Chapter 2: Tools Selection

1. MBSE Tools Investigation

Once the researches are classified into different categories, the next step is to identify the tools and frameworks used in selected researches to perform various MBSE activities. It is important to mention here that a tool is used to perform a specific MBSE activity whereas a framework is a complete environment supporting set of tools that can be used to perform various MBSE activities. On the basis of literature review, 39 preliminary MBSE tools have been identified as given in Table VI.

Table I: Preliminary tools selection

Sr. #	Name of Tool / Framework	Corresponding MBSE Activities	Relevant Researches
1	Topcased [62]	Modeling	[3][8][9][52]
2	Modelio Editor [68]	Modeling	[12]
3	Magic Draw [131]	Modeling	[6][16][20]
4	Eclipse GEF [89]	Modeling	[17]
5	Rhapsody [132]	Modeling	[4][19][22][57]
6	PapyrusMDT [98]	Modeling	[21][44][60]
7	EA MDG [85]	Modeling	[27]
8	Visual paradigm [126]	Modeling	[60]
9	MediniQVT [121]	Model Transformation	[2]
10	ATL [63]	Model Transformation	[3][8][39][44][46][50][59]
11	Xpand [97]	Model Transformation	[8][20]
12	Acceleo [67]	Model Transformation	[9][21][42][44]
13	MODCO [83]	Model Transformation	[13]
14	MODEASY [17]	Model Transformation	[17]
15	Apache Velocity [86]	Model Transformation	[16][18]
16	Eclipse EMF	Model	[21][41][37]

	[115]	Transformation	
17	GenERTiCA tool [94]	Model Transformation	[18]
18	MDworkbench [95]	Model Transformation	[19][22]
19	CatapultC [116]	Model Transformation	[22]
20	QVTO [104]	Model Transformation	[14]
21	AGG [109]	Model Transformation	[38][39][40]
22	MOFScript [112]	Model Transformation	[29][39][42][43]
23	MeTHAGeM [50]	Model Transformation	[50]
24	Epsilon [133]	Model Transformation	[12][58]
25	TMC [124]	Model Transformation	[60]
26	SYSML to B translator tool [125]	Model Transformation	[61]
27	PRISM [66]	Model Verification	[7][52]
28	UPPAAL [123]	Model Verification	[55][59]
29	SIMULINK [65]	Simulation	[4][25]
30	ActiveHDL [129]	Simulation	[14]
31	QEMU [130]	Simulation	[15]
32	GTKWave [72]	Simulation	[36]
33	radCASE [1]	• Modeling • Model Transformation	[1]
34	Artisan Studio [114]	• Modeling • Model Transformation	[15]
35	Verilog Dynamic Verifier [36]	• Model Verification • Model Transformation	[36]
36	radCHECK [1]	• Model Verification • Simulation	[1]

37	Zot [118]	• Model Verification • Simulation	[12][58]
38	Gaspard2 [102]	• Modeling • Model Transformation • Simulation	[23][26]
39	TTool [117]	• Modeling • Model Verification • Simulation	[55]

The *Corresponding MBSE activities* column of the Table VI identifies the MBSE activities for which tool has been used in selected researches. Few selected researches (e.g. [1]) develop their own tool to perform particular MBSE activities. Such tools (e.g. radCASE[1], radCHECK[1]) are also identified and listed in the Table VI. We also identified that few selected researches use similar tool to perform particular MBSE activity so such researches are given in *Relevant Researches* column of the Table VI against each tool. It has been analyzed that few selected researches do not provide sufficient information about the tools used in their work. For example, Elvinia et.al [27] explains that they perform both M2M and M2T transformations. They also describe that EA MDG [85] tool is used for modeling and integrated into Eclipse technology to perform both transformations. However, it is unclear which specific tool / tools are used to perform both transformations because there is variety of transformation tools supported by Eclipse technology. Similarly, Carlos et al. [10] propose multi view power modeling approach by using UML and its SYSML/MARTE profiles but do not provide any information about the modeling tool used in their work. This limitation is present in very few selected researches so it does not affect the ultimate results of our research objective.

The identification of given tools (Table VI) is not sufficient to select appropriate tool for our project. Moreover, the classification of tools is based on their practical utilization in selected research works. Consequently, there is a possibility that the selected tool can be customized to perform multiple MBSE activities, not discussed in the selected research works. Furthermore, it is also required to search for additional MBSE tools those have been missed during the literature review. Therefore, we further evaluate the selected tools to provide a rational representation. First, initial evaluation is performed and then various important characteristics of the tools are used for evaluation. The details are given in subsequent sections:

1.1 Initial Tools Evaluation

In this phase, we perform initial evaluation of selected tools (Table VI) to check the availability of sufficient information required for detailed tools evaluation. This includes web address, license type, working environment and help documentation. Furthermore, we also search for important tools used in the industries and missed during the SLR. Consequently, it has been observed that the information available for some tools is not adequate to perform further evaluation. Therefore, we exclude such tools for further evaluation.

1.2 Tools Characteristics

Once the initial evaluation is done, further evaluation is performed by defining some tools characteristics. These characteristics help us to select appropriate MBSE tools according to the requirements of our project. Two types of tools characteristics used in this article are: general characteristics and activity specific characteristics. General characteristics are independent of a particular MBSE activity (Section 4.2.1). On the other hand, activity specific characteristics are related to the corresponding MBSE activity such as modeling, transformation, verification and validation (Section 4.2.2).

