A Systematic Investigation of Tools in Model Based System Engineering for Embedded Systems

Muhammad Rashid[1]
Computer Engineering Department,
College of Computer and Information Systems,
Umm Al-Qura University, Saudi Arabia.
mfelahi@uqu.edu.sa

Muhammad Waseem Anwar[2]
Consultant (Research & Development),
MODEVES Project, National Science,
Technology and Innovation plan, Saudi Arabia.
waseemanwar@hotmail.com

Aamir M. Khan[3]
College of Engineering,
University of Buraimi,
Buraimi, Oman.
aamir.m@uob.edu.om

***Abstract*—Model Based System Engineering (MBSE) approach has been frequently used for embedded systems as it is highly supportive for early design verification. However, selection of appropriate tools to perform various MBSE activities is always challenging. In this paper, latest MBSE tools have been investigated in contemporary research practices through Systematic Literature Review (SLR). This facilitates practitioners and researchers to select appropriate MBSE tools according to their requirements.**

Keywords— MBSE; tools; transformation; verification; embedded systems

1. INTRODUCTION

Development of embedded systems is always challenging due to the diversity of their behavioral and temporal aspects. The situation gets more complex while achieving the business objectives like time-to-market and productivity demands. These circumstances urge to verify the design of embedded systems as early as possible. Consequently, MBSE approach has been commonly put in to practice for embedded systems development [11-12][18-19][21-23].

The primary MBSE activity, as shown in Fig. 1, is to specify embedded systems requirements. The practice of UML and its SYSML / MARTE profiles is common to model embedded system requirements [1-20]. In addition, different property specification approaches [31-32][55-57] have also been used to specify behavioral / temporal aspects of embedded systems in the model. Once the requirements are modeled, MBSE model transformation activity has been performed to get the desired output model for further verification and validation (simulation) of the design. There are two commonly used transformation approaches i.e. Model-to-Model (M2M) [2-3] and Model-to-Text (M2T) [17-18].

Model verification activity is performed to ensure the correctness of model / system. The use of formal verification approaches [47-49] is common in this activity. During verification activity, any errors found in the design of the system should be corrected as shown in Fig. 1. Validation of the model / system is accomplished through simulation. The target source code has been generated, by making use of appropriate transformation approach, and used for simulation of the model / system.

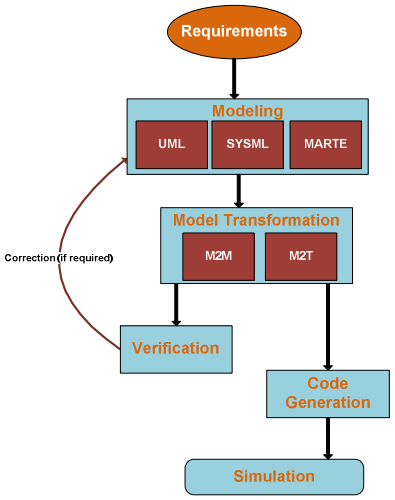

Fig. 1. Major MBSE Activities for Embedded Systems

Although MBSE approach is highly supportive for embedded systems development, it is always challenging for practitioners and researchers to select appropriate tools in order to perform various MBSE activities. Therefore, in this paper, latest MBSE tools have been investigated in contemporary research practices through Systematic Literature Review (SLR). The overview of research is shown in Fig. 2.

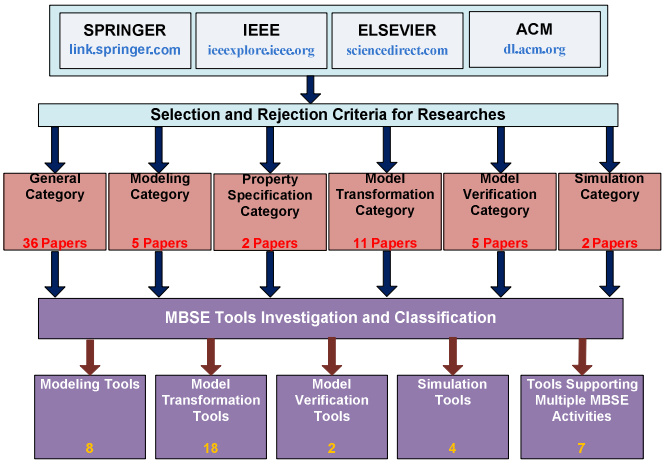

Fig. 2. Overview of Research

A Review protocol has been developed (Section 2.2) that incorporates selection and rejection criteria. The search process is carried out by using four scientific databases, as shown in Fig. 2. We define six categories (Section 2.1) to classify 61 selected researches. The tools for each MBSE activity have been identified in Section 3. Finally, Section 4 concludes the paper.

