Chapter 1: Literature Review

1. Introduction

Model Based System Engineering (MBSE) is a renowned approach for the development of complex systems. It has features set to reduce development complexity, enhance productivity, efficient change management and significantly improve product time-to-market. Therefore, it has been frequently researched and customized for the development of embedded systems [11-12][18-19][21-23]. The major MBSE activities for the development of embedded systems are shown in Figure 1.

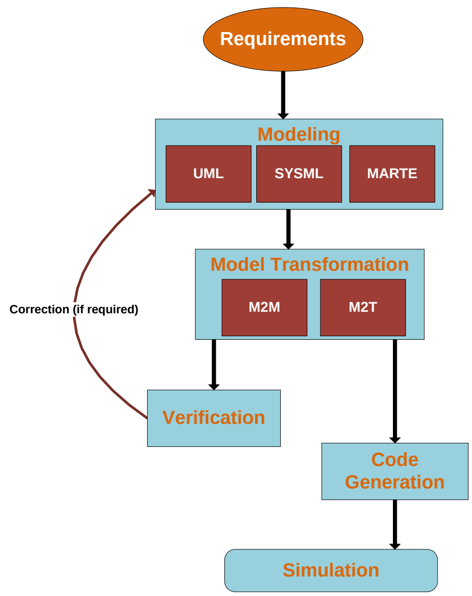

Figure 1: Major MBSE Activities for Embedded Systems

Modeling structural and temporal aspects of embedded systems is foremost activity. All other MBSE tasks (i.e. model transformation, verification and validation) are based on the models development methodology. Therefore, models are developed by taking into consideration the model transformation, verification and simulation requirements. For example, one of the major challenge is to model behavioral / temporal aspects of complex embedded systems for further verification and validation [12][34]. UML and its SYSML/MARTE profiles are frequently used in contemporary research practices [1-20] to specify embedded systems requirements. Furthermore, different properties specification techniques / languages have been proposed by researchers [31-32][55-57] to model behavioral / temporal aspects. Once requirements are modeled, different model transformation techniques have been applied to develop platform specific model and / or source code generation. Two types of transformations are commonly used i.e. Model-to-Model (M2M) transformation [2-3] and Model-to-Text (M2T) transformation [17-18].

The verification is performed to ensure the correctness of the model / system and it is tightly coupled with the modeling technique used to specify behavioral / temporal aspects. Various formal verification techniques [47-49] have been used to verify the behavioral /

temporal aspects of the system. If the model does not satisfy the verification requirements, then corrections have been made in the model as shown in Figure 1. The validation of the model / system has been performed through simulation. Model transformation is frequently performed to produce the desired source code from the model which is used for simulation in order to validate the model / system.

Although, researchers put a lot of efforts in the field of MBSE for embedded systems, it is still a challenging area due to the diversity of behavioral / temporal characteristics of embedded systems. It is always difficult to select appropriate modeling techniques and UML profiles to model embedded systems requirements. Moreover, there is a considerable dependency among different MBSE activities (i.e. modeling, model transformation, verification and simulation) that require sufficient knowledge of all the phases for the development of embedded systems. Similarly, there are separate toolsets for each MBSE activity and selection of appropriate tools for embedded system development is always problematic for researchers and practitioners.

Hence, there is a need of generic solution that provide support for all MBSE tasks in order to ease modeling, verification and validation process for variety of safety-critical embedded systems. Therefore, keeping in view the current state of affair, MODEVES project is introduced to ease the model-based development, for variety of safety-critical embedded systems.

1.1 Objective

This Literature Review report is developed as a first step towards the tools selection and requirements specifications of MODEVES. In addition, it also provides latest MBSE trends and approaches to assist the development of MODEVES. The significant objectives of the literature review report are:-

- Identification of contemporary research practices pertaining to the development of embedded systems by using MBSE approach.
- Investigation of latest modeling trends to specify embedded systems requirements in the context of UML and its SYSML/MARTE profiles.
- Investigation of transformation approaches (i.e. M2M or M2T) frequently used in contemporary research practices.
- Identification of significant tools for modeling, model transformation, verification and simulation (validation) activities in the context of MBSE for embedded systems.

This report leads us to distinguish MODEVES goals from current research and industrial practices. It further helps us for requirement specification process of MODEVES. Furthermore, it also facilitates us for tools selection process.

1.2 Overview

The complete overview of the literature review process and tools selection is shown in the Figure 2