1.2.1 General Characteristics

These characteristics are applicable to all the tools:

License Type: It is the primary characteristic frequently considered for the selection of appropriate tool. We investigate common license types like General Public License (GPL) [141], Eclipse Public License (EPL) [142], Apache Software License (ASL) [143] and CeCILL [144] supported by various MBSE tools. Furthermore, we also provide the name of the vendors in case of proprietary license type.

Customizability: This characteristic can be defined as "The ability of a tool to support desired alterations in order to perform a particular MBSE activity". For example, Open source tools can easily be customized [117-118] while it is difficult to customized proprietary tools due to unavailability of the source code. We define three parameters to explain customizability of a tool: "No" means tool is not customizable. *"Partial"* means tool is able to support minor alterations. *"Full"* means tool is able to support major alterations according to given requirements. For example, if the tool supports minor necessary alterations required for its seamless integration in a given working environment, it will be considered as partially customizable. On the other hand, if the tool also allows modifications to its major basic functions (e.g. transformation type etc), it will be considered as fully customizable.

Supported Platforms: This characteristic defines the type of operating system supported by the tool. Windows and Linux platforms are more frequently used.

1.2.2 Activity Specific Characteristics

These characteristics are specific to a particular MBSE activity:

1.2.2.1 Modeling Characteristics

Modeling characteristics are related to the modeling activity in MBSE. The defined parameters in this category are:

Support of XMI Format: Import / Export to XML Metadata Interchange (XMI) is an important characteristic of modeling tools and frequently used to perform various data exchange activities. We will evaluate whether a particular modeling tool support this functionality.

Profile Support: From the SLR, it is clear that UML, SYSML and MARTE profiles are frequently used to model embedded systems requirements. We will investigate the modeling tools for potential UML, SYSML and MARTE profiles support.

1.2.2.2 Model Transformation Characteristics

The only defined parameter in the model transformation category is the type of transformation.

Transformation Type: There are two important transformation approaches i.e. Model-to-Model (M2M) and Model-to-Text (M2T). We will investigate the transformation tools for the supported transformation type. i.e. M2M, M2T or Both

1.2.2.3 Model Verification Characteristics

The only defined parameter in the model verification category is the verification approach:

Verification approach: This characteristic describes verification approach supported by the verification tool. The model verification is usually performed through formal verification methods. We will investigate the verification tool by considering its formal verification technique used to perform model verification e.g. probabilistic automata, Finite State Machine (FSM) etc.

1.2.2.4 Simulation Characteristics

From the SLR, it is observed that most of research works integrate available simulation tools to perform the required simulation. Therefore, the most important characteristic of simulation tool is its integration capability as given below:

Integration Capability: This characteristic defines the ability of a simulation tool to support its seamless integration in a given working environment. We define three parameters to explain integration capability of a tool. "Low" means tool provides low integration capability. *"Medium"* means tool provides good integration support. *"High"* means tool provides high integration features for its potential integration in a given working environment.

1.3 Tools Evaluation Outcomes

We thoroughly analyze identified tools (Table VI) according to pre-defined characteristics (Section 4.2). Furthermore, we also search for any important tools those have been missed during the SLR but frequently used in research / industrial practices [134-137]. On the basis of detailed tools evaluation, we categorize the tools into four MBSE activities (i.e. modeling, model transformation, model verification and simulation).

We analyze six tools those are frequently used to model embedded systems requirements through UML/SYSML/MARTE notations as given in Table VII

Table II: Modeling tools used in MBSE for embedded systems

Sr. #	Tool Name	General Characteristics			Activity Specific Characteristics	
		License Type	**Customizability**	**Supported Platforms**	**XMI Support**	**Profiles Support**
1	Magic Draw [131]	Proprietary (No Magic)	Partial	Windows Linux	Yes	UML / SYSML / MARTE
2	Modelio Editor [68]	GPL	Full	Windows Linux	Yes	UML / SYSML / MARTE
3	Rhapsody [132]	Proprietary (IBM)	Partial	Windows Linux	Yes	UML / SYSML / MARTE
4	PapyrusMDT [98]	EPL	Full	Windows Linux	Yes	UML / SYSML / MARTE
5	Artisan Studio [114]	Proprietary (Atego)	No	Windows	Yes	UML / SYSML / MARTE
6	Visual paradigm [126]	Proprietary (Visual paradigm)	No	Windows Linux	Yes	UML / SYSML

The tools presented in Table VII are mainly used to perform model development activities and support different UML profiles. However, some modeling tools also provide partial support for other MBSE activities as well. For example, Modelio [68] and Rhapsody [132] provide partial facility to generate source code (which is a kind of model transformation) but it is inadequate for the development of complex and large embedded systems.

We identify twelve model transformation tools as given in Table VIII.

Table III: Tools for model transformation in MBSE for embedded systems

Sr. #	Tool Name	General Characteristics			Activity Specific Characteristics
		License Type	**Customizability**	**Supported Platforms**	**Transformation Type**
1	MediniQVT [121]	EPL	Full	Windows Linux Mac	M2M
2	ATL [63]	EPL	Full	Windows Linux	M2M
3	Xpand [97]	EPL	Full	Windows Linux	M2T
4	Acceleo [67]	EPL	Full	Windows Linux	M2T
5	Apache	ASL	Full	Windows	M2T

	Velocity [86]			Linux	
6	MDworkbench [95]	Proprietary (Sodius)	Partial	Windows Linux	Both
7	QVTO [104]	EPL	Full	Windows Linux	M2M
8	MOFScript [112]	EPL	Full	Windows Linux	M2T
9	Epsilon [133]	EPL	Full	Windows Linux	Both
10	JET [134]	EPL	Full	Windows Linux	M2T
11	Henshin [135]	EPL	Full	Windows Linux	Both
12	Viatra2 [136]	EPL	Full	Windows Linux	Both

Variety of tools are available to perform either model-to-model or model-to-text transformations as given in Table VIII. Some tools are capable of performing both M2M and M2T transformations. These tools are highly customizable to perform the desired model transformations. These tools can easily be integrated into given working environment due to their high customizability.

