Model-driven Techniques for Data Model Synthesis

Drazen Brdjanin and Slavko Maric

***Abstract*—This article presents a survey of model-driven techniques for data model synthesis. During an extensive research, we identified more than 70 research papers in the field and more than 15 different graphical notations used for the source model representation. We have classified the proposed approaches into four distinct groups: function-oriented, process-oriented, communication-oriented and goal-oriented. Their contributions are presented in chronological order and evaluated based on several main criteria. Although the idea of model-driven design of the data model is more than 25 years old, the survey shows the richness and diversity of ideas, but only a small number of implemented automatic generators.**

***Index Terms*—Communication-oriented, data model, function-oriented, goal-oriented, model-driven, process-oriented, survey.**

Review Paper
DOI: 10.7251/ELS1317130B

I. Introduction

THE data model constitutes one of the most important artifacts in the information system design process, as well as the crucial component of software system models. Consequently, the automatization of data model design has been the subject of research for many years.

Since Chen's eleven heuristic rules [1] for the translation of information requirements specified in a natural language into an E-R diagram, a lot of research has been done in the field of natural language processing (NLP) on extracting knowledge from requirements specifications and automating data model design. At present, a natural language is the most frequently used way for requirements specifications and the majority of approaches to automated data model design are NLP-based approaches. Their effectiveness and limitations are usually deeply related to the source language, and their utilization is questionable for languages with complex morphology. Currently, there are several non-NLP-based alternatives to automated data model design, such as approaches taking models (graphically specified business/software requirements) as the basis for automated data model design instead of requirements specifications expressed in a natural language.

The first papers that focused on the model-driven design of the data model appeared in the second half of the 1980s. The first papers reporting the model-driven tools for (semi)automatic data model synthesis were published in the mid-1990s. However, although the idea of model-driven design of the data model is more than 25 years old, the fully automatic model-driven synthesis of the data model is still the subject of intensive research. In the existing literature there are only a small number of papers presenting the implementation of the automatic model-driven generator of the target data model with the corresponding evaluation results, while the great majority of papers only present modest achievements in (semi)automated, or even manual, data model synthesis.

In the related literature there are no papers presenting the systematic literature review (SLR) in the field of model-driven synthesis of the data model. Several existing papers [2]–[5] give only a partial overview of the field by focusing on some particular source notation or by presenting a wider overview of techniques for synthesizing different (not only data) models and taking different source (not only graphical) specifications as the base. Jilani et al. in [3] present a comparative study of seven approaches to the synthesis of UML diagrams based on data flow diagrams (DFD), but this study is not focused on data model synthesis and does not cover all existing approaches taking the DFD as a starting point. Franch et al. in [2] present the classification of the combined usage of the i^* notation with other notations and modeling frameworks. However, only two of all identified papers address the synthesis of data models based on i^*. Loniewski et al. in [4] present the SLR of the use of requirements engineering techniques in model-driven software development. However, they are not exclusively focused on the model-driven synthesis of the data model, and only a few papers, out of approximately 70 identified, belong to the target group of model-driven approaches to data model synthesis. Yue et al. in [5] present the SLR of transformation approaches between user requirements and analysis models, but this review is focused on the textual specification of user requirements as a starting point.

Inspired by the lack of an appropriate review of the existing model-driven techniques for data model synthesis, we have conducted an extensive survey of the related literature and identified more than 70 papers addressing the model-driven synthesis of the data model. In this paper we present the results of this survey and provide a classification of the identified techniques based on the primary focus, i.e. the orientation of a source notation. The contribution of the related papers within each classification group is presented in chronological order and evaluated based on several main criteria.

The article is structured as follows: after the introduction, the second section presents the taxonomy of source notations and the corresponding classification of the existing papers; the subsequent sections briefly present all categories and their main representatives; the seventh section presents the comparative study of the existing approaches; the final section concludes the paper.

Manuscript received 10 October 2013. Accepted for publication 5 November 2013. This article partially presents results from the PhD thesis "Automated conceptual database design based on business model", defended at Faculty of Electrical Engineering, University of Banja Luka.

D. Brdjanin and S. Maric are with the Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina (phone: +387-51-221-851, e-mail: {bdrazen,ms}@etfbl.net).

II. Classification of existing approaches

We conducted an extensive survey of the related literature and identified more than 70 primary sources (articles, conference papers, PhD theses) addressing the model-driven synthesis of the data model (MDSDM). Based on the primary focus of a source notation, all identified papers can be classified into four main groups: (i) *function-oriented*, (ii) *process-oriented*[1], (iii) *communication-oriented*[2], and (iv) *goal-oriented*. The corresponding paper distribution is given in Table I, while the taxonomy of source notations is presented in Fig. 1.

TABLE I
Distribution of papers according to source notation classification

Source notation	Function-oriented (FOM)	Process-oriented (POM)	Communication-oriented (COM)	Goal-oriented (GOM)
Number of papers	22	36	10	11

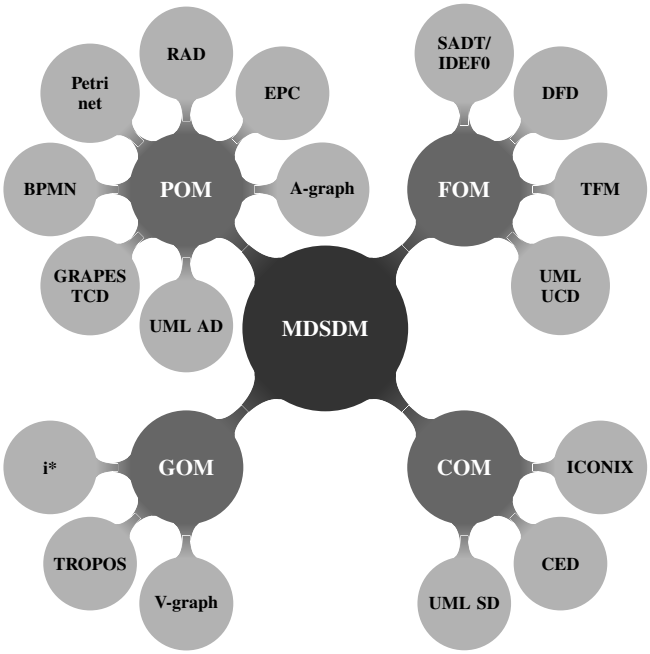

Fig. 1. Taxonomy of source notations in MDSDM.