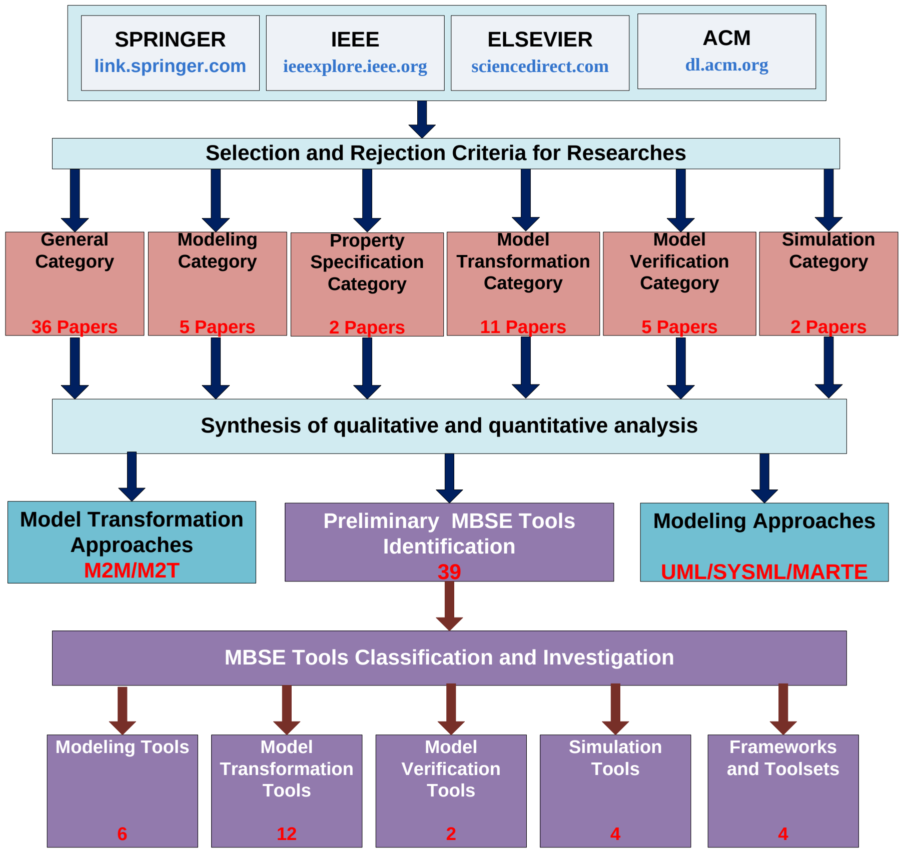

Figure 2: Overview of Research

A comprehensive research methodology (Section 2), based on Systematic Literature Review (SLR) [64] has been developed to perform this literature review. A review protocol is developed (Section 2.2) that contains selection and rejection criteria (Section 2.2.2). As shown in Figure 2, four scientific databases are selected for search process (Section 2.2.3) and six categories are defined (Section 2.1) to classify the 61 selected researches. Similarly, data extraction elements (Section 2.2.5) are defined to perform comprehensive analysis and synthesis of the selected researches. Consequently, the major findings of literature review have been summarized in Section 3. On the basis of SLR, we preliminary identified 39 tools (Section 4) those have been used in the selected researches to perform particular MBSE activities. We realize that further tools investigation (Section 4) is required to select the appropriate MBSE tools for MODEVES. Therefore, various tools characteristics (Section 4.2) are defined for further evaluation. The characteristics-based evaluation eliminated few preliminary MBSE tools lacking certain common characteristics and included some additional tools those have been missed during the SLR. Consequently, 28 MBSE tools in five different categories (Section 4.3) are presented as shown in Figure 2. We present the selection of tools for MODEVES in Section 5. Finally, industrial feedback regarding MBSE tools is presented in Section 6.

2. Research methodology

Systematic Literature Review [64] has been used to carry out this research. It is a proper and replicate process to document pertinent details on precise research area for reviewing and investigating all existing research related to research objective. This research incorporates various stages: 1) Categories definition 2) Review protocol development 3) Selection and rejection criteria 4) Search process. 4) Quality assessment. 5) Data extraction and synthesis.

2.1 Categories Definition

We have defined six categories in order to organize the selected researches. This categorization significantly improves the accuracy of the answers of our research questions. The details of categories are given below:

2.1.1 General Category

There might be a number of researches where complete solution covering all MBSE activities (i.e. modeling, model transformation, verification and simulation) is proposed for embedded systems. Moreover, there might be a number of researches covering all MBSE activities but do not particularly intended for embedded systems, however, these researches have great potential to be used for embedded systems. Furthermore, some researches might cover more than one MBSE activities simultaneously (e.g. modeling, model transformation or model transformation, verification etc). All aforesaid researches will be included in the general category.

2.1.2 Modeling Category

Requirement specification is the foremost activity. Therefore, all proposed categories may contain some information about the model development methodology. UML and its SYSML/MARTE profiles are normally used individually as well as collectively to specify embedded systems requirements. However, there might be a number of researches particularly intended to propose / investigate modeling methodologies by using UML and its SYSML/MARTE profiles. Such researches will not provide any significant proposal regarding any other MBSE activities (i.e. model transformation, verification and validation). All such researches will be included in the modeling category.

2.1.3 Property Specification Category

It is an important activity to specify behavioral / temporal aspects of embedded systems through property specification techniques / languages. The researches may propose novel approaches to specify behavioral / temporal aspects and constraints of embedded systems. All such researches will be included in the property specification category.

2.1.4 Model Transformation Category

Model transformation is an important MBSE activity. There are latest research works pertaining to either Model-to-Model or Model-to-Text transformation techniques or both of these. There might also be some research works to ensure the correctness of model transformation. All such researches will be included in the model transformation category.

2.1.5 Model Verification Category

There might be a number of researches pertaining to formal and informal model verification techniques to ensure the correctness of the model. Normally, behavioral / temporal aspects of the model have been verified through formal verification methods. All such researches will be included in the model verification category.

2.1.6 Simulation Category

The simulation is normally used for system validation by utilizing the generated source code. There might be a number of researches those integrate available simulation tools. However, there might also be few researches where new simulation mechanism has been proposed. Similarly, there is a possibility of some research works where particular simulation aspects are also included in the model to ensure the accuracy of simulation. All such researches will be included in the simulation category.

It is worth-mentioning here that the *General Category* will contain a number of researches where solutions are either provided for all MBSE activities or at least two MBSE activities simultaneously. On the other hand, the researches particularly targeted for any one of the MBSE activity (i.e. modeling, model transformation, verification and simulation) will be included in their corresponding defined categories.

2.2 Review Protocol Development

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

2.2.1 Background and Research Questions

Background of the research is already described in the introductory part of this article (Section 1). Keeping in view the objectives (Section 1.1) of this research, following research questions have been developed:

RQ1: What important researches have been reported from 2008 to 2014 where MBSE approach has been utilized to support the development of embedded systems?

We get the answer of the question in Section 3.3. Details are summarized in Table III.

RQ2: Which of the UML and its SYSML/MARTE profiles are more frequently utilized to model embedded system requirements during 2008 to 2014 researches?

We get the answer of the question in Section 3.1

RQ3: Which of the Model-to-Model or Model-to-Text transformation approach is more frequently utilized during 2008 to 2014 researches?

We get the answer of the question in Section 3.2

RQ4: What are the significant tools for requirement specifications, model transformation, verification and simulation (validation) activities in the context of MBSE for embedded systems?

We get the answer of the question in Section 4

After getting the answers of these research questions, we achieve our project objectives (Section 1.1) in Section 3.4 and Section 5.

2.2.2 Selection and Rejection Criteria

We define concrete criteria for the selection and rejection of research works. Six parameters are defined to ensure the correctness of the answers of our research questions. The research work will be selected on the basis of these parameters as given below:

1. ***Subject-Relevant***: Select the research work only if it is relevant to our research context. It must support the answers of our research questions and must be relevant to one of the six predefined categories (Section 2.1). Reject irrelevant researches those do not belong to any of the six predefined categories.
2. ***2008-2014***: Selected research work must be published from 2008 to 2014. Reject all researches those are published before 2008 to ensure the inclusion of latest research works.
3. ***Publisher***: Selected research work must be published in one of the four renowned scientific databases i.e. IEEE [84], SPRINGER [139], ELSEVIER [138] and ACM [140].
4. ***Crucial-effects***: Selected research work must have crucial positive effects regarding embedded system development through MBSE approach. Reject research work if its proposal doesn't have significant consequences on embedded system development.
5. ***Results-oriented***: Selected research work must be results-oriented. The proposal and ultimate outcomes of the research must be supported by solid facts and experimentation. Reject the research work if its proposal is verified through weak validation method.

6. ***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.