We identified two model verification tools as given in Table IX

Table IV: Model verification tools

Sr. #	**Tool Name**	**General Characteristics**			**Activity Specific Characteristics**
		License Type	**Customizability**	**Supported Platforms**	**Verification Approach**
1	PRISM [66]	GPL	Full	Windows Linux	DTMC CTMC MDP PA PTA
2	Zot [118]	GPL	Full	Windows Linux	TRIO Metric Temporal Logic

The two general-purpose model verification tools have been identified as given in Table IX. These tools are frequently used in various research / industrial practices due to their high level of customizability. PRISM [66] is capable of analyzing models based on Discrete-Time Markov Chains (DTMCs), Markov Decision Processes (MDPs), Continuous-Time Markov Chains (CTMCs), Probabilistic Timed Automata (PTAs) and Probabilistic Automata (PAs). The Zot tool is customized by Quadri et al. [12] to even support partial model simulation features for embedded systems. The presented verification tools can easily be tailored according to given model verification requirements. On the other hand, the various other verification tools are also available but these are inflexible due to their partial customization support and limitations. For example, UPPAAL [123] is timed automata based tool that provide verification support but it is bond with its own Graphical editor and modeling methodology.

We identified four simulation tools as given in the Table X

Table V: Simulation tools

Sr. #	Tool Name	General Characteristics			Activity Specific Characteristics
		License Type	**Customizability**	**Supported Platforms**	**Integration Capability**
1	SIMULINK [65]	Proprietary (Math Works)	Partial	Windows Linux	High
2	ActiveHDL [129]	Proprietary (ALDEC)	Partial	Windows	Low
3	QEMU [130]	GPL	Full	Windows Linux	High
4	Catapult C [116]	Proprietary (Calypto)	Partial	Windows Linux	Low

The selection of an appropriate simulation tool is always challenging. The simulation tools are highly bonded with their own model development environment. Moreover, a seamless integration of simulation tools in a particular working environment is very difficult. Furthermore, full feature support for different hardware description languages within a single simulation tool is not possible. For example, none of the four simulation tools (Table X) fully support System Verilog Assertions. Mentor Graphics Questa [137] support SVA but it lacks certain other features. Therefore, selection of appropriate simulation tool is totally dependent on the specific simulation requirements.

We have also identified frequently used toolsets and frameworks to perform various MBSE activities. A toolset is a collection of tools used to perform various MBSE activities in particular environment whereas framework is the complete environment that support variety of MBSE tools / toolsets work together to perform wide-ranging MBSE activities. The identified toolsets and frameworks are given in Table XI

Table VI: Frameworks and toolsets to perform various MBSE activities for embedded systems

Sr. #	Name	License Type	Supported Platforms	Activity Specific characteristics
				Type
1	EMF	EPL	Windows	Framework

	[115]		Linux	
2	Topcased [62]	EPL	Windows Linux	Toolset
3	Gaspard2 [102]	GPL	Windows Linux	Toolset
4	TTool [117]	CeCIll	Windows Linux	Toolset

Framework: Eclipse Modeling Framework (EMF) is a leading framework supporting various tools and techniques for the development of wide-ranging MDE (Model Driven Engineering) applications. There are varieties of modeling, model transformation and verification tools those are built on top of EMF (e.g. Topcased, Gaspard2 etc).

Toolset: Topcased is a complete toolset for the development of embedded systems through MBSE approach. It is built on top of Eclipse Modeling Framework. Topcased is currently migrating to PolarSys [145] (open source tools for embedded systems). Gaspard2 provides a complete toolset to support MBSE modeling, model transformation, verification and validation activities for the development of embedded systems. It is also build on Eclipse framework. TTool is a toolset that support verification and simulation of embedded systems, developed in particular UML and SYSML diagrams. It supports TURTLE, DIPLODOCUS and AVATAR profiles [117]. The major drawback of TTool is its inflexibility as it is mandatory to develop models only in supported UML and SYSML diagrams / profiles.

2. Tools selection for MODEVES

The selection of appropriate MBSE tools is always challenging due to the wide-ranging behavioral / temporal aspects of embedded systems that require sophisticated tools to perform modeling, model transformation, verification and simulation activities. Moreover, a seamless integration of different tools within a single platform is another problem. It is always difficult to find a complete solution for verification and simulation of models due to the complexity of embedded systems. We present a comprehensive analysis and identification of MBSE tools (Section 4) to perform modeling, model transformation, verification and simulation activities. On the basis of tools evaluation and identification (Section 4), we select following frameworks and tools for MODEVES:

1. **Framework:** From literature review(Chapter 1) and tools selection (Section 4), it has been analyzed that Eclipse platform [147] is playing the leading role by providing various frameworks (e.g. EMF, GMF etc) and tools to support different MBSE activities for embedded systems. Various tools are built on top of the Eclipse platform and available as eclipse plug-in (e.g. papyrus, Topcased etc) Further, Eclipse-based frameworks and tools are highly customizable and can be altered according to diverse MBSE requirements.

Therefore, we use eclipse platform and Eclipse Modeling Framework (EMF) in our project.

