AUTOMATED DATA MODEL EVALUATION

Zoltan Kazi, Ljubica Kazi, Biljana Radulovic

University of Novi Sad, Technical faculty „Mihajlo Pupin" Zrenjanin, Djure Djakovica bb, Serbia

e-mail: zoltan.kazi@gmail.com, ljubicakazi@ptt.rs, bradulov@ptt.rs

Abstract

Modeling process is essencial phase within information systems development and implemetation. This paper presents methods and techniques for analysis and evaluation of data model correctness. Recent methodologies and development results regarding automation of the process of model correctness analysis and relations with ontology tools has been presented.

Key words: *Database modeling, Data model correctness, Evaluation.*

1. INTRODUCTION

Data model correctness is one aspect of data quality, as a general concept. The relevance of data quality in both organizational and decisional processes is recognized by several international institutions and organizations, which resulted in significant number of contributions to the research by database and information system communities.

There are many data quality tools developed in the research and commercial purpose. Many data quality software tools are advertised and used in various data-driven applications to improve the quality of data models and business processes.

Data modeling has different aspects of quality regarding various data model types, as well as issues regarding process of data model creation, evaluation and correction. Conceptual modeling is considered most difficult and error-prone, especially for novice designers. Therefore, many efforts are made to create rules and heuristics for error detection and consultative support to modeling process. Still, CASE tools are to be improved for support to error detection in semantic domain.

This paper presents a review of methods and technologies that are used for data model correctness analysis and evaluation. It also shows recent research and development efforts and results, including those made by authors, in automating the process of data model evaluation.

2. THEORETICAL BACKGROUND

Data models are usually created in the process of information systems development by using CASE tools that integrate business process modeling results to data modeling.

Data models are specific theoretically based specifications that are used for creation of real databases of information systems (Elmasri & Navathe, 2007). Data model is a formal abstraction through which the real world is mapped in the database (Ullman, Garcia – Molina & Widom, 2002). Data model enables representation of a real world system through a set of data entities and their connections. They can be represented in various ways:

- Diagram (schema) – graphical representation, using specific set of symbols with methodology based meanings,

- Data dictionary representation – where elements of data model are listed and textually described in non-structural or semi/structural way,
- Formal languages representation, such as predicate logic calculus.

One of the fundamental principles of the database approach is that a database allows unified representation of all data managed in an organization. This is achieved only when methodologies are available to support integration across organizational and application boundaries.