2. RESEARCH METHODOLOGY

Systematic Literature Review [64] has been used for investigation of tools and described in the following:

2.1 Categories Definition

Six categories have been defined to classify selected researches:

General Category: The general category contains all research works providing solution of more than one MBSE activity (e.g. modeling, model transformation and verification) simultaneously for the development of embedded systems.
Modeling Category: Modeling embedded system requirements is the primary MBSE activity. Therefore, there is a fair possibility that all other categories might contain some modeling related information. However, modeling category contains research works where only modeling aspects of embedded systems are discussed / researched by making use of UML and its SYSML / MARTE profiles.
Property Specification Category: This category contains research works where only property specification techniques / languages have been discussed / researched for specifying behavioral / temporal aspects of embedded systems.
Model Transformation Category: This category contains research works where only model transformation techniques / approaches have been discussed / researched. Particularly, the application of M2M and / or M2T transformation approaches has been discussed / researched for the development of embedded systems.
Model Verification Category: This category contains research works where only model verification approaches (both formal and informal) have been discussed / researched to ensure the correctness of behavioral / temporal aspects of the model. .
Simulation Category: This category contains research works where only simulation approaches / tools have been discussed / researched in order to validate the model.

It is important to mention here that the *General Category* contains a number of researches where solutions are either provided for all MBSE activities or at least two MBSE activities simultaneously for the development of embedded systems.

2.2 Review Protocol Development

Once the categories are defined, we develop review protocol on the basis of predefined SLR standards [64]. Consequently, the developed protocol defines the selection and rejection criteria, search process, quality assessment and data synthesis of the extracted data. The details are given in subsequent sections:

2.2.1 Selection and Rejection Criteria

We develop concrete selection and rejection criteria in order to get the answer of our research question. The selection and rejection of a particular research work is based on the six parameters given below:

Subject-Relevant: Select the research work only if it is relevant to one of the six predefined categories (Section 2.1).
2008-2014: Select the research work only if it is published from 2008-2014.
Publisher: Select the research work only if it is published in one of the four renowned scientific databases i.e. IEEE [62], SPRINGER [66], ELSEVIER [63] and ACM [67].
Crucial-effects: Select the research work only if it has significant positive effects regarding embedded system development through MBSE approach.
Results-oriented: Select the research work only if its proposal and ultimate outcomes must be supported by solid facts and experimentation.
Repetition: All the researches in a particular research context cannot be included. Consequently, reject researches those are identical in the given research context and only one of them is selected.

2.2.2 Search Process

We select four scientific databases (i.e. IEEE, ELSEVIER, SPRINGER and ACM) for search process as given in selection and rejection criteria (Section 2.2.1). These scientific databases provide high quality journal and conference proceedings. We use different search terms like MBSE, SYSML, MARTE, embedded systems etc to accomplish the search process. We also use AND / OR operators to get the most relevant search results. The details of search terms and relevant screenshots can be viewed at [68].

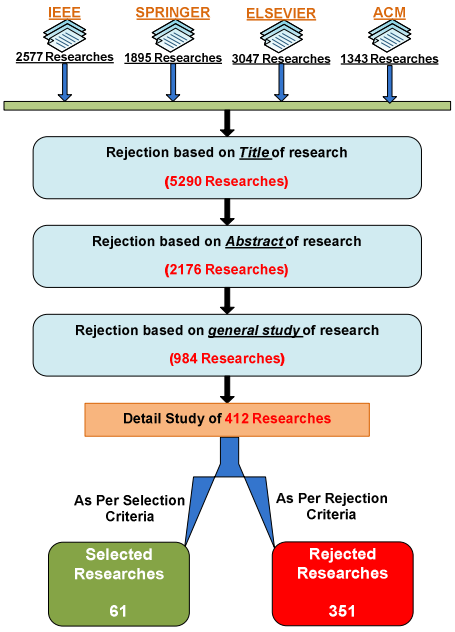

Fig. 3. Search Process

The following steps have been performed during the search process (depicted in Fig. 3):

1. We specify various search terms in four scientific databases and analyze approximately 8862 search results as per selection and rejection criteria.
2. We discard 5290 researches by reading their Title as per selection and rejection criteria.
3. We discard 2176 researches by reading their Abstract.
4. We perform general study of 1396 researches by reading different relevant sections of each research. We discard 984 researches those do not meet selection and rejection criteria. We select remaining 412 relevant researches for detailed study.
5. We perform detailed study of 412 researches and discard 351 researches.
6. Finally, we select 61 researches, fully compliance with the pre-defined selection and rejection criteria

2.2.3 Quality Assessment

We have developed quality criterion to understand the important outcomes of selected researches. The developed criterion also defines the credibility of each selected research and its decisive findings. For example, the significant parameters like validation method and originality of the research have been evaluated in the quality criterion. The complete details of quality assessment can be viewed at [69].