2. **Modeling Tool:** As we already analyzed (Section 3.4) that our modeling methodology will be based on UML and its SYSML/MARTE profiles, we select modeling tool that support all UML and its SYSML/MARTE profiles. Another project requirement is the seamless integration of modeling tool in the given project environment. Consequently, modeling tool should be customizable (Section 4.2.1). *We will select any one of papyrus and magic draw modeling tools to be used in our project.*
3. **Model Transformation Tool**: We identify the renowned model transformation tools (Table VIII). *We will select any one of ATL, Acceleo, MOFScript and Epsilon transformation tools as these are the most appropriate tools to meet transformation requirements of the project.*
4. **Model Verification Tool:** As we use Assertion Based Verification (ABV) approach in our project, any model verification tool based on formal method (e.g. Finite automata etc) is optional for our project.
5. **Simulation Tool:** We identify four simulation tools (Table X) frequently used in the research practices. However, it is analyzed that Assertion Based Verification (ABV) (using System Verilog Assertions) is rarely discussed in the contemporary research literature. Therefore, none of the identified simulation tools (Table X) support System Verilog Assertions. We further investigate the industrial practices to identify the simulation tools those fully support SystemVerilog Assertions. Consequently, we select Mantor Grraphics Questa Sim to be used in our project.

References

[1] Giuseppe Di Guglielmo, Luigi Di Guglielmo, Andreas Foltinek,, Masahiro Fujita, Franco Fummi, Cristina Marconcini and Graziano Pravadelli: **On the integration of model-driven design and dynamic assertion-based verification for embedded software**, Journal of Systems and Software, Volume 86, Issue 8, August 2013, Pages 2013–2033. DOI: 10.1016/j.jss.2012.08.061

[2] George-Dimitrios Kapos, Vassilis Dalakas, Anargyros Tsadimas, Mara Nikolaidou and Dimosthenis Anagnostopoulos: **Model-based system engineering using SysML: Deriving executable simulation models with QVT**, 8th Annual IEEE Systems Conference (SysCon) 2014, Pages 531-538. DOI: 10.1109/SysCon.2014.6819307

[3] Samir Berrani, Ahmed Hammad, Hassan Mountassir: **Mapping SysML to Modelica to Validate Wireless Sensor Networks Non-Functional Requirements**, 11th International Symposium on Programming and Systems (ISPS) 2013, Pages 177-186. DOI: 10.1109/ISPS.2013.6581484

[4] Takashi Sakairi, Eldad Palachi, Chaim Cohen, Yoichi Hatsutori, Junya Shimizu, and Hisashi Miyashita: **Designing a control system using SysML and Simulink**, Proceedings of SICE Annual Conference 2012, Pages 2011-2017.

[5] Stancescu,S. Neagoe, L. ; Marinescu, R. ; Enoiu, E.P.: **A SysML model for code correction and detection systems**, Proceedings of 33rd International Convention MIPRO 2010, Pages 189-191.

[6] Anargyros Tsadimas, George-Dimitrios Kapos, Vassilis Dalakas, Mara Nikolaidou and Dimosthenis Anagnostopoulos: **Integrating simulation capabilities into SysML for enterprise information system design**, 9th International Conference on System of Systems Engineering (SOSE) 2014, Pages 272-277. DOI: 10.1109/SYSOSE.2014.6892500

[7] Samir Ouchani, Otmane Aıt Mohamed and Mourad Debbabi: **A Formal Verification Framework for BlueSpec System Verilog**, Forum on Specification and Design Languages (FDL) 2013, Pages 1-7.

[8] Fabrice Bouquet, Jean-Marie Gauthier, Ahmed Hammad and Fabien 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, DOI: 10.1109/dMEMS.2012.12

[9] Gianmaria DeTommasi, Riccardo Vitelli, Luca Boncagni, and André C. Neto: **Modeling of MARTe-Based Real-Time Applications With SysML**, IEEE Transactions on Industrial Informatics 2013, Volume 9, Issue 4, Pages 2407-2415, DOI: 10.1109/TII.2012.2235073

[10] Carlos Gomez, Julien DeAntoni and Frederic Mallet: **Multi-View Power Modeling based on UML MARTE and SysML**, 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA) 2012, Pages 17-20, DOI: 10.1109/SEAA.2012.66

[11] Ermeson Andrade, Paulo Maciel, Gustavo Callou and Bruno Nogueira: **A Methodology for Mapping SysML Activity Diagram to Time Petri Net for Requirement Validation of Embedded Real-Time Systems with Energy Constraints**, Third International Conference on Digital Society ICDS 2009, Pages 266-271, DOI: 10.1109/ICDS.2009.19

[12] Imran R. Quadri, Etienne Brosse, Ian Gray, Nicholas Matragkas, Leandro Soares Indrusiak, Matteo Rossi, Alessandra Bagnato and Andrey Sadovykh: **MADES FP7 EU Project: Effective High Level SysML/MARTE Methodology for Real-Time and Embedded Avionics Systems**, 7th International Workshop Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC) 2012, Pages 1-8, DOI: 10.1109/ReCoSoC.2012.6322882

[13] Grzegorz Bazydlo, Marian Adamski and Lukasz Stefanowicz: **Translation UML diagrams into Verilog**, 7th International Conference on Human System Interactions (HSI) 2014, Pages 267-271, DOI: 10.1109/HSI.2014.6860487

[14] Michal Doligalski and Marian 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, DOI: 10.1109/INDIN.2013.6622878