2.2.3 Search Process

Selection and rejection criteria, presented in Section 2.2.2, shows that we have selected four scientific databases (i.e. IEEE, ELSEVIER, SPRINGER and ACM) in order to carry out this SLR. These scientific databases contain high impact journals and conference proceedings. In addition, we have also studied relevant books, white papers and technical reports for sustenance of our study and investigation. To accomplish search process, we use different search terms like MBSE, model transformation, model verifier etc. We also make use of AND / OR operators to acquire most relevant search results. The search terms along with the results for each scientific database are summarized in Table I.

Table I: Details of search terms with operators and search results.

Sr. #	Search Term	Operator	No. of Search Results			
			IEEE	SPRINGER	ELSEVIER	ACM
1	MBSE	N-A	67	2	150	119
2	MDA	N-A	797	3327	7313	3520
3	MDE	N-A	355	256	2581	1990
4	SYSML	N-A	274	11	373	790
5	UML	N-A	2564	377	5645	3796
6	MARTE	N-A	199	80	1470	176
7	Model Transformation	AND	710	89	1254	787
8	Model Verification	AND	302	122	3074	146
9	Formal Verification	AND	4941	110	1005	1297
10	Model Simulation	AND	980	1610	18790	272
11	Model embedded system	OR	10659	19480	2403	21605
		AND	1	0	4	2
12	MARTE	OR	18	0	21	43

	SYSML	AND	1	0	0	2
13	SYSML UML	OR	82	7	148	141
		AND	3	0	15	19
14	MDA embedded systems	OR	42	689	4516	246
		AND	0	0	0	0
15	Property specification	OR	1948	5794	1477	1853
		AND	66	13	319	111
16	Model to Text transformation	OR	135	5839	36	929
		AND	7	4	74	57
17	Model to Model Transformation	OR	8790	23686	9144	17658
		AND	20	9	120	101
18	Model verifier	OR	175	248	101	154
		AND	16	2	24	25
19	Model code generation	OR	2667	16307	860	2162
		AND	0	0	6	7
20	Embedded system Simulation	OR	5610	3457	975	1680
		AND	6	0	6	5

We use "*2008-2014*" filter for all the search terms to only get the researches published during 2008-2014. Some search terms comprise of single word (e.g. MBSE). Consequently, the use of AND/OR operator is not applicable (shown as N-A in Table I). The search results obtained through the AND operator do not guarantee the relevance of our research context. Therefore, we also use the OR operator to attain potential search results required for our research. However, using the OR operator provide thousands of search results so it is not possible to scan all the results. Consequently, we also use advance search options (e.g. "where abstract contained", "where title contained" etc) provided by these scientific databases in order to get precise results. Similarly, after investigating the primarily search results, we further use more specific search terms to get desired search results. For example, we use "CCSL" to get the desired search results for properties specification techniques in models. The relevant screenshots pertaining to Table I can be viewed at **<<our website address>>**

The following steps have been performed during the search process (depicted in Figure 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* as per selection and rejection criteria.
4. We perform general study of 1396 researches by reading different relevant sections of each research. On basis of our general study, 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. We select 61 researches fully compliance with our pre-defined selection and rejection criteria

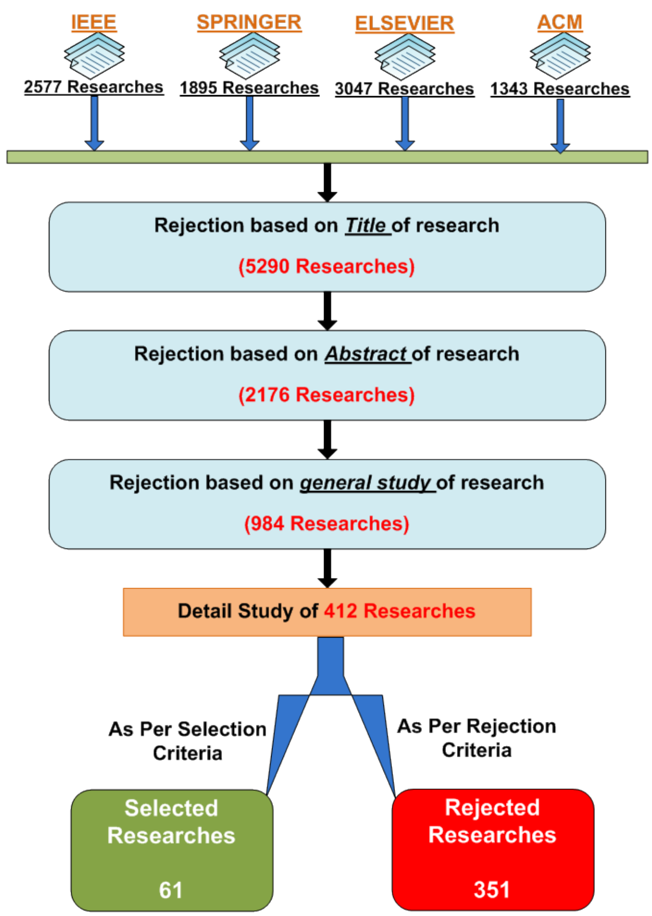

Figure 3: Search Process

2.2.4 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: -

(1) The data appraisal of the research is based on the concrete facts and theoretical perceptive without any vague statements.
(2) The validation of research has been performed through proper validation methods e.g. case study etc
(3) The research provides information about the tools used in it to perform MBSE activities.
(4) As we intend to investigate latest MBSE tools and trends, the objective is to include most recent researches as much as possible. Therefore, we try to include most recent researches as 69 % researches are from 2012 to 2014 and overall 87% researches are included from 2010 to 2014 as shown in Figure 4
(5) The journal publications provide more research details as compared to conference publications. Therefore, we try to include journal publications as much as possible to attain good quality of our SLR. We identified 61 researches and 43% of those are journal publications as shown in Figure 5.
(6) Originality of the research is another important factor. Therefore, we only include researches those are published in at least one of the four renowned and globally accepted scientific databases i.e. IEEE, SPRINGER, ELSEVIER and ACM.