2.2.4 Data Extraction and Synthesis

For data extraction, we extract significant details of each selected research work to ensure its compliance with the selection and rejection criteria. However, we cannot include the details of data extraction in the article due to length restrictions. Therefore, the data extraction details of all 61 selected research works can be found at [70]. For data synthesis, we perform detailed analysis of each selected research work in order to identify the tools which are used to perform different MBSE activities for embedded systems.

3. RESULTS

On the basis of SLR, we identified overall 39 tools, frequently used in selected research works to perform different MBSE activities. It is observed that there are some tools which are used in various selected research works to perform a particular MBSE activity. Consequently, relevant researches are identified against each tool as given in the tables, presented in the subsequent sections.

3.1 Modeling Tools

On the basis of SLR, we identify eight tools, used / discussed in the selected research works in order to model embedded systems requirements as given in Table I. However, there is a fair possibility that some of the identified modeling tools might also provide support for other MBSE activities as well. For example, Topcased [71] can also be used to perform MBSE model transformation and verification activities for embedded systems. However, Topcased is only used to model embedded systems requirements in various selected research works [3][8][9][52]. Consequently, it is identified as a modeling tool. Similarly, such possibilities might also exist for model transformation, verification and simulation tools identification. So these possibilities are beyond the objective of this research.

Table I. MBSE modeling tools for embedded systems

Sr. #	Tool Name	Relevant Researches
1	Topcased [71]	[3][8][9][52]
2	Modelio Editor [72]	[12]
3	Magic Draw [73]	[6][16][20]
4	Eclipse GEF [74]	[17]
5	Rhapsody [75]	[4][19][22][57]
6	PapyrusMDT [76]	[21][44][60]
7	EA MDG [77]	[27]
8	Visual Paradigm [78]	[60]

3.2 Model Transformation Tools

We identify 18 tools, used in the selected research works, to perform MBSE model transformation for embedded systems, as given in Table II.

Table II. MBSE model transformation tools for embedded systems

Sr. #	Tools Name	Relevant Researches
1	MediniQVT [79]	[2]
2	ATL [80]	[3][8][39][44][46][50][59]
3	Xpand [81]	[8][20]
4	Acceleo [82]	[9][21][42][44]
5	MODCO [83]	[13]
6	MODEASY [17]	[17]
7	Apache Velocity [84]	[16][18]
8	UML-RSDS [85]	[46]
9	GenERTiCA [86]	[18]
10	MDworkbench [87]	[19][22]
11	CatapultC [88]	[22]
12	QVTO [89]	[14]
13	AGG [90]	[38][39][40]
14	MOFScript [91]	[29][39][42][43]
15	MeTHAGeM [50]	[50]
16	Epsilon [92]	[12][58]
17	TMC [93]	[60]
18	SYSML to B translator [94]	[61]

It has been observed that Model-to-Model (M2M) and Model-to-Text (M2T) are most commonly used transformations approaches. For example, Acceleo [82] is frequently used for M2T transformation and ATL [80] is commonly used for M2M transformation. Some tools are even used to perform both M2M and M2T transformations within a single research work. For example, Epsilon [92] is used by Quadri et al. [12] to perform both transformation approaches. Further, researchers even develop their own transformation tools to accomplish particular research objective. SYSML to B translator [94] is a typical example of this trend.

3.3 Model Verification Tools

On the basis of SLR, we identify 2 model verification tools, as given in Table III.

Table III. MBSE model verification tools for embedded systems

Sr. #	Tool Name	Relevant Researches
1	PRISM [95]	[7][52]
2	UPPAAL [96]	[55][59]

It has been observed that the utilization of available model verification tools normally result in additional model transformation complexity. For example, it is very difficult to transform the model according to particular requirements of verification tool. Therefore, researchers prefer to develop their own model verification mechanism rather to use available verification tools The work presented in [35] is a typical example. Consequently, the verification tools are rarely used / discussed in selected research works so we identify only two verification tools as given in Table III.