[15] Wolfgang Mueller, Da He, Fabian Mischkalla Arthur Wegele, Paul Whiston, Pablo Penil and Eugenio Villar: **The SATURN Approach to SysML-based HW/SW Codesign**, IEEE Computer Society Annual Symposium on VLSI (ISVLSI) 2010, Pages 506-511, DOI: 10.1109/ISVLSI.2010.95

[16] Sergio H. M. Durand and Vanderlei 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, DOI: 10.1109/IECON.2012.6389493

[17] Stephen K. Wood, David H. Akehurst, Oleg Uzenkov, W. Gareth J. Howells, and Klaus D. McDonald-Maier: **A Model-Driven Development Approach to Mapping UML State Diagrams to Synthesizable VHDL**, IEEE Transactions on Computers 2008, Volume 57, Issue 10, Pages 1357-1371, DOI: 10.1109/TC.2008.123

[18] Tomas G. Moreira, Marco A. Wehrmeister, Carlos E. Pereira, Jean-Francois Petin and Eric Levrat: **Automatic Code Generation for Embedded Systems: From UML Specifications to VHDL Code**, IEEE 8th International Conference on Industrial Informatics (INDIN) 2010, Pages 1085-1090, DOI: 10.1109/INDIN.2010.5549590

[19] Jorgiano Vidal, Florent de Lamotte, Guy Gogniat, Philippe Soulard and Jean-Philippe Diguet: **A co-design approach for embedded system modeling and code generation with UML and MARTE**, Conference and Exhibition Design Automation and Test in Europe (DATE) 2009, Pages 226-231, DOI: 10.1109/DATE.2009.5090662

[20] Eamonn Linehan and Siobhan Clarke: **An aspect-oriented, model-driven approach to functional hardware verification**, Journal of Systems Architecture, Elsevier 2012, Volume 58, Issue 5, Pages 195-208, DOI: 10.1016/j.sysarc.2011.02.001

[21] Fernando Herrera , Hector Posadas, Pablo Penil, Eugenio Villar, Francisco Ferrero, Raul Valencia and Gianluca Palermo: **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, DOI: 10.1016/j.sysarc.2013.10.003

[22] Stephane Lecomte, Samuel Guillouard, Christophe Moy, Pierre Leray and Philippe Soulard : **A co-design methodology based on model driven architecture for real time embedded systems**, Mathematical and Computer Modelling, Elsevier 2011, Volume 53, Issue 3-4, Pages 471-484, DOI: 10.1016/j.mcm.2010.03.035

[23] Imran Rafiq Quadri, Abdoulaye Gamatie, Pierre Boulet, Samy Meftali and Jean-Luc 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, DOI: 10.1016/j.sysarc.2012.01.001

[24] DeJiu Chen, Lei Feng, Tahir Naseer Qureshi, Henrik Lonn and Frank 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, DOI: 10.1007/s00607-013-0314-4

[25] Yves Vanderperren, Wolfgang Mueller and Wim Dehaene: **UML for electronic systems design: a comprehensive overview**, Design Automation for Embedded Systems, SPRINGER 2008, Volume 12, Issue 4, Pages 261-292, DOI: 10.1007/s10617-008-9028-9

[26] Majdi Elhaji, Pierre Boulet, Abdelkrim Zitouni, Samy Meftali, Jean-Luc Dekeyser and Rached Tourki: **System level modeling methodology of NoC design from UML-MARTE to VHDL**, Design Automation for Embedded Systems, SPRINGER 2012, Volume 16, Issue 4, Pages 161-187, DOI: 10.1007/s10617-012-9101-2

[27] Elvinia Riccobene and Patrizia 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, DOI: 10.1007/s10617-012-9097-7

[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, DOI: 10.1007/s11334-009-0105-4

[29] Muhammad Zohaib Iqbal, Andrea Arcuri and Lionel Briand: **Environment modeling and simulation for automated testing of soft real-time embedded software**, Software and System Modeling, SPRINGER 2013, DOI: 10.1007/s10270-013-0328-6

[30] Huascar Espinoza, Daniela Cancila, Bran Selic and Sebastien Gerard: **Challenges in Combining SysML and MARTE for Model-Based Design of Embedded Systems**, LNCS SPRINGER 2009, Volume 5562, Page 98-113 DOI: 10.1007/978-3-642-02674-4_8

[31] Ling Yin, Jing Liu, Zuohua Ding ; Mallet, F., de Simone, R.: **Schedulability Analysis with CCSL Specifications**, 20th Asia-Pacific Software Engineering Conference (APSEC) 2013, Pages 414-421 DOI: 10.1109/APSEC.2013.62

[32] Frederic Mallet: **Automatic Generation of Observers from MARTE / CCSL**, 23rd IEEE Symposium on Rapid System Prototyping (RSP) 2012, Pages 86-92

DOI: 10.1109/RSP.2012.6380695

[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 DOI: 10.1007/s11334-011-0170-3

[34] Milena Rota Sena Marques, Eliane Siegert and Lisane Brisolara: **Integrating UML, MARTE and SysML to improve requirements specification and traceability in the embedded domain**, 12th IEEE International Conference on Industrial Informatics (INDIN) 2014, Pages 176-181 DOI: 10.1109/INDIN.2014.6945504

[35] Frederic Mallet, Jean-Vivien Millo and Robert de Simone: **Safe CCSL specifications and Marked Graphs**, 11th IEEE / ACM International Formal Methods and Models for Codesign (MEMOCODE) 2013, Pages 157-166

[36] Naiyong Jin, Chengjie shen, Jun chen and Taoyong Ni: **Engineering of An Assertion-based PSL^{Simple}-Verilog Dynamic Verifier by Alternating Automata**, Electronics Notes in Theoretical Computer Science 2008, Volume 207, Pages 153-169 DOI: 10.1016/j.entcs.2008.03.091