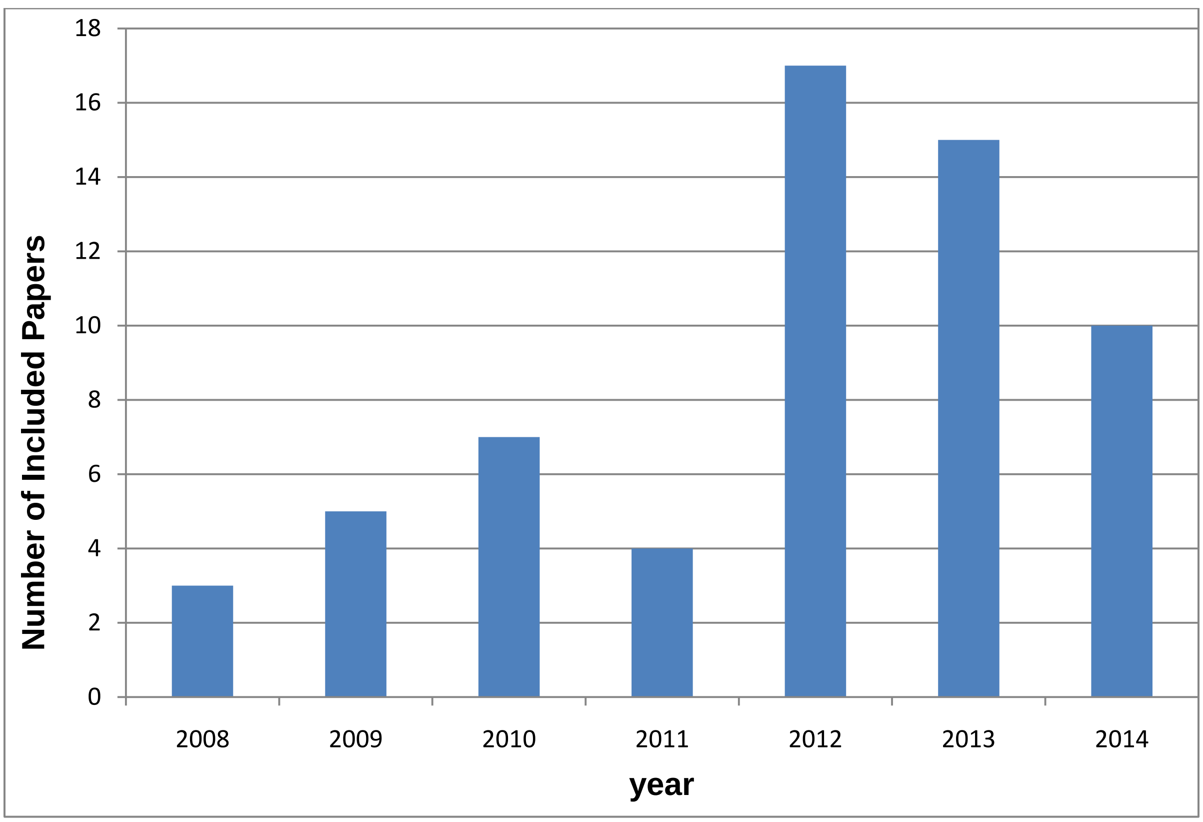

Figure 4: Number of Selected Researches per Year

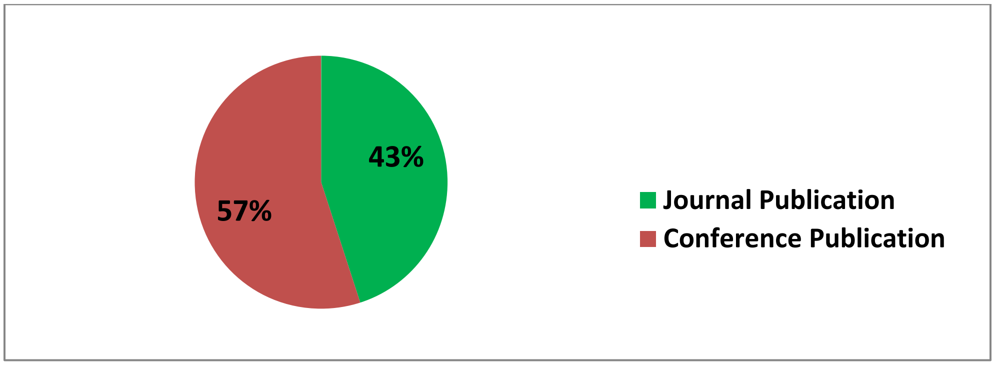

Figure 5: SLR Statistics Pertaining to Journal / Conference Publication

2.2.5 Data Extraction and Synthesis

We thoroughly study all selected researches in order to extract relevant data for getting answers of our research questions. The relevant data extracted from all selected researches have been defined in Table II

Table II: Details of data extraction for each research

SR. #	Extracted Data	Details
1	Bibliographic information	Title, author, publication year, publisher details and type of research i.e. journal or conference publication
Extraction of data		
2	Overview of research	The basic proposal and objective of selected research
3	Result of research	Results acquired from the selected research
4	Data collection of research	Quantitative or qualitative
5	Assumptions	Assumptions (if any) to achieve or validate the results of the selected research
6	Validation method of research	Validation method used in the selected research to validate its proposal
Synthesis of data		
7	Classification of research	Relevance of selected research to one of the predefined categories (Section 2.1)
8	UML profiles utilization	UML and its SYSML / MARTE profiles used in the selected research (Table IV)
9	Transformation	Type of transformation used in the selected research

	type	i.e. M2M, M2T or both (Table V)
10	Identification of tools	Tools used in the selected research to perform particular MBSE activity (Table VI)

We extract general information of each selected research like publisher, publication year etc as defined in serial #1 of Table II. For data extraction, defined from serial # 2 to 6, we extract important details of each selected research to ensure its compliance with the selection and rejection criteria. For data synthesis, defined from serial # 7 to 10, we perform detailed analysis of each research. For example, all selected researches have been thoroughly studied and analyzed in order to assign them to corresponding category. Similarly, each selected research is intensively studied and analyzed to extract the accurate information regarding UML profiles utilization, transformation type and identification of tools as defined in serial # 8, 9 and 10 respectively.

3. Results

We have identified 61 researches and classified them into six categories. The summary of classification is shown in Table III

Table III: Classification results for identified researches

Sr. #	Category	Number of Researches	Researches identification
1	General	36	[1][2][3][5][6][7][8][9][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][29][33][44][52][53][55][56][58][59][60][61]
2	Modeling	5	[10][28][30][34][57]
3	Property Specification	2	[31][32]
4	Model Transformation	11	[37][38][39][40][41][42][43][45][46][50][51]
5	Model Verification	5	[35][47][48][49][54]
6	Simulation	2	[4][36]

From the results of SLR, we analyze that various MBSE activities are simultaneously researched to provide complete development solution for embedded systems. Therefore, 36 researches have been selected under the general category as shown in Table III. We also analyze that modeling is the foremost activity and mostly discussed along with other MBSE activities. For example, Takashi et al. [4] propose simulation of SYSML model through SIMULINK [65] tool. However, they also provide adequate SYSML modeling information in the context of simulation. We observe that there are very few researches where "only" modeling aspects are discussed, as shown in Table III. Similarly, we identify two researches those are specific to property specification category only. However, various property specification techniques are also proposed along with other MBSE activities. For example, Daniel et al.[55] propose TEPE for property specification but they also provide detailed solution for transformation and verification by using TTool [117]. Consequently, we place such researches under the general category. Similar is the case with the model transformation (e.g. [1][12]), model verification (e.g. [11][33]) and simulation (e.g. [6]) categories.