3.4 Simulation Tools

As far as the simulation category is concerned, our investigations based on SLR allow us to identify 4 simulation tools, as shown in Table IV. It has been analyzed that it is always demanding to select the appropriate simulation tools as these are highly dependent on their own working environment. For example, SIMULINK [65] tool only simulates the models, developed in its own working environment and additional efforts are required to simulate other models such as models developed in SYSML / MARTE etc. In addition, it is not possible to simulate the code of different hardware description languages through a single simulation tool.

Table IV. MBSE simulation (validation) tools for embedded systems

Sr. #	Tool Name	Relevant Researches
1	SIMULINK [65]	[4][25]
2	ActiveHDL [97]	[14]
3	QEMU [98]	[15]
4	GTKWave [99]	[36]

3.5 Tools Supporting Multiple MBSE Activities

From the SLR, it has been analyzed that there are some tools, used in the latest researches to perform more than one MBSE activities simultaneously. We identify seven such tools as given in Table V.

Table V. Tools supporting multiple MBSE activities for embedded systems. Abbreviations used: Modeling = M, Model Transformation = MT, Model Verification = MV, Simulation = S

Sr. #	Tool Name	MBSE Activities	Relevant Researches
1	radCASE [1]	M, MT	[1]
2	Artisan Studio [100]	M, MT	[15]
3	Verilog Dynamic Verifier [36]	MT, MV	[36]
4	radCHECK [1]	MV, S	[1]
5	Zot [101]	MV, S	[12][58]
6	Gaspard2 [102]	M, MT, S	[23][26]
7	TTool [103]	M, MV, S	[55]

It has also been analyzed that researchers develop their own tools to perform different MBSE activities for embedded systems development. Consequently, these tools are identified and given in Table V. For example, Guglielmo et al. [1] develop radCASE and radCHECK tools to perform various MBSE activities. Furthermore, it has been observed that the researchers frequently customize the available open source tools in order to meet their particular requirements. For example, Daniel et al. [55] customize TTool [103] to achieve the specified research objective.

3.6 Discussion

From the findings of the current research, it can be concluded that eclipse is the leading platform, providing variety of MBSE tools for embedded systems. From the analysis of the modeling tools, Topcased, Rhapsody, Papyrus and Magic Draw are the most popular and frequently used tools. Topcased and Papyrus are more attractive to research community due to their open source license. An interesting fact is that Papyrus is integrated in Topcased to perform modeling tasks. On the other hand, ATL, Acceleo, MOFScript and Epsilon are the popular model transformation tools. Similarly, SIMULINK is the leading simulation tool. Although we highlight the popular and most frequently used tools in the discussion, it does not guarantee the applicability of these tools for the accomplishment of particular MBSE requirements. Each tool has its own strength and weaknesses. Therefore, the selection of a tool is always based on the MBSE requirements under consideration.

3.7 Limitations of Research

Although, SLR guidelines and review protocol is strictly followed to perform this research, there are still few limitations. For example, few search terms return thousands of results so it is not possible to completely scan all results. Moreover, several research works have been rejected by reading their titles and it is possible that research contents are not properly defined in the title. Furthermore, we select four renowned scientific databases for our research. However, there are various other scientific databases those also provide latest research works in journal and conference proceedings. Therefore, there is fair possibility that we miss some relevant research works from other scientific databases. However, we are confident that ultimate results of our research should not be affected much because selected scientific databases provide plentiful research works.

4. CONCLUSION AND FUTURE WORK

This paper investigates the latest tools to perform Model Based System Engineering (MBSE) activities for embedded systems development. To achieve this objective, Systematic Literature Review (SLR) has been performed to identify 61 research works where MBSE approach is put into practice for the development of embedded systems. The selected researches have been intensively investigated in order to identify 39 MBSE tools. These tools are further classified according to their relevance with the particular MBSE activity. This facilitates practitioners and researchers to select appropriate MBSE tools according to their requirements.

In future, we intend to perform further investigations on identified tools by considering various important tools characteristics those are not considered in this paper. For example, we will try to investigate the general tools characteristics like license type and supported platforms. In addition, we will also try to investigate MBSE activity specific characteristics. For example, considering MBSE modeling characteristics, we will try to investigate the modeling tools regarding the UML and its SYSML / MARTE profiles support.

REFERENCES

[1] Di Guglielmo et al.: On the integration of model-driven design and dynamic assertion-based verification for embedded software, Journal of Systems and Software, Vol. 86, Issue 8, August 2013, Pages 2013–2033.