[37] Joel Greenyer and Ekkart 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 DOI: 10.1007/s10270-009-0121-8

[38] Jong-Won Ko, Kyung-Yong Chung and Jung-Soo Han: **Model transformation verification using similarity and graph comparison algorithm**, Multimedia tools and applications, SPRINGER 2013, DOI: 10.1007/s11042-013-1581-y

[39] Roy Gronmo, Birger Moller-Pedersen and Goran K. Olsen: **Comparison of Three Model Transformation Languages**, Model Driven Architecture – Foundation and applications, SPRINGER 2009, Volume 5562, Pages 2-17 DOI: 10.1007/978-3-642-02674-4_2

[40] Peter Feher and Laszlo 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 DOI: 10.1109/ecbs.2012.6487443

[41] Esther Guerra, Juan de Lara, Dimitris Kolovos and Richard Paige: **A Visual Specification Language for Model-to-Model Transformations**, IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) 2010, Pages 119-126 DOI: 10.1109/VLHCC.2010.25

[42] Louis M. Rose, Nicholas Matragkas, Dimitrios S. Kolovos and Richard F. Paige: **A Feature Model for Model-to-Text Transformation Languages**, ICSE Workshop on Modeling in Software Engineering (MISE) 2012, Pages 57-63 DOI: 10.1109/MISE.2012.6226015

[43] Anderson Ledo, Franklin Ramalho and Nata Melo: **MetaTT – A Metamodel Based Approach for Writing Textual Transformations**, 6th Brazilian Symposium on Software Components Architectures and Reuse (SBCARS) 2012, Pages 61-70 DOI: 10.1109/SBCARS.2012.21

[44] Dionisio Doering: **A Model Driven Engineering Methodology for Embedded System Designs - HIPAO2**, 12th IEEE International Conference on Industrial Informatics (INDIN) 2014Pages 787-790 DOI: 10.1109/INDIN.2014.6945614

[45] Juan de Lara and Esther 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 DOI: 10.1016/j.jlamp.2014.08.005

[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, Volume 85, Part A, Pages 5-40 DOI: 10.1016/j.scico.2013.07.013

[47] Tuomas Launiainen, Keijo Heljanko and Tommi Junttila: **Efficient Model Checking of PSL Safety Properties**, 10th International Conference on Application of Concurrency to System Design (ACSD) 2010, Pages 95-104 DOI: 10.1109/ACSD.2010.27

[48] Meng Zhang, Deyuan Gao and Xiaoya Fan: **Formal Verification of Mixed-signal Circuits using Extended PSL**, 8th International Conference on ASIC 2009, Pages 1288-1293 DOI: 10.1109/ASICON.2009.5351231

[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 (DDECS) 2013, Pages 254-257 DOI: 10.1109/DDECS.2013.6549828

[50] Veronica Andrea Bollati, Juan Manuel Vara, Alvaro Jimenez and Esperanza Marcos: **Applying MDE to the (semi-)automatic development of model transformations**, Information and Software Technology 2013, Volume 55, Issue 4, Pages 699-718 DOI: 10.1016/j.infsof.2012.11.004

[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 DOI: 10.1016/j.jlap.2012.03.006

[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] Luciane Telinski Wiedermann anger, Inali Wisniewski Soares and Paulo Cezar Stadzisz: **A Brazilian survey on UML and model-driven practices for embedded software development**, The Journal of systems and software 2013, Pages 997-1005

[54] Bijan 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] Daniel Knorreck and Ludovic Apvrille: **TEPE: A SysML Language for Time-Constrained Property Modeling and Formal Verification**, ACM SIGSOFT Software Engineering Notes 2011, Volume 36, Issue 1, Pages 1-8 DOI: 10.1145/1921532.1921556

[56] Ning Ge, Marc Pantel and Xavier Cregut: **Formal Specification and Verification of Task Time Constraints for Real-Time Systems**, LNCS SPRINGER 2012, Volume 7610, Pages 143-157 DOI: 10.1007/978-3-642-34032-1_16

[57] Razieh Behjati, Tao Yue, Shiva Nejati, Lionel Briand and Bran Selic: **Extending SysML with AADL Concepts for Comprehensive System Architecture Modeling**, LNCS SPRINGER 2011, Volume 6698, Pages 236-252 DOI: 10.1007/978-3-642-21470-7_17

[58] Luciano Baresi, Gundula Blohm, Dimitrios S. Kolovos, Nicholas Matragkas, Alfredo Motta, Richard F. Paige, Alek Radjenovic and Matteo Rossi: **Formal verification and validation of embedded systems: the UML-based MADES approach**, Software and Systems Modeling SPRINGER 2013, DOI: 10.1007/s10270-013-0330-z

[59] Xiaopu Huang, Qingqing Sun, Jiangwei Li, Minxue Pan and Tian Zhang: **An MDE-Based Approach to the Verification of SysML State Machine Diagram**, Proceedings of the Fourth Asia-Pacific Symposium on Internetware Software, Article 9, ACM 2012, DOI: 10.1145/2430475.2430484

[60] Minh Chau Nguyen, Eunkyoung Jee, Jinho Choi and Doo-Hwan Bae: **Automatic Construction of Timing Diagrams from UML/MARTE Models for Real-Time Embedded Software**, Proceedings of the 29th Annual ACM Symposium 2014, Pages 1140-1145, DOI: 10.1145/2554850.2555011