3.1 UML/SYSML/MARTE Utilization

We have also identified the application of UML and its SYSML/MARTE profiles individually as well as simultaneously to model embedded systems requirements. The results are summarized in Table IV

Table IV: Results pertaining to UML and its SYSML/MARTE profiles utilization

Sr. #	**Profiles**	**Number of Researches**	**Researches identification**
1	UML	7	[1][13][14][16][17][24][53]
2	SYSML	15	[2][3][4][5][6][7][8][9][27][33][44][52][55][57][61]
3	MARTE	0	
4	UML and SYSML together	2	[15][27]
5	UML and MARTE together	12	[18][19][20][21][22][23][26][28][29][56][58][60]
6	SYSML and MARTE together	4	[11][12][30][59]
7	UML, SYSML and MARTE altogether	3	[10][25][34]

We include all researches in Table IV where UML and its SYSML/MARTE profiles are comprehensively discussed and / or practically used individually as well as simultaneously. However, there are some researches, especially in model transformation (e.g. [41][43]) and model verification (e.g. [47-48]) categories, where these profiles are rarely discussed and /or practically used. Therefore, we exclude all such researches due to their irrelevance with our particular research objective. Another interesting finding is that MARTE profile is rarely used alone to model large and complex embedded systems requirements. In fact, MARTE profile has very good features to model temporal aspects of the system but it cannot completely handle all behavioral and structural aspects of large systems. However, its utilization along with UML and SYSML certainly enhances the modeling capabilities for embedded systems. Therefore, it is frequently used with UML and SYSML as depicted in Table IV.

3.2 Model Transformation Techniques

We analyze model transformation techniques used in selected researches. The results are summarized in Table V

Table V: Model transformation statistics

Sr. #	Transformation Type	No. of Researches	Relevant Researches
1	Model to Model (M2M)	14	[2][3][8][14][15][37][38][39][40][41][45][46][51][59]
2	Model to Text (M2T)	11	[9][17][18][20][21][29][42][43][52]
3	Both M2M and M2T	12	[1][6][12][19][22][23][26][27][44][50][58][61]

The model transformation activity has been performed in overall 37 researches as given in Table V. We classified these researches into three transformation types i.e. M2M, M2T, both M2M and M2T. The corresponding researches against each transformation type have been identified as given in Table V.

3.3 Overview of Selected Researches

We extract the important data for each selected research, according to the attributes defined in Table II, in order to get the answers of our research questions. The important data extraction for each selected research is given below:

Overview of Selected Researches

Sr #	Author	Overview	Method of Validation	Category	Details
1 **2012**	Guglielmo et al. [1]	This paper presents integration approach of Model Driven Design (MDD) and Assertion Based Verification (ABV) in embedded systems for efficient design and verification.	Experiments and examples	General	This paper introduces integration of MDD and ABV by developing two tools i.e. radCASE and radCHECK. radCASE is developed to specify requirements in UML. Use case diagrams are used to specify high level requirements. Structure can be defined through Class, Composite structure, object and component diagrams. Behavior can be defined in state chart, sequence and activity diagrams. The developed models are then transformed to EFSM. It also incorporates the facility to generate complete C-code including ABV as defined in design. On the other hand, radCHECK tool provides facilities for dynamic ABV verification of embedded system design developed in redCASE. Ulisse stimuli generation engine is used in radCHECK to produce effective stimuli for verification.
2 **201**	Kapos et.al [2]	This paper first highlights problems of SysML model simulation in MDA. Then, the generation of executable simulation code from SysML system models	Case study (Discrete event simulation)	General	This paper highlights problems of SysML model simulation in MDA. Thereafter, the generation of executable simulation code from SysML system models is

4		are presented.			presented. Authors also provide comprehensive literature review about MDA Modeling in SysML, Model transformation techniques / tools and code generation methodologies. Authors use QVT to produce executable DEVS models from SysML models. MediniQVT tool [121] is used to develop DEVS [122] MOF 2.0 meta-model and QVT transformation
3 **2 0 1 3**	Berrani et al. [3]	This paper proposes transformation of SysML requirement diagram in Modelica to improve WSN properties using MDA approach	Case Study (Traffic lights)	General	This paper introduces transformation of SysML requirement diagram in Modelica to improve WSN properties using MDA approach. Authors introduces modeling scheme for wireless sensor network (WSN) where SysML model is developed in Topcased tool [62] using Block, internal block, parametric, state machine diagrams to specify behavioral aspects. Thereafter, ATL [63] tool is used to transform SysML model into Modelica [128] and then Modelica model is transform into text.
4	Sakairi et al. [4]	This paper introduces concept of SysML model and Simulink integration	Example of Dual Clutch Transmissio	Simulation	This paper proposes a tool to integrate SysML with

2012		for MBSE.	n (DCT)		Simulink [65] simulation tool. Dual Clutch Transmission (DCT) model is developed in Rhapsody using SYSML and validation is performed through Simulink. Block diagram is used to specify simulation context and Simulink sub-models are referred through SysML. .
5 **2010**	Stancescu et al. [5]	This paper proposes Verilog code generation from SysML model	Example (Reed-Solomon decoder)	General	This paper proposes scheme for mapping Verilog modules to SysML parts, Verilog ports & signal to SysML flow port and Verilog process to SysML allocations. The SYSML model is developed and exported in XMI format to generate Verilog code for validation.
6 **2014**	Tsadimas et al. [6]	This paper introduces the concept of Evaluation View (diagram) to integrate simulation capabilities into SysML for DEVS simulation environment.	Case Study (Registry application)	General	This paper proposes SYSML based Evaluation View to describe the system under verification and conditions under which performance should be evaluated. Thereafter, executable simulated code is generated through evaluation view. Authors also include the simulation results in Evaluation View to verify performance requirements.

