maps specific ETL data items (e.g. data sources, activities and data targets like cubes) onto specific documentation concepts (e.g. headlines, sections and paragraphs). Structural needs of the different professions may be fulfilled by providing multiple mappings for different target user roles.

Given the data in the three models ETL, organization and ETL-Documentation-Weaving, the creation of valuable information through a structured combination will be possible (Laudon et al., 2010).

Documentation Meta-model

Since the aim of the presented framework is a flexible and hence configurable solution for ETL documentation generation, different kinds of target document formats must be supported. Those formats should be able to express the separation of content, for example with multiple documents like PDF, or categories like in wiki systems. To avoid the implementation of multiple transformations between the three source models and different output formats, a platform-independent documentation model for storing the ETL process description is necessary, according to the platform-independent ETL model.

There exist a number of different platform-independent formats for documentation, for example Darwin Information Typing Architecture (DITA) (DITA, 2011), DocBook (DocBook, 2011) and OpenDocument (OpenDocument, 2011). From those formats DITA is the most promising, since it has a topic-oriented approach in contrast to DocBook and OpenDocument, which are especially designed for the creation of hierarchical documents like books. The topic-oriented approach naturally implies the possibility of reusing pieces of information. Additionally, the DITA Open Toolkit (DITA-OT) (DITA-OT, 2011) supports a wide range of output formats out of the box, like PDF, RTF, XHTML, DocBook, OpenDocument Text (ODT) and Compiled HTML Help (CHM). Furthermore, DITA-OT may be extended by individual transformations for required output

formats. Since the concrete output formats requested by a certain user are yet unknown, it is advantageous to select a document format that supports the generation of a large number of different output formats out of the box.

MediaWiki: Meta-model and Transformation

In addition to the static document formats supported by DITA-OT and other documentation formats, an interactive documentation system is useful for capturing additional user-contributed annotations. Wiki systems, for example, are a technology that invites users to contribute their knowledge.

For the purpose of the reuse of the MediaWiki meta-model and the related model-to-text transformation (M2T), a transformation from the DITA document model into the MediaWiki model must be developed. In the case of reuse of the given MediaWiki-related components, user-contributed annotations of ETL documentation are supported as well.

CONCLUSION AND FURTHER WORK

The previous remarks introduce a framework that aims at closing the scientific gap in the field of automatic ETL documentation generation considering the eight attributes of high-quality documentation. A practical application of the system promises a significant reduction of manual efforts in this task. Furthermore, the resulting documentation is more up to date, since the automation allows a simpler integration of a continuous documentation process into the maintenance processes of data warehouses. The reduction of manual interventions helps to avoid media disruptions and thus sources of error, significantly increasing the documentation quality.

The application of an independent documentation meta-model allows supporting a wide range of output formats. Thereby it is possible to align the document structures with role-dependent and department-specific information requirements. ETL as a filling process affects the data stored in a data warehouse in a high degree. Due to this relation the generated ETL documentation raises the significance of the storage layer documentation, especially concerning data quality and data origin.

Following the research process proposed by (Peffers et al., 2007), the practicability of the presented framework has to be demonstrated. A demonstration by means of a software prototype is highly promising both from a scientific and practical viewpoint. It offers a platform to scientists, allowing them to apply modern methods of the model-driven software engineering to the specialties of ETL processes. Practitioners get assistance in accomplishing complex issues in the process of development, maintenance and documentation of ETL processes.

ACKNOWLEDGEMENTS

This research has been partly funded by the European Social Fund and the Federal State of Saxony, Germany.