[61] Erwan Bousse, David Mentre, Benoit Combemale, Benoit Baudry and Takaya Katsuragi: **Aligning SysML with the B Method to Provide V&V for Systems Engineering**, Proceedings of the Workshop on Model-Driven Engineering, Verification and Validation, ACM 2012, Pages 11-16, DOI: 10.1145/2427376.2427379

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

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

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

[65] Math works Simulink, Last Accessed August 2014. www.mathworks.com/products/**simulink**/

[66] M. Kwiatkowska, G. Norman, and D. Parker, “PRISM 4.0: Verification of Probabilistic Real-Time Systems,” in CAV, ser. LNCS. Springer, 2011, pp. 585–591

[67] Eclipse Acceleo, Last Accessed August 2014. https://wiki.eclipse.org/**Acceleo**/

[68] Modelio, “Open source UML Editor and MDE Workbench,” 2012, www.modelio.org.

[69] Ian Gray and Neil C. Audsley, “Developing Predictable Real-Time Embedded Systems Using AnvilJ,” in IEEE Real-Time and Embedded Technology and Applications Symposium. IEEE Computer Society, 2012, pp. 219–228.

[70] I. Beer et al. ‘‘The temporal logic sugar,’’ in Proc. Comput. Aided Verific 2001.

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

[72] Stephen Williams. Icarus verilog. http://www.icarus.com/eda/verilog/index.html

[73] E.Jakumeit, S .Buchwald, M.Kroll, GrGen.NET: The expressive, convenient and fast graph rewrite system,Int.J.Softw.ToolsTechnol.Transf.12(2010)263–271.

[74] Z.Drey, C.Faucher, F.Fleurey, V.Mahe, D.Vojtisek, Kermeta Language Reference Manual, https://www.kermeta.org/docs/KerMeta-Manual.pdf,April2009.

[75] ObjectManagementGroup,Query/View/TransformationSpecification,2010.

[76]F.Jouault, F.Allilaire, J.Bézivin, I.Kurtev, ATL: A model transformation tool, Sci.Comput.Program.72(1–2)(2008)31–39.

[77] OMG, "MOF Model to Text Transformation Language, v1.0 [online]," [Accessed 17 January 2012] Available at: http://www.omg.org/spec/MOFM2T/1.0/, 2008.

[78] Microsoft T4, Last Accessed September 2014. http://msdn.microsoft.com/en-us/library/bb126445.aspx

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

[81] R. Armony et al. ''The ForSpec temporal logic: A new temporal property-specification language,'' Tools and Algorithms for Construct. Analy. Syst., 2002.

[82] N. Fedida, J. Havlicek, N. Levi, and H. Miller, CBV Semantics, 2002. [Online]. Available: http://www.vhdl.org/vfv/hm/att-0672/01-cbv_semantics.pdf

[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] IEEE scientific database, Last Accessed, July 2014. http://ieeexplore.ieee.org/

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

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

[87] IBM, "Eclipse Universal Tool Platform," http://www.eclipse.org, 2008.

[88] D.H. Akehurst, W.G. Howells, and K.D. McDonald-Maier, "Kent Model Transformation Language," Proc. Model Transformations in Practice Workshop, part of the ACM/IEEE Eighth Int'l Conf. Model Driven Eng. Languages and Systems (MoDELS), 2005.

[89] IBM, "Eclipse Modeling Framework," http://www.eclipse.org/gef/ 2008.

[90] D.H. Akehurst and O. Patrascoiu, "Tooling Metamodels with Patterns and OCL," Proc. First Int'l Workshop Metamodelling for MDA, A. Evans, P. Sammut, and J.S. Willans, eds., Nov. 2003.

[91] F. Coyle, and M. Thornton, "From UML to HDL: a Model-Driven Architectural approach to hardware-software co-design". Information Systems: New Generations Conference (ISNG), Las Vegas, 2005.

[92] Xilinx official website, Last Accessed September, 2014. http://www.xilinx.com/

[93] Math Works HDL coder, Last Accessed October, 2014 http://www.mathworks.com/products/slhdlcoder/

[94] 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.

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

[96] L. Li, F.P. Coyle, M.A. Thornton, UML to SystemVerilog synthesis for embedded system models with support for assertion generation, in: Proceedings of the ECSI Forum on Design Languages, 2007.

[97] 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.

[98] Papyrus MDT. Available from: <http://www.eclipse.org/modeling/mdt/papyrus/>.

[99] H. Posadas, E. Villar, Automatic Communication Modelling for early exploration of HW/SW allocation based on native co-simulation, in: Proc. Of XXVI Conf. on Design of Circuits and Integrated Systems, DCIS'11, 2011.

[100] INRIA, KerMeta meta-programming environment, www.kermeta.org.

[101] SODIUS, MDWorkbench platform, www.mdworkbench.com.

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

[103] A. Gamatié, S. Le Beux, E. Piel, A. Etien, R.B. Atitallah, P. Marquet, J.-L. Dekeyser, A model driven design framework for high performance embedded systems, Research Report RR-6614, INRIA, 2008. Available from: <http://hal.inria.fr/inria-00311115/en>.

[104] OMG, M2M/Operational QVT Language, 2007. Available from: <http://tiny.cc/tFGGx>.

[105] Acceleo, MDA code generator, 2009. Available from: <http://www.acceleo.org/pages/home/en>.