7 **2 0 1 3**	Ouchani et al. [7]	This paper proposes the verification of system design (SysML) through formal verification framework.	Case study (audio player system)	General	This paper proposes formal verification framework for the verification of SysML based system design. The requirements are specify in SYSML activity diagrams through PCTL properties. The SysML model is verified through PRISM [66]. Thereafter, novel algorithm is proposed to transform SysML model to BlueSpec SytemVerilog code.
8 **2 0 1 2**	Bouquet et al. [8]	This paper proposes scheme to generate VHDL-ASM code from SysML model.	Case study (Smart Surface system)	General	This paper proposes scheme to generate VHDL-ASM code from SysML model. Topcased tool is used to develop SYSML model through block definition and internal block diagrams along with behavior and constraints in parametric diagram. Model to Model transformation is performed through ATL and code generation from model to text is performed through Xpand.
9 **2 0 1 3**	Tommasi et al. [9]	This paper proposes SysML modeling scheme based on the MARTe framework for designing real time application	Case study (Control System FTU)	General	This paper proposes SysML modeling scheme based on the MARTe framework for designing real time application. Topcased tool has been used for Modeling and Acceleo [67] tool is used for model-to-text transformation for code generation.
10	Gomez et al. [10]	This paper proposes multi-view modeling scheme for embedded	Examples and theoretical	Modeling	This paper proposes multi-view modeling scheme for embedded

2012		systems using UML, SysML and MARTE	reasoning		systems using UML, SysML and MARTE. Each domain of proposed scheme can be managed in various views interconnected with each other. MARTE is used to for hardware model and non-functional properties. SysML is used to define equations.
11 **2009**	Andrade et al. [11]	This paper proposes requirement validation scheme for embedded systems by mapping SysML activity diagram to time petri net.	Case Study (pulse-oximeter)	General	This paper proposes SysML activity to specify requirements. MARTE notations are used for execution time / energy consumption constraints. The developed model is mapped into ETPN which is used to verify different behavioral and structural aspects of ERTS systems including execution time and energy consumption.
12 **2012**	Quadri et al. [12]	This paper proposes combined utilization of SysML / MARTE notations for modeling embedded system requirements by developing tools for design specifications, validation and simulation for MADES FP7 Project.	Case Study (Car collision avoidance system)	General	This paper discuss various tools / techniques of MADES Project for design specifications, validation and simulation are discussed e.g., MADAS language, diagrams etc. Car-Collision-Avoidance case study is modeled in Open source Modelio Editor [68] using MADES language. Hardware code is generated according to specifications by using Xilinx ISE [119] and EDK tools. On the other hand, Software code is also generated as per design specifications by using Anvil J [69] and CTV [113]. Zot tool [118]

					is used for verification and simulation.
13 2014	Bazydlo et al. [13]	This paper proposes novel methodology to transform UML state machine diagram into Verilog code.	Example (Trolley Control System)	General	This paper transforms UML models based on state machine diagram into Verilog code. Firstly, UML model is exported in XMI and temporary HCFSM model is developed. Secondly, Verilog code is generated from HCFSM. Features of some important current MDE tools and techniques are also discussed by authors e.g., MODCO [83] and MODEASY[84] etc
14 2013	Doligslski et. al [14]	This paper proposes modeling scheme based on UML and Petri net. VHDL / Verilog code is generated.	Examples and theoretical reasoning	General	This paper proposes HCFSM based modeling scheme using UML state machine for reconfigurable logical controllers. HCfgPN is developed which is used to generate VHDL / Verilog code. Transformation is done using QVT and VHDL / Verilog code is automatically generated using Eclipse M2M engine. Simulation is done using Aldec ActiveHDL and in-circuit verification is performed throughVirtex II-Pro
15 2010	Mueller et al. [15]	This paper proposes UML / SysML profiles to support co-modeling of C / SystemC with code generation facility in ARTiSAN studio.	Case Studies (Broadband wireless communication and ALPR)	General	This paper introduces UML/SysML profiles used in SATURN project for synthesizable C and SystemC in order to support SystemC / C co-modeling by customizing Artisan Studio [114] SysML editor. This

					provides co-simulation facilities in QEMU software. This further helps to generate VHDL code from SystemC which can be used in ISE / EDK framework for FPGA configuration. Brief description of Artisan Studio is given that have automatic code generation facilities in various languages e.g., Java, ADA, SystemC etc
16 **2012**	Durand et al. [16]	This paper introduces new tool for transformation of UML model into the code of Bluespec System Verilog.	Examples and theoretical reasoning	General	This paper introduces new tool to generate Bluespec System Verilog code from UML models comprise activity and state diagrams. Proposed tool accept XMI format and then parse the XMI to generate BSV code. Magic draw tool having XMI export facility is used to develop UML model. XMI file analysis including tags evaluation and information identification is accomplished using XMLtool[85]. Apache Velocity [86] software is used to generate BSV code through VTL templates.
17 **2008**	Wood et al. [17]	This paper proposes transformation of UML models (state diagrams) into VHDL code by developing MODEASY transformation tool.	Case Study (Early Warning System)	General	This paper proposes MODEASY tool to transform UML state diagram model into VHDL code. QVT alike language [88] is used to specify transformation rules. MODEASY tool is developed using Eclipse platform [87] along with

					GEF [89] for visual editors. OCL-based template language [90] is used for code generation.
18 **2 0 1 0**	Moreira et al. [18]	This paper proposes methodology to automatically generate VHDL code from UML models with the help of GenERTiCA tool	Case Study (Valve component system)	General	This paper proposes new technique to automatically generate VHDL code form UML models. Script-based approach is used to generate code through GenERTiCA tool [94]. Authors also discuss tools and techniques for VHDL code generation from models. For Example, StateCAD [92] tool that generate VHDL and System Verilog code from its bubble diagrams model. Simulink HDL Coder [93] generates Verilog and VHDL code from Simulink models.
19 **2 0 0 9**	Vidal et al. [19]	This paper proposes MoPCoM technique to automatically generate VHDL code from UML / MARTE models.	Case study (Viterbi decoder)	General	This paper proposes MoPCoM techniques that define three design levels (AML, EML and DML) where DML design rules are used for automatic code generation. IBM Rhapsody is used for modeling. MDworkbench [95] tool is used for transformation.
20 **2 0 1 1**	Linehan and Clarke [20]	This paper presents a domain-specific modeling language for e HVL. The e Modeling language meta model is implemented in UML / MARTE from which e code is generated automatically.	Example of automotive semiconductor industry	General	This paper introduces new UML / MARTE profile that support code template for automatic e HVL code generation. Magic Draw tool is used for modeling and XMI format is used for code generation through Xpand [97].
21	Herrera et	This paper proposes novel UML/MARTE	Example of EFR	General	This paper introduces embedded systems UML /