The chronological overview of the identified MDSDM approaches is given in Fig. 2. The approaches are grouped by a source notation and then aggregated in accordance with the introduced classification. Different marks are used to differentiate the source model completeness and the level of automatization for the identified papers. The arrows are used to emphasize the related papers presenting the improvements in the same approach.

[1]The main criterion for the differentiation between the process-oriented and function-oriented notations is the capacity for the explicit representation of a workflow.

[2]Although some communication-oriented notations might be considered as process-oriented, the main reason for making a separate category lies in the fact that the given notations are focused on the interaction between the process participants.

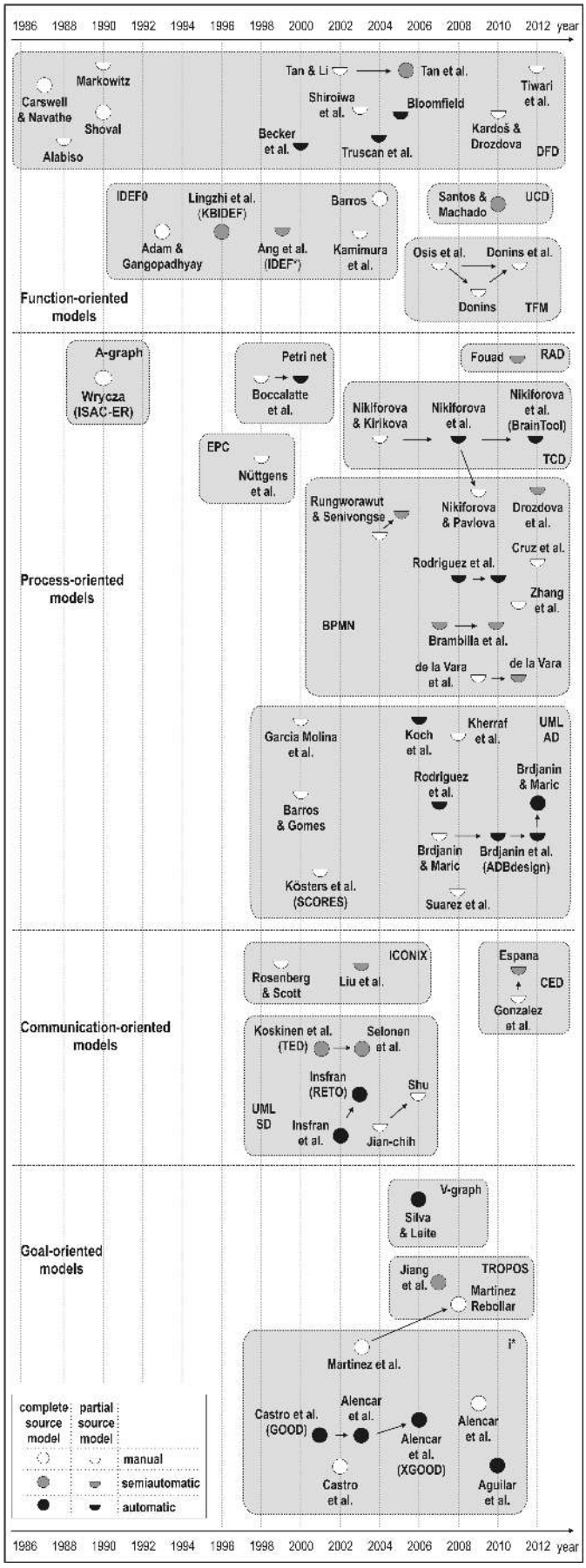

Fig. 2. Chronological overview of MDSDM approaches.

III. Function-oriented models as starting point

The first ideas about the MDSDM based on function-oriented models (FOMs) appeared in the second half of the 1980s, but the initial implementations of (semi)automatic generators did not occur before the mid-1990s.

The chronological overview of the identified approaches is presented in Fig. 2 (together with other categories).

Our survey shows that FOMs, used as a basis for the MDSDM, have been represented by four different notations (Fig. 1): DFD, SADT/IDEF0, TFM (Topological Functioning Model) and UML UCD (Use Case Diagram).

Since the Carswell & Navathe's SA-ER proposal [6] in 1987, the DFD [7] (including its modifications) has been a function-oriented notation commonly used ($>50\%$) as a starting point for the MDSDM. Among the 13 identified papers [6], [8]–[19], there is no paper that presents an automatic generator taking a complete model of functional system requirements represented by the DFD-hierarchy. Just one paper [13] presents a semiautomatic generator taking a complete source model represented by the DF Net hierarchy, while four papers [12], [15]–[17] present the automated generation of the data model based on an incomplete source model, i.e. a single DFD.

The SADT/IDEF0 is used in five papers [20]–[24]. Only two papers [21], [22] report the (semi)automated generation of the target data model. The TFM [25] is used in several papers [26]–[28], but all of them present only manual data model derivation from the TFM. The UML UCD is used in [29] for semiautomated data model synthesis.