[2] G. Kapos, V. Dalakas, A. Tsadimas, M. Nikolaidou and D. Anagnostopoulos: Model-based system engineering using SysML: Deriving executable simulation models with QVT, 8th Annual IEEE Systems Conference 2014, Pages 531-538.

[3] S. Berrani, A. Hammad, H. Mountassir: Mapping SysML to Modelica to Validate Wireless Sensor Networks Non-Functional Requirements, 11th Int. Symposium on Programming and Systems 2013, Pages 177-186.

[4] T. Sakairi, E. Palachi, C. Cohen, Y. Hatsutori, J. Shimizu, and H. Miyashita: Designing a control system using SysML and Simulink, Proceedings of SICE Annual Conference 2012.

[5] Stancescu et al.: A SysML model for code correction and detection systems, 33rd International Convention MIPRO 2010, Pages 189-191.

[6] A. Tsadimas, G. Kapos, V. Dalakas, M. Nikolaidou and D. Anagnostopoulos: Integrating simulation capabilities into SysML for enterprise information system design, 9th International Conference on System of Systems Engineering 2014, Pages 272-277.

[7] S. Ouchani, O. A. Mohamed and M. Debbabi: A Formal Verification Framework for BlueSpec System Verilog, Forum on Specification and Design Languages (FDL) 2013, Pages 1-7.

[8] F. Bouquet, J. Gauthier, A. Hammad and F. Peureux: Transformation of SysML Structure Diagrams to VHDL-AMS, Second Workshop on Design, Control and Software Implementation for Distributed MEMS (dMEMS) 2012, Pages 74-81.

[9] G. DeTommasi, R. Vitelli, L. Boncagni, and A. C. Neto: Modeling of MARTe-Based Real-Time Applications With SysML, IEEE Tran. on Industrial Informatics 2013, Volume 9, Issue 4, Pages 2407-2415.

[10] C. Gomez, J. DeAntoni and F. Mallet: Multi-View Power Modeling based on UML MARTE and SysML, 38th EUROMICRO Conference on Software Engineering and Advanced Applications 2012.

[11] E. Andrade, P. Maciel, G. Callou and B. Nogueira: A Methodology for Mapping SysML Activity Diagram to Time Petri Net for Requirement Validation of Embedded Real-Time Systems with Energy Constraints, International Conference on Digital Society 2009.

[12] I. R. Quadri et al: MADES FP7 EU Project: Effective High Level SysML/MARTE Methodology for Real-Time and Embedded Avionics Systems, ReCoSoC 2012, Pages 1-8.

[13] G. Bazydlo, M. Adamski and L. Stefanowicz: Translation UML diagrams into Verilog, In HSI 2014, Pages 267-271.

[14] M. Doligalski and M. Adamski: UML state machine implementation in FPGA devices by means of dual model and Verilog Translation UML diagrams into Verilog, 11th International Conference on Industrial Informatics (INDIN) 2013, Pages 177-184.

[15] W. Mueller, Da He, F. Mischkalla, A. Wegele, P. Whiston, P. Penil and E. Villar: The SATURN Approach to SysML-based HW/SW Codesign, IEEE Computer Society Symposium on VLSI 2010, Pages 506-511.

[16] S. H. M. Durand and V. Bonato: A tool to support Bluespec System Verilog coding based on UML diagrams, IEEE 38th Annual Conference on Industrial Electronics IECON 2012, Pages 4670-4675.

[17] S. K. Wood et al: A Model-Driven Development Approach to Mapping UML State Diagrams to Synthesizable VHDL, IEEE Tran. on Computers 2008, Volume 57, Issue 10, Pages 1357-1371.

[18] T. G. Moreira, M. A. Wehrmeister, C. E. Pereira, J. Petin and E.Levrat: Automatic Code Generation for Embedded Systems: From UML Specifications to VHDL Code, IEEE Conference on Industrial Informatics (INDIN) 2010, Pages 1085-1090.

[19] J. Vidal, F. Lamotte, G. Gogniat, P. Soulard and J. Diguet: A co-design approach for embedded system modeling and code generation with UML and MARTE, In DATE 2009, Pages 226-231.

[20] E. Linehan and S. Clarke: An aspect-oriented, model-driven approach to functional hardware verification, Journal of Systems Architecture, Elsevier 2012, Volume 58, Issue 5, Pages 195-208.

[21] F. Herrera et al.: The COMPLEX methodology for UML/MARTE Modeling and design space exploration of embedded systems, Journal of Systems Architecture, Elsevier 2014, Volume 60, Issue 1, Pages 55-78.