2013	al [21]	DSE framework for complex embedded systems	vocoder system		MARTE modeling and DSE framework. The main component of framework is MOST tool for automatic design space exploration. COMPLEX Eclipse Application (CEA) tool comprises Eclipse EMF [115] and Acceleo for Model-to-text transformation. PapyrusMDT [98] tool is used for modeling. SCoPE+ [99] tool is used for analysis.
22 **2010**	Lecomate et al. [22]	This paper proposes MOPCOM design technique for building UML/MARTE model and automatic VHDL source code generation from the model.	Example of wireless communication (MIMO decoder)	General	The propose MOPCOM uses three modeling levels (i.e. AML, EML and DML). MOPCOM uses KerMeta [100] language for formalization and validation of models. Rhapsody is used for modeling. Sodius MDWorkbench [101] tool is used for model-to-model transformation. Model-to-text transformation for code generation is performed using CatapultC [116] tool. VHDL, C/C++ and SystemC codes are generated.
23 **2012**	Quadri et al. [23]	This paper investigates the various aspects of UML/ MARTE modeling in embedded systems. Thereafter, Gaspard2 is proposed as an alternative solution	Examples and theoretical reasoning	General	This paper investigates different advantages and limitation of UML/MARTE models to specify embedded systems requirements. Authors prove with examples that Gaspard2 [102][103] framework resolve different issues discover in UML /

					MARTE models. QVTO [104] tool (QVT standard) and Acceleo [105] tool is used for code generation
24 **2013**	DeJin et al. [24]	This paper presents architectural centric methodology for verification and validation of embedded systems.	Case study	General	This paper introduces UML notations to specify behavioral / temporal constraints through EAST-ADL. The models are then transformed to SPIN models for verification.
25 **2008**	Vanderperren et al. [25]	This paper presents comprehensive overview of UML modeling capabilities in System-on-Chip (Soc) design. Furthermore, various UML modeling and transformation tools are also discussed.	Examples and theoretical reasoning	General	This paper investigates the various aspects of UML profiles (SysML and MARTE) in SoC design. Different approaches are presented to efficiently specify structural and behavioral requirements of SoC. Thereafter, different diagrams of SysML and MARTE profiles are discussed for SoC requirements specification. Finally, comprehensive analysis of modeling and transformation tools is presented.
26 **2012**	Elhaji et al. [26]	This paper presents new modeling technique by extending various notations in UML / MARTE. Thereafter, VHDL code is generated from the model and successfully used for the verification of the system.	Case studies (Mesh & torus , honeycomb and GEXspidergon topologies)	General	This paper presents technique to design NoC (Network-on-Chip) by extending various UML / MARTE notations. Gspard2 tool [106] is used for code generation which is available as Eclipse plug-in. SoC design in Gaspard2 is associated to MOC [107] which is based on ArrayOL [108].

27 **2012**	Riccobene et al. [27]	This paper proposes the integration of SysML and SystemC UML profiles.	Case study (Counter System)	General	This paper proposes integration of SysML and SystemC UML profiles by mapping structural aspects of SysML to SystemC UML profile. On the other hand, behavioral aspects are managed though SysML Control flow graphs and SystemC process state machine.
28 **2009**	Penil et al. [28]	This paper proposes novel technique to automatically generate SystemC code from UML/MARTE profile.	Examples and Experiments	Modeling	This paper proposes automatic source code generation of SystemC by extracting system hierarchy and structure from UML / MARTE design. Authors describe the set of designing rules for mapping UML/MARTE models with SystemC specifications. HetSC [109][110] is used for code generation
29 **2013**	Iqbal et al. [29]	This paper presents technique for modeling the environment using UML / MARTE and OCL. These models are used for simulator generation to enable black-box system testing of RTES (Real-time embedded systems)	Case Study (automated bottle recycling system)	General	This paper propose RTES scheme to model constraints, structure and behavior of the environment in UML / MARTE and OCL [111]. Simulator code along with automated test cases is generated in Java using MOFScript[112] through Model-to-text transformation
30 **2009**	Huascar et al. [30]	This paper investigates challenges of combining both SYSML and MARTE profiles for embedded system design	Examples and theoretical reasoning	Modeling	This paper proposes various methodologies to integrate SYSML and MARTRE profiles. it also presents different scenarios where SYSML and MARTE profiles can be potentially combined

					together for the improvements of embedded systems design.
31 **2 0 1 3**	Ling et al. [31]	This paper investigates the potential utilization of CCSL to model scheduling requirements.	Examples and theoretical reasoning	Property specification	This paper investigates the potential utilization of CCSL to model scheduling requirements. State based ccLTS semantics have been proposed for CCSL. This eases the model verification process. The requirements specified in CCSL are then transformed in NuSMV models.
32 **2 0 1 2**	Frederic Mallet [32]	This paper proposes automatic observer generation technique from CCSL specification.	Examples and theoretical reasoning	Property specification	This paper proposes automatic observer generation technique from CCSL specification. The proposal is based on a novel state based semantics for CCSL operators.
33 **2 0 1 1**	Samir and ahmed [33]	This paper presents a novel approach for the verification of interoperability between various system components.	CyCab example	General	The system is modeled in SYSML Block definition, internal block and sequence diagrams. It further presents algorithm for transformation of SysML models into interface automata for verification
34 **2 0 1 4**	Milena et al. [34]	This paper proposes MDEReq to improve requirement specification and change management of embedded systems	Case study (ABS)	Modeling	This paper proposes MDEReq to improve requirement specification and change management of embedded systems by integrating UML models with SYSML and MARTE notations. MDEReqTraceTool has been developed to analyze models and generate traceability matrixes for change management.

35 **2013**	Frederic Mallet [35]	This paper introduces the novel safety mechanism of CCSL specification by transforming it into Marked Graphs for conditions verification	Examples and theoretical reasoning	Model Verification	This paper presents novel safety mechanism by transforming CCSL constraints into marked graph. Thereafter, classical results have been applied on marked graphs to evaluate the safety of CCSL.
36 **2008**	Naiyong et al. [36]	This paper proposes efficient technique for assertion based verification by making use of Alternating Finite Automata (AFA).	Examples and Experiments	Simulation	This paper introduces novel assertion based verification technique by making use of Alternating Finite Automata (AFA). PSL^{simple} –VDV (Verilog Dynamic Verifier) tool is developed by using two Opensource tools i.e. Icarus-Verilog [72] and the GTKWave [71].
37 **2010**	Joel and Ekkart [37]	This paper proposes QVT and TGGs based model transformation scheme.	Examples and theoretical reasoning	Model Transformation	This paper investigates the similarities and differences of QVT and TGGs. Thereafter, mapping scheme of QVT-Core and TGGs is presented. This leads to the combined utilization of both QVT and TGGs to improve model transformation process
38 **2013**	Jong et al. [38]	This paper presents novel scheme to verify the correctness of model transformation.	Experiments	Model Transformation	This paper proposes verification of Model transformation through Meta information, property matching scheme for similarity and graph comparison algorithm.