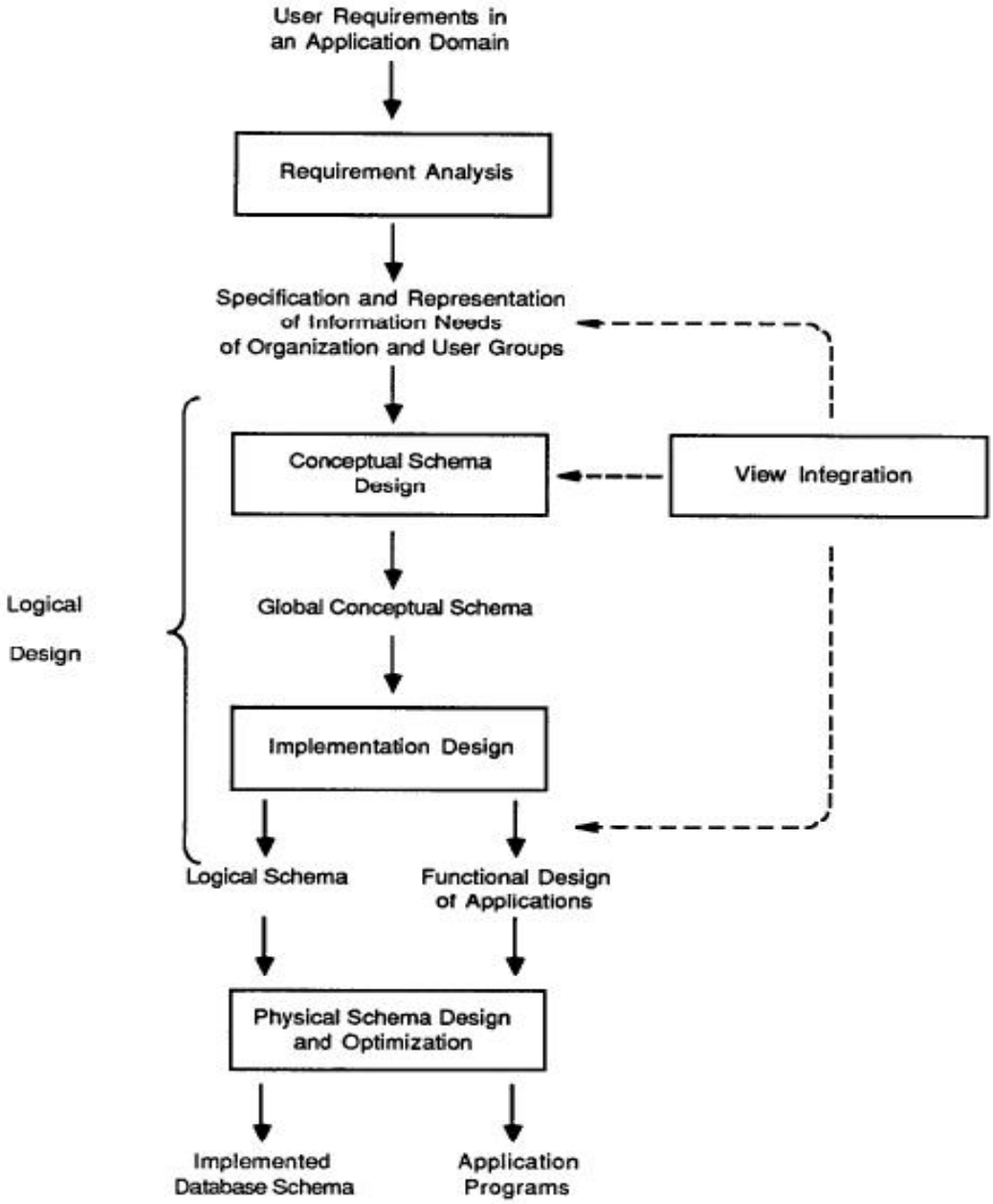


Fig. 1. Phases of database design. (Navathe et al., 1978)

According to (Emer et al., 2008) formal presentation of a data model could be represented as tuple S = (E, A, R, P), where:

- E is a finite set of entities,
- A is a finite set of attributes,
- R is a finite set of constraints concerning domain, definition, relationship and semantics associated to the elements and attributes,
- P is a finite set of association rules among elements, attributes and constraints.

The formal presentation of a conceptual data model (Kazi et al., 2010) is based on formal presentation shown in (Emer et al., 2008), but extended with one new element suitable for EER data model and it is represented as tuple S = (E, A, R, C, P) where:

- E is a finite set of entities,
- A is a finite set of attributes,
- R is a finite set of relationships,
- C is a finite set of constraints concerning domain, definition, relationship cardinality, mandatory attributes and semantics associated to the elements and attributes,
- P is a finite set of association rules among entities, attributes, relationships and constraints.

Methodologies for database design usually perform the design activity by separately producing several schemas, representing parts of the application, which are subsequently merged. Database schema integration is the activity of integrating the schemas of existing or proposed databases into a global, unified schema, The aim of research (Batini, Lenzerini & Navathe, 1986) is to provide first a unifying framework for the problem of schema integration.

3. RELATED WORK

3.1. Evaluation of data models

In paper (Yücesan et al., 2002) authors outline practical techniques and guidelines for verifying and validating of models generally. They provide examples of a number of typical situations where model developers may make inappropriate or inaccurate assumptions, and offer guidelines and techniques for carrying out verification and validation of models, in order to help avoiding major and serious modeling errors.

Quality of data model reflects quality of data in databases and business intelligence applications. Regarding data model quality, there are several approaches that deal with this problem during the process of data model creation, data model evaluation and data model correction. Data models that are included in quality evaluations include structured (ER, Relational, Object-Oriented) and semi-structured (XML) data models, described at conceptual, logical or physical level (Batini, Lenzerini & Navathe, 1986).

In the field of evaluation issues in the process of data model creation, some experimental research has been conducted regarding comparison of ER and object oriented modeling (Shoval, 1997, De Lucia et al., 2009), where it has been shown the significance of ER modeling and its value comparing to object-oriented approach. It has been shown that ER modeling has some advantages to object-oriented approach.

Data models could be evaluated from syntax and semantic aspect. The main difference between ontology and a database schema is that database schema is usually limited to describing a small subset of a mini-world from reality in order to store and manage data. An ontology attempts to describe part of reality or a domain of interest as completely as possible. (Formica et al., 2006) Ontology, that more completely (Ullman et al., 2002) describes the knowledge for the specific problem domain, could be used for comparing with the designed data model, so data model will be evaluated from semantic aspect (Kazi et al 2010).