[22] S. Lecomte et al.: A co-design methodology based on model driven architecture for real time embedded systems, Mathematical and Computer Modelling, Elsevier 2011, Vol. 53, Issue 3-4, Pages 471-484.

[23] I. R. Quadri, A. Gamatie, P. Boulet, S. Meftali and J. Dekeyser: Expressing embedded systems configurations at high abstraction levels with UML MARTE profile: Advantages, limitations and alternatives, Journal of Systems Architecture, Elsevier 2012, Volume 58, Issue 5, Pages 178-194.

[24] D. Chen, L. Feng, T. N. Qureshi, H. Lonn and F. Hagl: An architectural approach to the analysis, verification and validation of software intensive embedded systems, Computing, SPRINGER 2013, Volume 95, Issue 8, Pages 649-688.

[25] Y. Vanderperren, W. Mueller and W. Dehaene: UML for electronic systems design: a comprehensive overview, Design Automation for Emb. Systems, SPRINGER 2008, Volume 12, Issue 4, Pages 261-292.

[26] M. Elhaji, P. Boulet, A. Zitouni, S. Meftali, J. Dekeyser and R. Tourki: System level modeling methodology of NoC design from UML-MARTE to VHDL, Design Automation for Emb. Systems, SPRINGER 2012, Volume 16, Issue 4, Pages 161-187.

[27] E. Riccobene and P. Scandurra: Integrating the SysML and the SystemC-UML profiles in a model-driven embedded system design flow, Design Automation for Embedded Systems, SPRINGER 2012, Volume 16, Issue 3, Pages 53-91.

[28] P. Penil, J. Medina, H. Posadas and E. Villar: Generating heterogeneous executable specifications in SystemC from UML/MARTE models, Innovations in Systems and Software Engineering, SPRINGER 2010, Volume 6, Issue 1-2, Pages 65-71.

[29] M. Z. Iqbal, A. Arcuri and Lionel Briand: Environment modeling and simulation for automated testing of soft real-time embedded software, Software and System Modeling, SPRINGER 2013.

[30] H. Espinoza, D. Cancila, B. Selic and S. Gerard: Challenges in Combining SysML and MARTE for Model-Based Design of Embedded Systems, SPRINGER 2009, Volume 5562, Page 98-113.

[31] L. Yin, J. Liu, Z. Ding ; Mallet, F., de Simone, R.: Schedulability Analysis with CCSL Specifications, 20th Asia-Pacific Software Engineering Conference (APSEC) 2013, Pages 414-421.

[32] F. Mallet: Automatic Generation of Observers from MARTE / CCSL, 23rd IEEE Symposium on Rapid System Prototyping 2012, Pages 86-92.

[33] Samir Chouali and Ahmed Hammad: Formal verification of components assembly based on SysML and interface automata, Innovations in Systems and Software Engineering, SPRINGER 2011, Volume 7, Issue 4, Pages 265-274.

[34] M. Rota Sena Marques, E. Siegert and L. Brisolara: Integrating UML, MARTE and SysML to improve requirements specification and traceability in the embedded domain, 12th IEEE International Conference on Industrial Informatics 2014, Pages 176-181.

[35] F. Mallet, J. Millo and R. de Simone: Safe CCSL specifications and Marked Graphs, 11th IEEE / ACM International Formal Methods and Models for Codesign 2013, Pages 157-166

[36] N. Jin, C. shen, Jun chen and T. Ni: Engineering of An Assertion-based PSLSimple-Verilog Dynamic Verifier by Alternating Automata, Electronics Notes in Theoretical Computer Science 2008, Volume 207, Pages 153-169.

[37] J. Greenyer and E. Kindler: Comparing relational model transformation technologies: implementing Query/View/Transformation with Triple Graph Grammars, Software & Systems Modeling, SPRINGER 2010, Volume 9, Issue 1, Pages 21-46.

[38] J. Won Ko, K. Chung and J. Han: Model transformation verification using similarity and graph comparison algorithm, Multimedia tools and applications, SPRINGER 2013.

[39] R. Gronmo, B. M. Pedersen and G. K. Olsen: Comparison of Three Model Transformation Languages, Model Driven Architecture – Foundation and applications, SPR. 2009, Vol. 5562, Pages 2-17.

[40] P. Feher and L. Lengyel: The challenges of a model transformation language, IEEE 19th International Conference and Workshops on Engineering of Computer Based Systems (ECBS) 2012, Pages 324-329.