The survey of the MDSDM approaches, taking FOMs as a starting point, implies:

- the majority of approaches ($>65\%$) take an incomplete model of functional requirements, i.e. a single diagram, as a basis for data model synthesis;
- the target data model is mainly represented by the UML class diagram ($>60\%$), but some other notations are also used (e.g. IDEF1/IDEF1X in [20]–[22], E-R in [6], [9], [23], NIAM in [10] and OODM in [11]);
- since the great majority of approaches ($\sim 90\%$) are based on guidelines and informal rules, the automatization level is very low ($\sim 60\%$ is manual, while less than 20% is automatic);
- the percentage of the evaluated approach is very low – only one [13] approach ($<5\%$) was evaluated based on a controlled experiment, while all others were not evaluated at all, or were just shown by some illustrative examples without any quantitative and/or qualitative evaluation;
- the semantic capacity of FOMs has not been sufficiently identified to enable the automatic synthesis of the target data model, since the existing approaches, which automatically generate the data model, do not generate an adequate data model structure regarding the proper number of classes, their associations and association end multiplicities.

IV. Process-oriented models as starting point

Process-oriented models (POMs) constitute the largest category of models being used as a starting point for the MDSDM. Although the first data model synthesis based on the POM (A-graph) was proposed by Wrycza [30] in 1990, the boom of these approaches was influenced by the appearance and development of metamodel-based notations, particularly the UML AD (UML activity diagram) in the late 1990s/early 2000s, and the BPMN (Business Process Model and Notation) a few years later, as well as the ATL [31] and QVT [32] transformation languages.

The survey shows that POMs, used as a basis for data model synthesis, have been represented by seven different notations (Fig. 1): UML AD, BPMN, GRAPES-BM/TCD, Petri Net, RAD (Role Activity Diagram), EPC (Event-driven Process Chain) and A-graph. The chronology of the identified approaches is presented as a part of Fig. 2.

The UML AD is used in 14 papers [33]–[46]. Only one paper [46] presents an automatic data model generator (named ADBDESIGN) based on the complete source model, i.e. a source model containing a finite set of UML ADs representing all business processes in a domain; several papers [36]–[38], [42]–[45] present the automated, mainly ATL- and QVT-based, data model generation based on the incomplete source model, but with a modest *precision* (measure showing the percentage of correct automatically generated concepts) and *recall* (automatically generated percentage of the target model), while the others present manual data model derivation.

Although the BPMN is used in 12 papers [47]–[58], there is no paper presenting an automatic generator of the data model based on the complete source model. There are two QVT-based proposals [50], [53], but with modest achievements in the automated generation of the analysis level class diagram, as well as several proposals [48], [49], [52], [56], [58] for the semiautomated generation.

There are also several related papers [59]–[63] proposing the usage of the TCD notation (a part of the GRAPES-BM language) as a starting point for data model synthesis, initially through the intermediate model, while the final release [63] presents the BRAINTOOL generator, which generates the data model directly from the TCD. However, like the majority of all proposals, they do not consider the complete source model, and the proposed approach is also not evaluated.

Among the remaining papers proposing data model synthesis based on Petri nets in [64], [65], EPC in [66], A-graph in [30], and RAD in [67], only two sources [65], [67] present software tools for the (semi)automated generation of the data model based on the incomplete source model.

The survey of the MDSDM approaches based on POMs implies:

- only two ($\sim 5\%$) papers [30], [46] take the complete source model, while others just consider a single diagram as a basis for data model synthesis;
- the target data model is predominantly represented by the UML class diagram ($>90\%$), while the E-R is used only in [30], [64], [65];

- the majority of proposed approaches are based on guidelines ($\sim 30\%$) and informal rules ($\sim 40\%$), but the development of transformation languages (ATL and QVT) has made an important contribution to the formalization and automatization of the MDSDM approaches in the recent years, so the participation of (semi)automatic techniques is significant ($\sim 60\%$);
- a small percentage ($\sim 10\%$) of approaches were evaluated based on a controlled experiment [56], [67] or a single case study [45], [46], while all others were not evaluated at all, or just illustrated by some examples without any quantitative/qualitative evaluation;
- only one paper [46] presents an automatic data model generator based on the complete source model, but the generated data model can be considered just as an initial data model, i.e. an analysis level data model;
- the semantic capacity of POMs has not yet been sufficiently identified to enable the automatic synthesis of the complete target data model, since the existing approaches still do not have a significant recall in the automated generation of some types of associations and class members.

V. Communication-oriented models as starting point

Communication-oriented models (COMs) constitute a smaller category of models used for data model synthesis. The survey shows that COMs, used as a starting point for data model synthesis, have been represented by three different notations (Fig. 1): UML SD (Sequence Diagram), ICONIX (Robustness Diagram) and CED (Communicative Event Diagram). The chronology of the identified approaches is presented in Fig. 2.

The UML SD [68] was identified in six (60% of this category) sources [69]–[74] as a source notation for data model synthesis. According to [72], transformations from sequence to class diagrams can be classified as *strong*, which implies that the source model possesses a semantic capacity only for the semiautomatic synthesis of the target data model, as in [69], [72]. The semantic capacity of the UML SD can be augmented by the specialization of the standard notation, and the process of data model synthesis can be automatic, as in the RETO tool [70], [71]. The process of data model synthesis in other two studies [73], [74] is manual. Among the remaining proposals for data model synthesis based on the CED [75] in [76], [77] and the ICONIX [78] in [78], [79], the semiautomatic synthesis of the data model is proposed in [77], [79], while others propose the guidelines for manual data model synthesis.