3.2. Conceptual modeling errors

After creating ER data models, CASE tools enable automatic transformation to relational data models, following well-defined rules. But, while transforming to relational model, they are not able to capture errors that lead to normalization problems (Bock, 1997), so it is necessary to take care of certain types of ER modeling errors, in aim to avoid future normal forms errors in relational data model.

Batra & Antony (2001) present conceptual modeling errors as human errors at three performance levels: skill-based, rule based and knowledge based. Special software prototype CODA was implemented for the purpose of consulting support to conceptual database design to novice designers, and it has been shown (during experimental survey with students) that using this software enhance quality of data models. This software system includes heuristics and rules for recognition of typical errors in ER modeling during the process of data model creation, as well as support to further assistance toward solving these issues.

Skill-based errors in data modelling could be minimized by appropriate education process or training. Two main conceptual modeling training approaches have been compared in research (Batra et al., 2004): rule-based and pattern-based approach. It has been experimentally shown that more complex tasks influence lower designer performance. It has also been shown that rule-based approach is not significantly superior to pattern-based approach generally, but rule based-approach for novice designers given the significantly better performance in two of three complexity levels.

3.3. Conceptual and relational model quality metrics

One way to enhance the capability of an information system is to consider its conceptual model quality as well as its functional behaviour. Conceptual model quality can be defined as a set of perceivable characteristics expressed with quantifiable parameters that may be objective and/or subjective. The aim of empirical investigation (Cherfi et al., 2007) is to evaluate and compare perceived and measured quality of different conceptual model versions of the same universe of discourse. This research describes:

a) a set of metrics (clarity, simplicity, expressiveness, minimality) applied to different versions of ER conceptual schemas,

b) a framework enabling a comprehensive comparison of the conceptual schemas,

c) an experimentation leading to the evaluation of the same schemas by information system stakeholders such as designers, end-users, and students,

d) a comparison of the objective and subjective evaluations based on a sample of about 120 observations using different statistical methods.

According to results authors are able to identify quality criteria relevant to different groups of stakeholders, depending on several dimensions, such as their professional experience, and/or their specialization degree.

Referential integrity is an essential global constraint in a relational database, that maintains it in a complete and consistent state. Ordonez & García (2008) assume that the database may violate referential integrity and relations may be denormalized. They propose a set of quality metrics, defined at four granularity levels: database, relation, attribute and value, that measure referential completeness and consistency. Quality metrics are efficiently computed with standard SQL queries, that incorporate two query optimizations: left outer joins on foreign keys and early foreign key grouping. Experiments evaluate their proposed metrics and SQL query optimizations on real and synthetic databases, showing they can help in detecting and explaining referential errors.

3.2. Automating of data model evaluation

Research in the field of automating data models evaluation resulted in development of software prototypes that evaluate certain types of data models.

Software prototype for conceptual data model (ER) model evaluation (Sugumaran & Storey, 2006) use domain ontology in creating conceptual ER data model, as well as evaluation of externally created ER data model.

According to the experimental research (Emer, Vergilio & Jino, 2008), testing of relational database schema has been done with ADIA (Alternative Data Instance Analyses) system. This is a fault-based testing approach. The goal is to reveal constraint faults for schema elements, incorrect or absent restriction definitions. The data model is represented by a metamodel using UML notation. ADIA includes: Database instance alternatives, Original database instance with a simple modification, Queries, SQL statements automatically generated. The data model is represented by a meta-model M using UML notation Classes: Element, Attribute and Constraint.

In study (Formica & Missikoff, 2006) the problem of object oriented database (OODB) design is analysed, focusing in particular on the correctness of OODB schemas with IS-A hierarchies. Software prototype use object-oriented language elements for defining the physical structure of database by using TQL.

In paper (Choppella et al., 2007) authors reported their experience with exploring the use of PVS to formally specify and apply automated reasoning with ER data models. Working with a text-book example, they rely on PVS's theory interpretation mechanism to verify the correctness of the mapping across various levels of abstraction. Entities and relationships are specified as user defined types, while constraints are expressed as axioms. They demonstrate how the correctness of the mapping from the abstract to a conceptual ER model and from the conceptual ER model to a schema model is formally established by using typechecking. The verification involves proving the type correctness conditions automatically generated by the PVS type checker. The proofs of most of the type correctness conditions are fairly small (four steps or less). This holds out promise for complete automatic formal verification of data models.