[41] E. Guerra, J. de Lara, D. Kolovos and R. Paige: A Visual Specification Language for Model-to-Model Transformations, IEEE Symposium on Visual Languages and Human-Centric Computing 2010, Pages 119-126.

[42] L. M. Rose, N. Matragkas, Dimitrios S. Kolovos and R. F. Paige: A Feature Model for Model-to-Text Transformation Languages, ICSE Workshop on Modeling in Software Engineering 2012, Pages 57-63.

[43] A. Ledo, F. Ramalho and N. Melo: MetaTT – A Metamodel Based Approach for Writing Textual Transformations, 6th Brazilian Symposium on SCAR 2012, Pages 61-70.

[44] D. Doering: A Model Driven Engineering Methodology for Embedded System Designs - HIPAO2, 12th IEEE International Conference on Industrial Informatics (INDIN) 2014Pages 787-790.

[45] J. Lara and E. Guerra: Towards the flexible reuse of model transformations: A formal approach based on graph transformation, Journal of Logical and Algebraic Methods in Programming 2014, Volume 83, Issues 5-6, Pages 427-458.

[46] S. Kolahdouz-Rahimi, K. Lano, S. Pillay, J. Troya and P. Van Gorp: Evaluation of model transformation approaches for model refactoring, Science of Computer Programming 2014, Vol. 85, Part A, Pages 5-40.

[47] T. Launiainen, K. Heljanko and T. Junttila: Efficient Model Checking of PSL Safety Properties, 10th International Conference on Application of Concurrency to System Design (ACSD) 2010, Pages 95-104.

[48] M. Zhang, D. Gao and X. Fan: Formal Verification of Mixed-signal Circuits using Extended PSL, 8th Inte. Con. on ASIC 2009.

[49] B.N. Uchevler and Kjetil Svarstad: Assertion Based Verification Using PSL-like Properties In Haskell, 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems 2013, Pages 254-257.

[50] V. Andrea Bollati, J. M. Vara, A. Jimenez and E. Marcos: Applying MDE to the (semi-)automatic development of model transformations, Info. and Software Technology 2013, Vol. 55, Issue 4, Pages 699-718.

[51] Adrian Rutle, Alessandro Rossini, Yngve Lamo and Uwe Wolter: A formal approach to the specification and transformation of constraints in MDE, The Journal of Logic and Algebraic Programming 2012, Volume 81, Issue 4, Pages 422-457.

[52] Samir Ouchani, Otmane Ait Mohamed and Mourad Debbabi: A formal verification framework for SysML activity diagrams, Experts Systems with Applications 2014, Pages 2713-2728.

[53] L. Telinski W. anger, I. W. Soares and P. Cezar Stadzisz: A Brazilian survey on UML and model-driven practices for embedded software development, The Journal of sys. and software 2013, Pages 997-1005

[54] B. Alizadeh and Payman Behnam: Formal equivalence verification and debugging techniques with auto-correction mechanism for RTL designs, Microprocessors and Microsystems 2013, Pages 1108-1121

[55] D. Knorreck and L. Apvrille: TEPE: A SysML Language for Time-Constrained Property Modeling and Formal Verification, ACM SIGSOFT Software Engineering 2011, Vol. 36, Issue 1, Pages 1-8.

[56] N. Ge, M. Pantel and X. Cregut: Formal Specification and Verification of Task Time Constraints for Real-Time Systems, LNCS SPRINGER 2012, Volume 7610, Pages 143-157.

[57] R. Behjati, Tao Yue, Shiva Nejati, Lionel Briand and Bran Selic: Extending SysML with AADL Concepts for Comprehensive System Architecture Modeling, SPRINGER 2011, Vol. 6698, Pages 236-252.

[58] L. Baresi et al: Formal verification and validation of embedded systems: the UML-based MADES approach, Software and Systems Modeling SPRINGER 2013.

[59] X. Huang, Qi. Sun, J. Li, M. Pan and T. Zhang: An MDE-Based Approach to the Verification of SysML State Machine Diagram, Fourth Asia-Pacific Symposium on Internetware Software, ACM 2012.

[60] M. C. Nguyen, E. Jee, J. Choi and D. Bae: Automatic Construction of Timing Diagrams from UML/MARTE Models for Real-Time Embedded Software, Proceedings of the 29th ACM Symposium 2014.

[61] Erwan Bousse et al.: Proceedings of the Workshop on Model-Driven Engineering, Verification and Validation, ACM 2012, Pages 11-16.