The survey of the MDSDM approaches based on COMs implies:

- the majority (60%) of papers take the incomplete source model as a basis for data model synthesis;
- the target data model is exclusively represented by the UML class diagram;
- although the majority (70%) of papers are based on guidelines and informal rules, the majority (60%) of papers still present (semi)automated techniques for data model synthesis;
- only one (10%) proposal [77] was evaluated based on a controlled experiment, while all others were not evaluated;
- only one primary study [71] presents an automatic data model generator based on the complete source model, but with a modest recall of some association types and without proper evaluation.

VI. Goal-oriented models as starting point

Goal-oriented models (GOMs), as the main artifacts produced in the early phases of goal-oriented requirements engineering processes, constitute the fourth category of models used for data model synthesis. The survey shows that GOMs, used as a starting point for data model synthesis, have been represented, as depicted in Fig. 1, by the i^* notation [80] and some i^*-originated notations like TROPOS [81], [82] and V-graph [83]. The chronology of the identified approaches is presented in Fig. 2.

i^* models were identified in eight ($> 70\%$ of the entire category) papers [81], [84]–[90] as a basis for data model synthesis. Most of the identified papers present the automatic synthesis of the data model to some extent, while only two papers [81], [84] present manual data model derivation. Most of the papers reporting automated data model synthesis are mutually related and present improvements in the same approach and the same tool named GOOD/XGOOD [85]–[88]. Other proposals based on TROPOS models in [91], [92] and V-graph in [93] are also mainly automated to some extent, but not evaluated.

The survey of the MDSDM approaches based on GOMs implies:

- since i^* and i^*-originated notations enable the representation of functional and non-functional requirements by a unique diagram (but with difficult "readability"), all identified papers take the complete source model as a basis for data model synthesis;
- the target data model is exclusively represented by the UML class diagram;
- the large majority ($> 90\%$) of papers are based on guidelines and informal rules;
- the majority ($> 60\%$) of papers present (semi)automated techniques for data model synthesis, but the recall might be estimated as insufficient; and
- there is no evaluated approach at all.

VII. Comparative analysis

All papers were analyzed according to the following criteria (results are presented in Fig. 3):

- ***Source notation*** – a notation used for the source model representation;
- ***Source model completeness*** – a source model might be considered as **complete** or **partial** (complete/partial representation of requirements);

- ***Target notation*** – a notation used for the data model representation;
- ***Level of automatization*** – data model synthesis might be considered as **manual** (not supported by any software tool), **semiautomatic** (supported by a tool, but the designer's assistance is still required) or **automatic** (without designer's assistance);
- ***Level of formalism*** – data model synthesis might be driven by **guidelines** (synthesis is generally described without rules for data model synthesis), **informal rules** (rules specified in a natural language) or by **formal rules** (rules formally represented by a transformation language, formal algorithm, predicate logic, etc.);
- ***Approach evaluation*** – an approach might be considered as **evaluated** (based on a **controlled experiment** or a **case study**) or **not evaluated**.

The analysis implies that POMs and FOMs are predominantly ($> 70\%$) used as a basis for the MDSDM. Three notations (UML AD, DFD, BPMN), belonging to these two categories, are used in half of all identified sources. The DFD, as a traditional function-oriented notation, still constitutes the subject of research in the field of MDSDM, but it loses the precedence to the newer, metamodel-based notations UML AD and BPMN, as well as the goal-oriented notations.

Most of the papers ($\sim 70\%$) do not take the complete source model, but only incomplete, i.e. the partial model of business or system requirements, as a basis for data model synthesis. The incompleteness of the source model is a typical characteristic of all categories, except for approaches having GOMs as a basis for data model synthesis, since all of them are based on the i^* notation, which captures "all" requirements by a single diagram.

The UML class diagram is used for the representation of the target data model in a large majority ($>85\%$) of all identified papers. Other notations were mainly used in some approaches before the adoption of the UML standard.

Data model synthesis is mainly ($\sim 80\%$) informally specified, equally by guidelines and informal rules, while just one-fifth of all papers specify data model synthesis in some formal way. Consequently, the level of automatization is rather low – a half of all papers present only manual data model derivation, while the other half present (semi)automated data model generation to some extent. The development of transformation languages (ATL and QVT) has made a significant contribution to the formalization and automatization of data model synthesis in the recent years, so the participation of automatic techniques ($\sim 35\%$) is growing.

A deeper analysis implies that the semantic capacity of graphically represented business/software requirements (regardless of their orientation) has not yet been sufficiently identified to enable the automatic synthesis of the complete data model, since a large majority of the existing approaches still do not have a significant recall in the automated generation of class associations, association end multiplicities and class members.

Apart from the low level of formalism and automatization, the insufficiently identified semantic capacity of models representing business/software requirements constitutes also the main cause for the almost complete absence of evaluation of the proposed approaches. More than 90% of the analyzed approaches are not evaluated at all, while the evaluation in the evaluated papers was mainly focused on the approach usability, but not on the qualitative/quantitative measures for the implemented tools and generated data models.