4. PROPOSED APPROACH

Authors propose system of integrated software tools (Fig.2.) that could automate the process of data model evaluation and therefore enhance quality of data models. The basis of integration (Kazi et al. 2010) relies on using XML as result form that some software tools produce.

This system is to be used in the process with following steps:

- Creating axioms that describe general reasoning rules regarding specific type of data model (EER, RM, OOM etc.). Axioms are reasoning rules that are applied from data model

definitions and they are generally applicable to any data model that is to be evaluated. Axioms are used as general theoretical foundation of checking syntax part of quality of data model.

- Formalization of data models, by using first order predicate logic calculus as a formal language.
- Creating a system of axioms that will be merged with formally presented data models within the transformation tool, to be processed by an automated reasoning system. They are set of rules are to be applied over data model elements so the main conclusion could be made about data model quality. Axioms are used as reasoning rules that define the elements of correctness of a data model.
- Transformation of first order predicate calculus form of data model to a form that is suitable for processing in the selected system of automatic reasoning, i.e. to clauses in PROLOG.
- Merging clauses regarding data model and reasoning rules to an input file for PROLOG.
- Entering input file to automated reasoning system, setting queries and getting results.

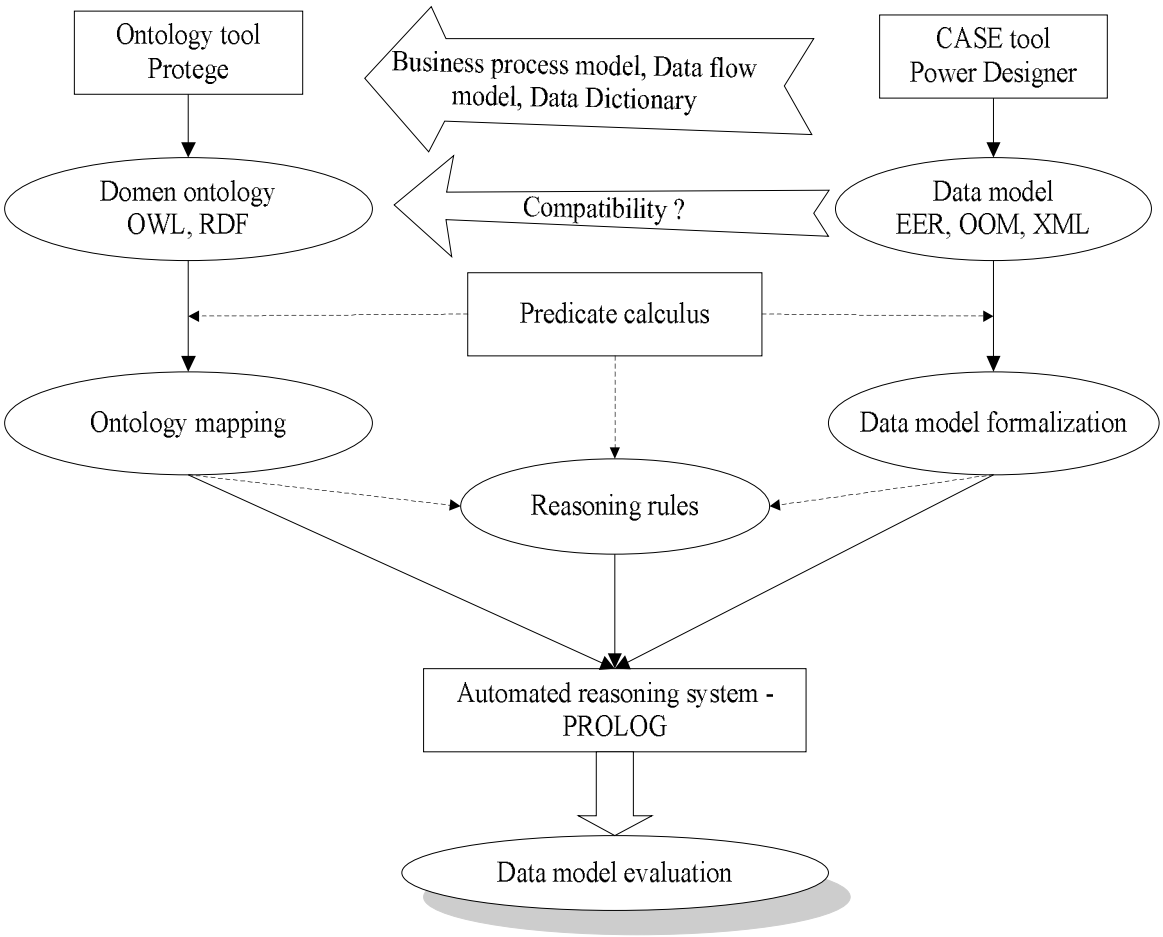


Fig. 2. Process of data model semantic evaluation (Kazi et al., 2010)