REFERENCES

1. Czarnecki, K. and Helsen, S. (2006). Feature-based survey of model transformation approaches. IBM Systems Journal, 45(3), 621 -645.
2. DITA. (2011). Retrieved from http://www.oasis-open.org/committees/dita
3. DITA-OT. (2011). Retrieved from http://dita-ot.sourceforge.net
4. DocBook. (2011). Retrieved from http://www.oasis-open.org/docbook
5. Forward, A. and Lethbridge, T. C. (2002). The Relevance of Software Documentation, Tools and Technologies: a Survey. Proceedings of the 2002 ACM symposium on Document engineering - DocEng'02 (pp. 26-33). New York, USA: ACM Press. doi:10.1145/585064.585065
6. Gluchowski, P., Hofmann, M., Jacobi, F., Krawatzeck, R. and Müller, A. (2011). Business-Intelligence-Umfrage 2011: Softwaregestütztes Lebenszyklusmanagement und aktuelles Dokumentationsgeschehen für Business-Intelligence-Systeme (pp. 1-31). Chemnitz, Deutschland. Retrieved from http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-75452
7. Hevner, A. R., March, S. T., Park, J. and Ram, S. (2004). Design Science in Information Systems Research. MIS Quarterly, 28(1), 75-105.
8. Hofmann, M., Müller, A., Jacobi, F. and Krawatzeck, R. (2012). Umfrage 2011: „Dokumentation von Business-Intelligence-Systemen" - Ergebnisse und Auswertung. In D. C. Mattfeld and S. Robra-Bissantz (Eds.), Tagungsband der Multikonferenz Wirtschaftsinformatik 2012 (MKWI'12) (pp. 1091-1104). Berlin, Germany: GITO Verlag.
9. Inmon, B. (1997). The data warehouse budget. DM Review Magazine.

10. Krawatzeck, R., Jacobi, F., Müller, A. and Hofmann, M. (2011). Konzeption eines Frameworks zur automatisierten Erstellung nutzerspezifischer IT-Systemdokumentationen. In H. Baars (Ed.), Workshop Business Intelligence 2011 (WSBI'11) der GI-Fachgruppe Business Intelligence, Business Intelligence - Impulse für die Forschung oder Impulse durch die Forschung (pp. 15-26). Stuttgart: CEUR Workshop Proceedings. Retrieved from http://ceur-ws.org/Vol-831/
11. Kurze, C. and Gluchowski, P. (2010). Computer-Aided Warehouse Engineering (CAWE): Leveraging MDA and ADM for the Development of Data Warehouses. AMCIS 2010 Proceedings. Paper 282.
12. Laudon, K. C. and Laudon, J. P. (2010). Management information systems: managing the digital firm (11th ed.). Upper Saddle River, New Jersey: Pearson.
13. Miller, J. and Mukerji, J. (2003). MDA Guide V1.0.1. Object Management Group. Retrieved from http://www.omg.org/cgi-bin/doc?omg/03-06-01
14. Muñoz, L., Mazón, J.-N., Pardillo, J. and Trujillo, J. (2010). Modelling ETL Processes of Data Warehouses with UML Activity Diagrams. On the Move to Meaningful Internet Systems: OTM 2008 Work-shops (pp. 44-53). Springer.
15. Object Management Group. (2003). Common Warehouse Metamodel (CWM) Specification, Version 1.1.
16. OpenDocument. (2011). Retrieved from http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office
17. Peffers, K., Tuunanen, T., Rothenberger, M. A. and Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45-77.
18. Vassiliadis, P., Simitsis, A. and Baikousi, E. (2009). A taxonomy of ETL activities. Proceeding of the ACM twelfth international workshop on Data warehousing and OLAP (pp. 25–32). Hong Kong, China: ACM. doi:10.1145/1651291.1651297
19. Vassiliadis, P., Simitsis, A. and Skiadopoulos, S. (2002). Conceptual modeling for ETL processes. Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP - DOLAP'02 (pp. 14-21). New York, New York, USA: ACM Press. doi:10.1145/583890.583893
20. Vercellis, C. (2009). Business intelligence: data mining and optimization for decision making. Hoboken, N.J.: Wiley.
21. Wallmüller, E. (2001). Die Rolle der Dokumentation in Software-Projekten. Software-Qualitätsmanagement in der Praxis: Software-Qualität durch Führung und Verbesserung von Software-Prozessen (2nd ed., pp. 149-156). Hanser Fachbuch.