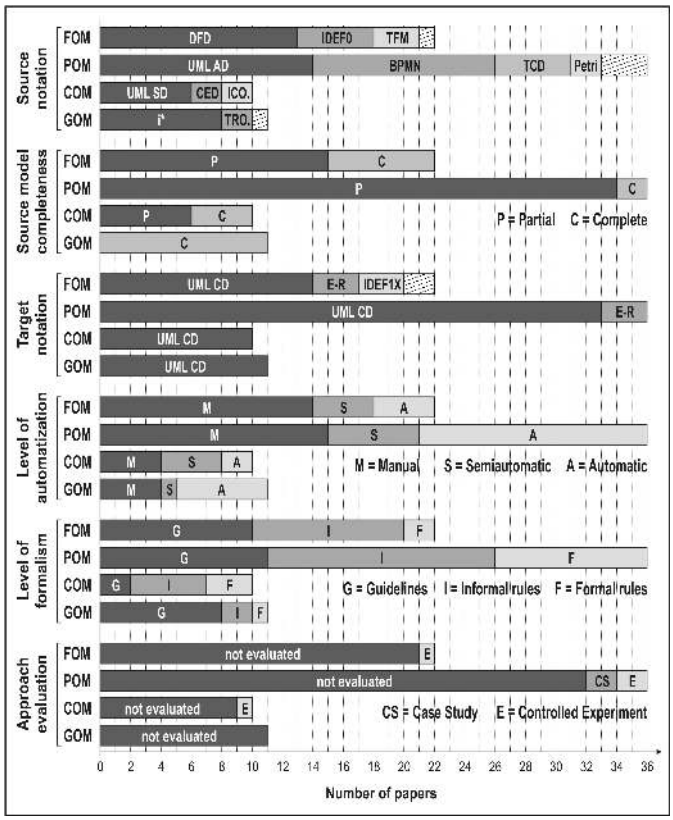

Fig. 3. Main characteristics of MDSDM approaches.

VIII. Conclusion

In this article we presented the results of the survey of model-driven techniques for data model synthesis. During the extensive research, we identified more than 70 research papers in the field and more than 15 different notations used for the source model representation.

Apart from the unique classification into four distinct groups: function-oriented, process-oriented, communication-oriented and goal-oriented, we gave the chronological overview of all identified papers classified according to the introduced classification, as well as the results of the evaluation based on several main criteria.

Although the idea of model-driven design of the data model is more than 25 years old, the survey shows that only a small number of papers present the implemented automatic model-driven generator of the data model and the corresponding evaluation results.

The main reason for the modest achievements in the automated model-driven design of the data model lies in the insufficiently identified semantic capacity of graphically represented business/software requirements for the automatic synthesis of the complete target data model.

REFERENCES

[1] Chen, P.P., "English sentence structure and entity-relationship diagrams," *Information Sciences*, vol. 29, no. 2-3, pp. 127–149, 1983.

[2] X. Franch, Maté, A., Trujillo, J.C., and C. Cares, "On the joint use of i* with other modeling frameworks: a vision paper," in *Proc. of RE '11*. IEEE, 2011, pp. 133–142.

[3] Jilani, A.A.A., M. Usman, A. Nadeem, Malik, Z.I., and Z. Halim, "Comparative study on DFD to UML diagrams transformations," *WCSIT*, vol. 1, no. 1, pp. 10–16, 2011.

[4] G. Loniewski, E. Insfran, and Abrahão, S., "A systematic review of the use of requirements engineering techniques in model-driven development," in *MODELS 2010*, ser. LNCS, Petriu, D.C, N. Rouquette, and Haugen, Ø., Eds. Springer-Verlag, Berlin Heidelberg, 2010, vol. 6395, pp. 213–227.

[5] T. Yue, Briand, L.C., and Y. Labiche, "A systematic review of transformation approaches between user requirements and analysis models," *Requirements Engineering*, vol. 16, no. 2, pp. 75–99, 2011.

[6] Carswell, J.L. and Navathe, S.B., "SA-ER: A methodology that links structured analysis and entity-relationship modeling for database design," in *Proc. of ER '87*. Elsevier, 1987, pp. 381–397.

[7] T. DeMarco, *Structured Analysis and System Specification*. New York, USA: Yourdon Press, 1978.

[8] Tan, H.B.K. and Li, W., "Systematic bridging the gap between requirements and OO design," in *Proc. of ASE 2002*. IEEE, 2002, pp. 249–252.

[9] Markowitz, V.M., "Representing processes in the Extended Entity-Relationship Model," in *Proc. of ICDE 1990*. IEEE, 1990, pp. 103–110.

[10] P. Shoval, "An integrated methodology for functional analysis, process design and database design," *Information Systems*, vol. 16, no. 1, pp. 49–64, 1990.

[11] B. Alabiso, "Transformation of data flow analysis models to object oriented design," in *Object-Oriented Programming Systems, Languages and Applications*, ser. SIGPLAN Notices, N. Meyrowitz, Ed. ACM Press, New York, 1988, vol. 23, pp. 335–353.

[12] Becker, L.B., Pereira, C.E., Dias, O.P., Teixeira, I.M., and Teixeira, J.P., "MOSYS: A methodology for automatic object identification from system specification," in *Proc. of ISORC 2000*. IEEE Computer Society, 2000, pp. 198–201.

[13] Tan, H.B.K., Yang, Y., and Blan, L., "Systematic transformation of functional analysis model in object oriented design and implementation," *IEEE Transaction on Software Engineering*, vol. 32, no. 2, pp. 111–135, 2006.

[14] M. Shiroiwa, T. Miura, and I. Shioya, "Meta model approach for mediation," in *Proc. of COMPSAC '03*. IEEE, 2003, pp. 480–485.

[15] D. Truscan, Fernandes, J.M., and J. Lilius, "Tool support for DFD-UML based transformation," in *Proc. of ECBS '04*. IEEE, 2004, pp. 378–387.

[16] Fernandes, J.M., J. Lilius, and D. Truscan, "Integration of DFDs into a UML-based model-driven engineering approach," *Software and Systems Modeling*, vol. 5, no. 4, pp. 403–428, 2006.

[17] T. Bloomfield, "MDA, meta-modelling and model transformation: Introducing new technology into the defence industry," in *ECMDA-FA 2005*, ser. LNCS, A. Hartman and D. Kreische, Eds. Springer-Verlag, Berlin Heidelberg, 2005, vol. 3748, pp. 9–18.

[18] Kardoš, M. and M. Drozdova, "Analytical method of CIM to PIM transformation in Model Driven Architecture (MDA)," *Journal of Information and Organizational Sciences*, vol. 34, no. 1, pp. 89–99, 2010.