5. RESULTS

Software tools, that present parts of the proposed system (Kazi et al., 2010) are:

- CASE tool for data modeling, which saves result of modeling in output form of files that in fact XML files that consist of parts of data model and its descriptions.
- PROLOG - as an automated reasoning tool. Input data to PROLOG is file with extension PRO, which is in fact a specially formatted textual file

They have been empirically tested with data models that consist of several entities (Kazi et al, 2010).

For the purpose of automated support to data model correctness analysis, authors developed software tool (Fig.3.) that enable importing model from CASE tool and evaluation of model according to previously defined rules (Kazi & Radulovic 2011).

XML file that represent model from CASE tool is converted to set of clauses and merged with evaluation rules so they all are unified as an input to PROLOG. Within PROLOG, all input is processed according to questions regarding certain aspects of data model quality.

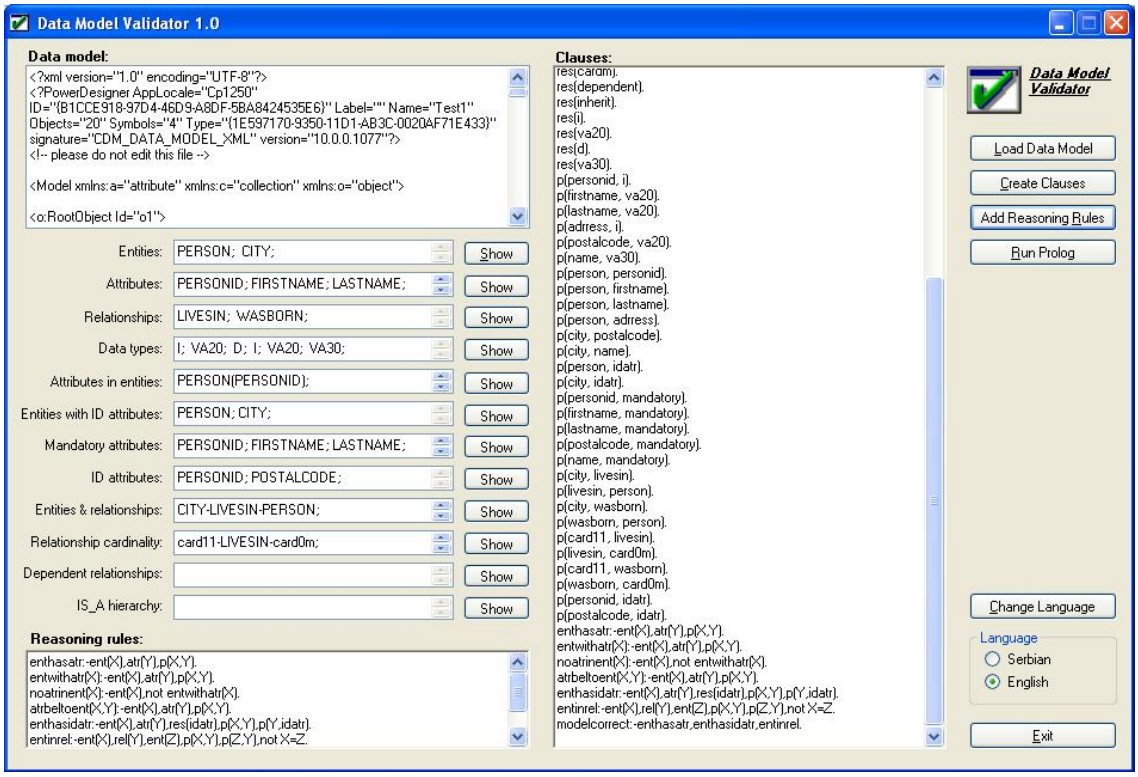


Fig. 3. Software tool prototype for data model correctness analysis (Kazi and Radulovic, 2011)

After loading data model, each element of model has been presented by clauses. Reasoning rules are also presented as clauses. They all are merged and after completion of the list of clauses, PROLOG could be started. PROLOG gives answers to particular questions regarding certain element of the model.