[106] Gamatié A, Le Beux S, Piel E, Ben Atitallah R, Etien A, Marquet P, Dekeyser JL (2011) A model driven design framework for massively parallel embedded systems. ACM Trans Embed Comput Syst 10(4):2–36

[107] Boulet P (2008) Formal semantics of Array-OL, a domain specific language for intensive multidimensional signal processing. INRIA res rep RR-6467

[108] Demeur A, Del Gallo Y (1998) An array approach for signal processing design. In: Proceedings of Sophia-Antipolis conference on micro-electronics (SAME98)

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

[110] Herrera F, Villar E (2006) A framework for embedded system specification under different models of computation in SystemC. In: Annual ACM IEEE design automation conference proceedings of the 43rd annual conference on design automation

[111] OMG: Object Constraint Language Specification, Version 2.2. Object Management Group Inc. (2010). http://www.omg.org/spec/OCL/2.2/

[112] 68. Oldevik, J.: MOFScript user guide. Version 0.6 (MOFScript v 1.1. 11) (2006)

[113] I. Gray and N. Audsley, "Exposing non-standard architectures to embedded software using compile-time virtualisation," in International conference on Compilers, architecture, and synthesis for embedded systems (CASES'09), 2009.

[114] Artisan Studio, Last Accessed October, 2014. http://www.atego.com/products/atego-modeler/

[115] Eclipse Modelling Framework. Available from: <http://www.eclipse.org/modeling/emf/>.

[116] Catapult Last Accessed, October, 2014. www.calypto.com/en/products/**catapult**/overview/

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

[118] Zot tool, Last Accessed, October 2014. http://home.deib.polimi.it/pradella/Zot/

[119] Xilinx ISE, Last Accessed, October 2014. http://www.xilinx.com/products/design-tools/ise-design-suite.html

[120] Ge, N., Pantel, M.: Time Properties Verification Framework for UML-MARTE Safety Critical Real-Time Systems. In: Vallecillo, A., Tolvanen, J.-P., Kindler, E., Störrle, H., Kolovos, D. (eds.) ECMFA 2012. LNCS, vol. 7349, pp. 352–367. Springer, Heidelberg (2012)

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

[122] DEVS Last Accessed, September 2014. http://en.wikipedia.org/wiki/DEVS

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

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

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

[126] Visual Paradigm, Last Accessed October 2014. http://www.visual-paradigm.com

[127] OMG. Action Language for Foundational UML (Alf), Concrete Syntax for a UML Action Language, FTF –Beta 1. Technical report, 2010.

[128] Modelica, Last Accessed November, 2014. https://www.**modelica**.org/

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

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

[131] Magic Draw, Last Accessed October 2014. http://www.nomagic.com/products/magicdraw.html

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

[133] Eclipse Epsilon Last Accessed October 2014. http://www.eclipse.org/epsilon/

[134] Eclipse JET Last Accessed October 2014. http://www.eclipse.org/modeling/m2t/?project=jet

[135] Eclipse HENSHIN Last Accessed October 2014. http://www.eclipse.org/henshin/

[136] Eclipse VIATRA Last Accessed October 2014. https://eclipse.org/viatra2/

[137] Mentor Graphics Questa SIM, Last Accessed November 2014 http://www.mentor.com/products/fv/questa/

[138] Elsevier Last Accessed July 2014. http://www.sciencedirect.com/

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

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

[141] General Public License, Last Accessed August 2014. http://www.gnu.org/licenses/gpl.html

[142] Eclipse Public License, Last Accessed August 2014. https://www.eclipse.org/legal/epl-v10.html

[143] Apache License, Last Accessed August 2014. http://www.apache.org/licenses/

[145] CeCILL License, Last Accessed August 2014. http://www.cecill.info/index.en.html

[146] PolarSys, Last Accessed October 2014. http://polarsys.org/

[147] Eclipse Platform, Last Accessed June, 2014. http://**eclipse**.org/**eclipse**/

[148] Magic Draw Clients, Last Accessed November 2014. *http://www.nomagic.com/about-us/our-clients.html*

[149] Magic Draw Blog, Last Accessed, November 2014. *http://blog.nomagic.com/why-top-engineering-companies-use-no-magic-inc-products-as-their-primary-mbse-selection/*

[150] PolarSys, Member list, Last Accessed November, 2014. http://polarsys.org/members

[151] Atego transport company, Last Accessed November 2014. http://www.atego.com/download-center/case-study/alstom-transport/

[152] Atego CIRA, Last Accessed November 2014. http://www.atego.com/download-center/case-study/cira/

[153] Atego PSA, Last Accessed November 2014. http://www.atego.com/download-center/case-study/psa-peugeot-citroen/

[154] ATL partners,Last Accessed November 2014. http://projects.eclipse.org/projects/modeling.mmt.atl

[155] Sodius partners, Last Accessed November 2014. http://sodius.com/about/partners

[156] QVT Last Accessed November 2014. http://en.wikipedia.org/wiki/QVT

[157] Epsilon Users, Last Accessed November 2014. http://eclipse.org/epsilon/users/

[158] Henshin Users, Last Accessed November 2014. https://projects.eclipse.org/projects/modeling.emft.henshin/

[159] Prism partners, Last Accessed November 2014. http://www.prismmodelchecker.org/sponsors.php

[160] MathWorks Users, Last Accessed November 2014. http://www.mathworks.com/company/user_stories/

[161] ALDEC partners,Last Accessed November 2014. https://www.aldec.com/en/company/partners

[162] Calpto Partners, Last Accessed November 2014. http://calypto.com/en/solutions

[163] Topcased Users, Last Accessed November 2014. http://www.topcased.org/index.php/content/view/32/49/

[164] TTool sponsors, Last Accessed November 2014. http://ttool.telecom-paristech.fr/reference.html