[19] K. Tiwari, A. Tripathi, S. Sharma, and V. Dubey, "Merging of Data Flow Diagram with Unified Modeling Language," *International Journal of Scientific and Research Publications*, vol. 2, no. 8, pp. 1–6, 2012.

[20] N. Adam and A. Gangopadhyay, "Integrating functional and data modeling in a computer integrated manufacturing system," in *Proc. of ICDE 1993*. IEEE, 1993, pp. 302–309.

[21] L. Lingzhi, Ang, C.L., and Gay, R.K.L., "Integration of Information Model (IDEF1) with Function Model (IDEF0) for CIM Information System Design," *Expert Systems with Applications*, vol. 10, no. 3/4, pp. 373–380, 1996.

[22] Ang, C.L., Khoo, L.P., and Gay, R.K.L., "IDEF*: a comprehensive modelling methodology for the development of manufacturing enterprise systems," *Int. Journal of Production Research*, vol. 37, no. 17, pp. 3839–3858, 1999.

[23] M. Kamimura, K. Inoue, A. Hasegawa, R. Kawabata, S. Kumagai, and K. Itoh, "Integrated diagrammatic representations for data design in collaborative processes," *Journal of Integrated Design & Process Science*, vol. 7, no. 4, pp. 35–49, 2003.

[24] Barros, O.H., "Business information system design based on process patterns and frameworks," BPTrends, available: www.bptrends.com, 2004.

[25] Osis, J., "Topological model of system functioning," *Automatics and Computer Science*, vol. 6, pp. 44–50, 1969.

[26] J. Osis, E. Asnina, and A. Grave, "Computation independent modeling within the MDA," in *Proc. of SwSTE'07*. Los Alamitos, USA: IEEE, 2007, pp. 22–34.

[27] U. Donins, "Software development with the emphasis on topology," in *ADBIS 2009 Workshops*, ser. LNCS, J. Grundspenkis and et al., Eds. Springer-Verlag, Berlin Heidelberg, 2010, vol. 5968, pp. 220–228.

[28] U. Donins, J. Osis, A. Slihte, E. Asnina, and B. Gulbis, "Towards the refinement of topological class diagram as a platform independent model," in *Proc. of MDA/MDSD'11*. SciTePress, 2011, pp. 79–88.

[29] Santos, M.Y. and Machado, R.J., "On the derivation of class diagrams from use cases and logical software architectures," in *Proc. of ICSEA '10*. IEEE, 2010, pp. 107–113.

[30] S. Wrycza, "The ISAC-driven transition between requirements analysis and ER conceptual modelling," *Information Systems*, vol. 15, no. 6, pp. 603–614, 1990.

[31] F. Jouault, F. Allilaire, J. Bezivin, and I. Kurtev, "ATL: A model transformation tool," *Science of Computer Programming*, vol. 72, no. 1-2, pp. 31–39, 2008.

[32] OMG, "MOF 2.0 Query/View/Transformation Specification, v1.0," OMG, 2008.

[33] J. Garcia Molina, M. Jose Ortin, B. Moros, J. Nicolas, and A. Troval, "Towards use case and conceptual models through business modeling," in *ER 2000*, ser. LNCS, Laender, A.H.F., Liddle, S.W., and Storey, V.C., Eds. Springer-Verlag, Berlin Heidelberg, 2000, vol. 1920, pp. 281–294.

[34] J. Barros and L. Gomes, "From activity diagrams to class diagrams," in *Workshop Dynamic Behaviour in UML Models: Semantic Questions, In conjunction with Third Int. Conf. on UML*, York, UK, 2000.

[35] G. Kösters, Six, H-W., and M. Winter, "Couppling Use Cases and Class Models as a Means for Validation and Verification of Requirements Specifications," *Requirements Engineering*, vol. 6, no. 1, pp. 3–17, 2001.

[36] N. Koch, "Transformation Techniques in the Model-Driven Development Process of UWE," in *Proc. of the Workshops at ICWE'06, Art. No. 3*. ACM, 2006.

[37] N. Koch, G. Zhang, and Escalona, M.J., "Model Transformations from Requirements to Web System Design," in *Proc. of ICWE'06*. ACM, 2006, pp. 281–288.

[38] A. Rodriguez, E. Fernandez-Medina, and M. Piattini, "Analysis-level classes from secure business processes through model transformations," in *TrustBus 2007*, ser. LNCS, C. Lambrinoudakis, G. Pernul, and Tjoa, A.M., Eds. Springer-Verlag, Berlin Heidelberg, 2007, vol. 4657, pp. 104–114.

[39] D. Brdjanin and S. Maric, "An example of use-case-driven conceptual design of relational database," in *Proc. of Eurocon 2007*. IEEE, 2007, pp. 538–545.

[40] E. Suarez, M. Delgado, and E. Vidal, "Transformation of a process business model to domain model," in *Proc. of WCE 2008*. IAENG, 2008, pp. 165–169.

[41] S. Kherraf, E. Lefebvre, and W. Suryn, "Transformation from CIM to PIM using patterns and archetypes," in *Proc. of ASWEC '08*. Los Alamitos, USA: IEEE Computer Society, 2008, pp. 338–346.

[42] D. Brdjanin, S. Maric, and D. Gunjic, "ADBdesign: An approach to automated initial conceptual database design based on business activity diagrams," in *ADBIS 2010*, ser. LNCS, B. Catania, M. Ivanovic, and B. Thalheim, Eds. Springer-Verlag, Berlin Heidelberg, 2010, vol. 6295, pp. 117–131.

[43] D. Brdjanin and S. Maric, "Towards the initial conceptual database model through the UML metamodel transformations," in *Proc. of Eurocon 2011*. IEEE, 2011, pp. 1–4.