6. CONCLUSION

This paper presents theoretical survey and research results in the field of data model evaluation and analyses. There are several approaches that deal with this problem during the process of data model creation, data model evaluation and data model correction. Research in the field of automating data models evaluation resulted in last few years with development of software prototypes that evaluate certain types of data models.

Research and development results of authors in the field of automation of data modeling evaluation is presented as concept and software prototype. Further research is directed towards integrating with ontology tools that enable using ontologies for different knowledge areas that describes semantic aspect of business domain. Main research goal is to find a correlation and mapping between ontology and a formal logic language, i.e. transformation of ontology to a form that is appropriate for automated reasoning system input. This way ontologies could be used in conceptual model evaluation.

REFERENCES

Batini, C. & Scannapieco, M. (2006) "Data Quality", Springer.

Batini, C., Lenzerini, M. & Navathe S.B. (1986) "A Comparative Analysis of Methodologies for Database Schema Integration", ACM Computing Surveys, Vol. 18, No. 4, ACM 0360-0300/86/1200-0323.

Batra, D. & Antony, S.R. (2001) "Consulting support during conceptual database design in the presence of redundancy in requirements specifications: an empirical study", International Journal of Human-Computer Studies, Elsevier.

Batra, D. & Wishart, N.A. (2004) "Comparing a rule-based approach with a pattern –based approach at different levels of complexity of conceptual modelling tasks", International Journal of Human-Computer Studies, Elsevier.

Bock, D.B. (1997) "Entity-Relationship Modelling and Normalization Errors", Journal of Database Management.

Cherfi, S.S., Akoka, J. & Comyn-Wattiau, I. (2007) "Perceived vs. Measured Quality of Conceptual Schemas: An Experimental Comparison", Twenty-Sixth International Conference on Conceptual Modeling - ER 2007.

Choppella, V., Sengupta, A., Robertson, E.L. & Johnson, S.D. (2007) Preliminary Explorations in Specifying and Validating Entity-Relationship Models in PVS, AFM'07, November 6, Atlanta, GA, USA, ACM ISBN 978-1-59593-879-4/07/11.

De Lucia, A. , Gravino, C., Oliveto, R. & Tortora, G. (2009) "An experimental comparison of ER and UML class diagrams for data modelling", Journal of Empirical Software Engineering, ISSN: 1382-3256, Springer Science+Business Media, LLC.

Elmasri, R. & Navathe, S.B. (2007) "Fundamentals of Database Systems", Addison Wesley.

Emer, M.C., Vergilio, S.R. & Jino, M. (2008) "Testing Relational Database Schemas with Alternative Instance Analysis", 20th International Conference on Software Engineering & Knowledge Engineering (SEKE'2008), San Francisco, USA.

Formica, A. & Missikoff, M. (2006) "Correctness of ISA hierarchies in Object-Oriented database schemas", Journal of Advances in Database Technology, Springer Berlin/Heidelberg.

Kazi, Lj., Kazi, Z., Radulovic, B. & Letic., D. (2010) "Using Automated Reasoning System for Data Model Evaluation", 8th International Symposium on Intelligent Systems and Informatics SISY 10-11. September, Subotica, Serbia.

Kazi, Z. & Radulovic, B. (2011) "Software tool for Automated Analysis of Conceptual Data Model", 34th International conference MIPRO, Opatija, Croatia.

Navathe, S.B. & Schkolnick, M. (1978) "View representation in logical database design", In Proceedings of the International Conference on Management of Data (Austin, Tex.). ACM, New York, pp. 144-156.

Ordonez, C. & García, J. (2008) "Referential integrity quality metrics", Decision Support Systems 44, pg. 495–508.

Shoval, P. (1997) "Experimental Comparisons of Entity-Relationship and Object-Oriented Data Models", Journal AJIS, Vol. 4, No. 2.

Sugumaran, V. & Storey, V.C. (2006) "The Role of Domain Ontologies in Database Design: An Ontology Management and Conceptual Design Environment", ACM Transactions on Database Systems, Vol. 31, No. 3.

Ullman, J., Garcia – Molina, H. & Widom, J. (2002) "Database Systems: The Complete Book", Department of Computer Science, Stanford University, Prentice Hall, New Jersey, USA.

Yücesan, E., Chen, C.-H., Snowdon, J.L., Charnes, J.M. & Carson, J.S.II (2002) "Model Verification and Validation", Proceedings of the 2002 Winter Simulation Conference.