[62] IEEE scientific database, http://ieeexplore.ieee.org/

[63] Elsevier, http://www.sciencedirect.com/

[64] B. Kitchenham, Procedures for Performing Systematic Reviews,TR/SE-0401/NICTA, Technical Report 0400011T, Keele University, 2004.

[65] Math works Simulink, www.mathworks.com/products/simulink/

[66] Springer, Last Accessed July 2014. http://link.springer.com/

[67] ACM, Last Accessed July 2014. http://dl.acm.org/

[68] MODEVES Project, SLR Search terms details http://www.modeves.com/searchterms.html

[69] MODEVES Project, SLR Quality Assessment details http://www.modeves.com/qualityassessment.html

[70] MODEVES Project, Overview of Selected Researches http://www.modeves.com/overview%20of%20researches.html

[71] Topcased Tool, Last Accessed August 2014. http://www.topcased.org/

[72] Modelio, "Open source UML Editor and MDE Workbench," 2012, www.modelio.org.

[73] Magic Draw, http://www.nomagic.com/products/magicdraw.html

[74] IBM, "Eclipse Modeling Framework," http://www.eclipse.org/gef/

[75] IBM Rhapsody Last Accessed September 2014. http://www-03.ibm.com/software/products/pk/en/ratirhapfami/

[76] Papyrus MDT, http://www.eclipse.org/modeling/mdt/papyrus/

[77] Sparx Systems Enterprise Architecture , http://www.sparxsystems.com/resources/mdg_tech/

[78] Visual Paradigm, http://www.visual-paradigm.com

[79] QVT Project Last Accessed October 2014. http://projects.ikv.de/qvt

[80] Eclipse ATL, Last Accessed August 2014. www.eclipse.org/atl/

[81] A. Bunker, G. Gopalakrishnan, S.A. Mckee, Formal hardware specification languages for protocol compliance verification: ACM Trans. Des. Autom. Electron. Syst. 9 (1) (2004) 1–32.

[82] Eclipse Acceleo, https://wiki.eclipse.org/Acceleo/

[83] S. Wood, D. Akehurst, O. Uzenkov, W. Howells, K. McDonald-Maier: "A model-driven development approach to mapping UML state diagrams to synthesizable VHDL", Computers, IEEE Transactions on, Vol. 57, No. 10, 2008, pp. 1357–1371.

[84] Apache Software Foundation. (2010, Nov.) The Apache Velocity Project. [Online]. Available: http://velocity.apache.org/

[85] K.Lano, S.Kolahdouz-Rahimi, TheUML RSDStoolset , http://www.dcs.kcl.ac.uk/staff/kcl/uml2web,2012.

[86] M. A. Wehrmeister, E. P. Freitas, C. E. Pereira, and F. Ramming, F., "GenERTiCA: A Tool for Code Generation and Aspects Weaving". 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC), Orlando, 2008.

[87] "MDWorkbench platform," www.mdworkbench.com, from SODIUS.

[88] Catapult, www.calypto.com/en/products/catapult/overview/

[89] OMG, QVT Language, 2007. Available from: <http://tiny.cc/tFGGx>.

[90] G. Taentzer, AGG: A Graph Transformation Environment for Modeling and Validation of Software. Springer, 2004, vol. 3062, pp. 446–453.

[91] Eclipse MOFSCRIPT http://eclipse.org/gmt/mofscript/

[92] Eclipse Epsilon http://www.eclipse.org/epsilon/

[93] TMC, Last Accessed October, 2014 http://se.kaist.ac.kr/nmchau/tmc/

[94] SYSML to B tool. Last Accessed October 2014. http://www.irisa.fr/triskell/Software/protos/sysml2b/index_html

[95] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of Probabilistic Real-Time Systems," In 23rd International Conference on Computer Aided Verification (CAV) 2011, pp. 585–591.

[96] UPPAAL tool Last Accessed October 2014. http://www.uppaal.com/

[97] Active HDL, Last Accessed, October, 2014. https://www.aldec.com/en/products/fpga_simulation/active-hdl

[98] Qemu, Last Accessed October 2014. www.qemu.org

[99] Tony Bybell. Gtkwave. http://home.nc.rr.com/gtkwave/index.html

[100] Artisan Studio, http://www.atego.com/products/atego-modeler/

[101] Zot tool, http://home.deib.polimi.it/pradella/Zot/

[102] GASPARD SoC Framework, 2010, <http://www.gaspard2.org/>.

[103] Ttool Last Accessed, October 2014. http://ttool.telecom-paristech.fr/