[44] Brdjanin, D. and Maric, S., "On automated generation of associations in conceptual database model," in *ER Workshops 2011*, ser. LNCS, De Troyer, O. and et al., Eds. Springer-Verlag, Berlin Heidelberg, 2011, vol. 6999, pp. 292–301.

[45] D. Brdjanin and S. Maric, "An Approach to Automated Conceptual Database Design Based on the UML Activity Diagram," *Computer Science and Information Systems*, vol. 9, no. 1, pp. 249–283, 2012.

[46] Brdjanin, D. and Maric, S., "Towards the automated business model-driven conceptual database design," in *Advances in Databases and Information Systems*, ser. AISC, T. Morzy, T. Harder, and R. Wrembel, Eds. Springer-Verlag, Berlin Heidelberg, 2012, vol. 186, pp. 31–43.

[47] W. Rungworawut and T. Senivongse, "From business world to software world: Deriving class diagrams from business process models," in *Proc. of the 5th WSEAS Int. Conf. on Aplied Informatics and Communications*. WSEAS, 2005, pp. 233–238.

[48] Rungworawut, W. and Senivongse, T., "Using ontology search in the design of class diagram from business process model," *PWASET*, vol. 12, pp. 165–170, 2006.

[49] M. Brambilla, J. Cabot, and S. Comai, "Automatic generation of workflow-extended domain models," in *MoDELS 2007*, ser. LNCS, G. Engels and et al., Eds. Springer-Verlag, Berlin Heidelberg, 2007, vol. 4735, pp. 375–389.

[50] A. Rodriguez, E. Fernandez-Medina, and M. Piattini, "Towards obtaining analysis-level class and use case diagrams from business process models," in *ER Workshops 2008*, ser. LNCS, Song, I.Y. and et al., Eds. Springer-Verlag, Berlin Heidelberg, 2008, vol. 5232, pp. 103–112.

[51] de la Vara, J.L., Fortuna, M.H., Sanchez, J., Werner, C.M.L, and Borges, M.R.S, "A requirements enegineering approach for data modelling of process-aware information systems," in *BIS 2009*, ser. LNBIP, W. Abramowicz, Ed. Springer-Verlag, Berlin Heidelberg, 2009, vol. 21, pp. 133–144.

[52] M. Brambilla, J. Cabot, and S. Comai, "Extending conceptual schemas with business process information," *Advances in Software Engineering, vol. 2010, Article ID 525121*, 2010.

[53] A. Rodriguez, Garcia-Rodriguez de Guzman, I., Fernandez-Medina, E., and Piattini, M., "Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach," *Information and Software Technology*, vol. 52, no. 9, pp. 945–971, 2010.

[54] J. Zhang, P. Feng, Z. Wu, D. Yu, and K. Chen, "Activity based CIM modeling and transformation for business process systems," *International Journal of Software Engineering and Knowledge Engineering*, vol. 20, no. 3, pp. 289–309, 2010.

[55] O. Nikiforova and N. Pavlova, "Application of BPMN instead of GRAPES for two-hemisphere model driven approach," in *ADBIS 2009 Workshops*, ser. LNCS, J. Grundspenkis and et al., Eds. Springer-Verlag, Berlin Heidelberg, 2010, vol. 5968, pp. 185–192.

[56] de la Vara, J.L., "Business process-based requirements specification and object-oriented conceptual modelling of information systems," PhD Thesis, Valencia Polytechnic University, 2011.

[57] Cruz, E.F., Machado, R.J., and Santos, M.Y., "From business process modeling to data model: A systematic approach," in *Proc. of QUATIC 2012*. IEEE, 2012, pp. 205–210.

[58] M. Drozdová, M. Mokryš, M. Kardoš, Z. Kurillová, and J. Papán, "Change of paradigm for development of software support for elearning," in *Proc. of ICETA 2012*. IEEE, 2012, pp. 81–84.

[59] O. Nikiforova and M. Kirikova, "Two-Hemisphere model driven approach: Engineering based software development," in *CAiSE 2004*, ser. LNCS, A. Persson and J. Stirna, Eds. Springer-Verlag, Berlin Heidelberg, 2004, vol. 3084, pp. 219–233.

[60] O. Nikiforova and N. Pavlova, "Open work of two-hemisphere model transformation definition into UML class diagram in the context of MDA," in *CEE-SET 2008*, ser. LNCS, Z. Huzar and et al., Eds. Springer-Verlag, Berlin Heidelberg, 2011, vol. 4980, pp. 118–130.

[61] O. Nikiforova, N. Pavlova, and J. Grigorjevs, "Several facilities of class diagram generation from two-hemisphere model in the framework of MDA," in *Proc. of ISCIS '08*. IEEE, 2008, pp. 1–6.

[62] O. Nikiforova and N. Pavlova, "Foundations on generation of relationships between classes based on initial business knowledge," in *Information Systems Development: Towards a Service Provision Society*, Papadopoulos, G.A. and et al., Eds. Springer, New York - Dordrecht, 2009, pp. 289–298.

[63] O. Nikiforova, K. Gusarovs, O. Gorbiks, and N. Pavlova, "BrainTool: A tool for generation of the UML class diagrams," in *Proc. of ICSEA 2012*. IARIA, 2012, pp. 60–69.

[64] A. Boccalatte, D. Giglio, and M. Paolucci, "An object-oriented modeling approach based on entity-relationship diagrams and Petri nets," in *Proc. of ICSMC 1998*. IEEE, 1998, pp. 1347–1352.

[65] Boccalatte, A., Giglio, D., and Paolucci, M., "ISYDES: the project of a tool aimed at information system development," in *Proc. of AIWORC 2000*. IEEE, 2000, pp. 293–298.

[66] M. Nüttgens, T. Feld, and V. Zimmermann, "Object-orientation in business process modeling through applying event driven process chains (EPC) in UML," in *Proc. of the UML - Technical Aspects and Applications*. Physica-Verlag, 1998, pp. 250–261.

[67] A. Fouad, "Embedding requirements within the model driven architecture," PhD Thesis, Bournemouth University, 2011.

[68] OMG, "Unified Modeling Language: Superstructure, v2.4.1," OMG, 2011.

[69] J. Koskinen, J. Peltonen, P. Selonen, T. Systa, and K. Koskimies, "Model processing tools in UML," in *Proc. of ICSE 2001*. IEEE Computer Society, 2001, pp. 819–820.

[70] E. Insfran, O. Pastor, and R. Wieringa, "Requirements Engineering-Based Conceptual Modelling," *Requirements Engineering*, vol. 7, no. 2, pp. 61–72, 2002.

[71] Insfran, E., "Requirements engineering approach for object-oriented conceptual modeling," PhD Thesis, Valencia Polytechnic University, 2003.

[72] P. Selonen, K. Koskimies, and M. Sakkinen, "Transformations Between UML Diagrams," *Journal of Database Management*, vol. 14, no. 3, pp. 37–55, 2003.

[73] Jian-chih, L., "Transformation from sequence diagram to class diagram," National Sun Yat-sen University, Taiwan, 2004.

[74] Hsu, C.T., "A methodology for transformation from sequence diagram to class diagram," National Sun Yat-sen University, Taiwan, 2006.

[75] S. España, A. González, and O. Pastor, "Communication analysis: a requirements engineering method for information systems," in *CAiSE 2009*, ser. LNCS, van Eck, P., J. Gordijn, and R. Wieringa, Eds. Springer-Verlag, Berlin Heidelberg, 2009, vol. 5565, pp. 530–545.

[76] A. González, S. España, M. Ruiz, and O. Pastor, "Systematic derivation of class diagrams from communication-oriented business process models," in *Enterprise, Business-Process and Information Systems Modeling*, ser. LNBIP, Halpin, T.A. and et al., Eds. Springer-Verlag, Berlin Heidelberg, 2011, vol. 81, pp. 246–260.

[77] España, S., "Methodological integration of communication analysis into a model-driven software development framework," PhD Thesis, Valencia Polytechnic University, 2011.

[78] D. Rosenberg and K. Scott, *Use Case Driven Object Modeling with UML*. Reading, USA: Addison-Wesley professional, 1999.

[79] D. Liu, K. Subramaniam, B. Far, and A. Eberlein, "Automating Transition from Use-cases to Class Model," in *Proc. of CCECE 2003*. IEEE, 2003, pp. 831–834.

[80] Yu, E., "Modelling strategic relationships for process reengineering," PhD Thesis, University of Toronto, 1995.

[81] J. Castro, M. Kolp, and J. Mylopoulos, "Towards requirements-driven information systems engineering: Tropos project," *Information Systems*, vol. 27, no. 6, pp. 365–389, 2002.

[82] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology," *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, 2004.

[83] Y. Yu, Leite, J.C.S.P., and J. Mylopoulos, "From goals to aspects: Discovering aspects from requirements goal models," in *Proc. of RE '04*. IEEE, 2004, pp. 38–47.

[84] A. Martinez, J. Castro, O. Pastor, and H. Estrada, "Closing the gap between organizational modeling and information system modeling," in *Proc. of WER 2003*, 2003, pp. 93–108.

[85] Castro, J.F., Alencar, F.M.R., Filho, G.A.C., and Mylopoulos, J., "Integrating organizational requirements and object oriented modeling," in *Proc. of ISRE 2001*. IEEE, 2001, pp. 146–153.

[86] Alencar, F.M.R., Filho, G.A.C., and Castro, J.F., "Support for structuring mechanism in the integration of organizational requirements and object oriented modeling," in *Proc. of WER 2002*, 2002, pp. 147–161.

[87] F. Alencar, F. Pedroza, J. Castro, and R. Amorim, "New mechanisms for the integration of organizational requirements and object oriented modeling," in *Proc. of WER 2003*, 2003, pp. 109–123.

[88] Alencar, F.M.R, Pedroza, F.P., J. Castro, Silva, C.T.L., and Ramos, R.A., "XGOOD: A tool to automatize the mapping rules between i* framework and UML," in *Proc. of CIbSE 2006*, 2006, pp. 125–138.

[89] F. Alencar, Marín, B., G. Giachetti, O. Pastor, and Pimentel, J.H., "From i* requirements models to conceptual models of a model driven development process," in *POEM 2009*, ser. LNBIP, A. Persson and J. Stirna, Eds. Springer-Verlag, Berlin Heidelberg, 2009, vol. 39, pp. 99–114.

[90] Aguilar, J.A., Garrigós, I., Mazón, J.N., and Trujillo, J., "An MDA approach for goal-oriented requirement analysis in web engineering," *Journal of Universal Computer Science*, vol. 16, no. 17, pp. 2475–2494, 2010.

[91] L. Jiang, T. Topaloglou, A. Borgida, and J. Mylopoulos, "Goal-oriented conceptual database design," in *Proc. of RE '07*. Los Alamitos, USA: IEEE, 2007, pp. 195–204.

[92] Martinez Rebollar, A., "Conceptual schemas generation from organizational models in an automatic software production process," PhD Thesis, Valencia Polytechnic University, 2008.

[93] Silva, L.F. and Leite, J.C.S.P., "Generating requirements views: A transformation-driven approach," *Electronic Communications of the EASST*, vol. 3, pp. 1–14, 2006.