39 **2009**	Roy et al. [39]	This paper evaluates various transformation attributes of three model transformation languages.	Examples and theoretical reasoning	Model Transformation	This paper presents comprehensive comparison of model transformation languages i.e. CGT, AGG and ATL. The results reveal that CGT has certain advantages over other two in terms of transformation efforts and simplicity.
40 **2012**	Peter and Laszlo [40]	This paper evaluates various attributes of graph rewriting-based transformation languages.	Case study	Model Transformation	This paper investigates various attributes of some popular model transformation languages used in different graph rewriting-based model transformation. The investigation is performed through case study. Thereafter, critical attributes of such transformation languages have been proposed.
41 **2010**	Esther et al. [41]	This paper proposes graphical language for Model-to-Model transformation.	Examples and experiments	Model Transformation	This paper proposes visual language to specify M2M transformations along with their transformation assurance properties. The proposed language support both trace-based and traceless model-to-model transformations. Furthermore, it is also capable to be compiled in OCL. Proof of concept implementation has been developed using Eclipse

					framework.
42 **2012**	Louis et al. [42]	This paper proposes basic feature set for Model-to-Text transformation languages.	Example and theoretical reasoning	Model Transformation	This paper investigates various features of contemporary M2T languages and proposes the basic features model for M2T languages. Applicability of proposed feature model is presented using MOFM2T [77] and Microsoft T4 [78]. However, this research work only proposes basic set of features those should be provided by model-to-text languages.
43 **2012**	Anderson et al. [43]	This paper proposes novel technique MetaTT for M2T transformation	Scenarios	Model Transformation	This paper proposes novel technique MetaTT for M2T transformation. A tool is implemented as a Proof-of-concept using MOFScript and Ecore to support some steps of proposed MetaTT. Validation is performed through four scenarios by comparing M2T transformations of MetaTT with other informal approaches.
44 **201**	Dionisio [44]	This paper proposes novel methodology HIPAO2 for embedded systems development using MDE approach	Case Study UAV	General	This paper proposes HIPAO2 for embedded systems development through MDE approach. Particularly, MDE

4					Modeling and Model transformation (along with code generation) phases are supported by HIPAO2. SysML is used for modeling by employing papyrus editor. ATL is used for M2M transformation and Acceleo is used for M2T transformation
45 **2014**	Juan and Esther [45]	This paper proposes novel transformation approach to improve reusability of transformations between meta-models	Examples and theoretical reasoning	Model Transformation	This paper proposes new graph based transformation technique to improve the reusability of transformations between meta-models. Binding and adapters are used to resolve specified differences between concept and meta-models
46 **2014**	Kolahdouz et al. [46]	This paper proposes framework for evaluation of model transformation techniques.	Case Study	Model Transformation	This paper proposes framework for ISO/IEC 9126-1 quality-attributes based evaluation of model transformation techniques. Five different model transformation techniques (GrGen [73],Kermeta [74],QVT-R [75],ATL [76],UML-RSDS [79]) have been evaluated through proposed framework to show its applicability. Proposed framework facilitates the selection of most appropriate transformation technique

					for particular problem.
47 **2010**	Tuomas et al. [47]	This paper proposes model verification technique for PSL safety properties	Experiments	Model Verification	This paper proposes transducers encoding based methodology to check the model for PSL safety properties. Validation is performed through experiments where two PSL research practices have been compared with proposed methodology.
48 **2009**	Meng et al. [48]	This paper proposes HAPSL to verify PSL properties	Experiments	Model Verification	This paper proposes HAPSL for the verification of both statistical and temporal properties of PSL. Proposed HAPSL is extension of PSL by utilizing Hybrid Automata. Validation is performed through experimentation on two circuits.
49 **2013**	Uchevler and Kjetil [49]	This paper proposes assertion based verification approach.	Case study	Model Verification	This paper proposes novel assertion based verification approach by implementing PSL operations in Haskell using CLaSH tool
50 **2013**	Veronica et al. [50]	This paper proposes MeTHAGeM framework to support model transformations for MDD	Case studies	Model Transformation	This paper proposes MeTHAGeM framework to support model transformations for MDD. It incorporates the collection of DSLs to facilitate the modeling of

					model transformations to reduce the development efforts of model transformations. Furthermore, it also improves the interoperability problems among various model transformation languages.
51 **2012**	Adrian et al. [51]	This paper proposes model transformation technique for including constraints in transformation rules.	Examples and theoretical reasoning	Model Transformation	This paper proposes graph based model transformation technique that facilitates the inclusion of constraints in transformation rules. Proposed technique introduces three stages of model transformation i.e. combined meta-model, specification of transformation rules including constraints and finally, applying the model transformation
52 **2014**	Samir et al. [52]	This paper proposes model verification framework for the verification of behavioral aspects.	Case studies	General	This paper proposes framework for the verification of behavioral aspects specify in SysML activity diagrams. It provides mechanism to represent behavioral aspects modeled in SysML activity diagrams into PRISM model which is then used for verification of PCTL properties.

53 **2013**	Luciane et al [53]	This paper presents Brazilian survey to investigate the usage of UML and MDE for embedded systems	Survey	General	This paper performs survey in Brazil to highlight the potential utilization of UML and MDE for embedded systems. This survey investigates current industrial practices to highlight some critical statistics about UML and MDE utilization for embedded systems
54 **2013**	Bijan and Payman [54]	This paper proposes new approach for equivalence verification and error detection / removal mechanism of RTL designs	Experiments	Model Verification	This paper proposes novel methodology to improve the equivalence verification and error detection / removal mechanism for RTL designs. HED and SMT solvers are used to verify the equivalence among two models at RTL level. Furthermore, mutation based approach is used to improve error detection / removal mechanism. Validation is performed through experiments by comparing proposed methodology using various benchmarks.
55 **2011**	Daniel et al. [55]	This paper proposes TEPE language to graphically express temporal properties through parametric diagrams of SysML	Case study Elevator Ssytem	General	This paper proposes TEPE language to graphically express temporal properties through parametric diagrams of SysML. Moreover, TEPE is incorporated in SysML

					profile AVATAR which has rich capabilities to specify time-constraint properties for early verification. Furthermore, open source tool (TTool [117]) is customized to incorporate AVATAR-TEPE support for editing diagrams and their transformation to UPPAAL (up to some extent).
56 **2012**	Ning et al. [56]	This paper proposes novel methodology to formally specify and verify task-level time constraints.	Case study	General	This paper presents novel methodology to formally specify and verify task-level time constraints. It extends CCSL to include particular improvements for time constraints. TPN based verification method is develop for verification. Proposed methodology is incorporated in verification framework (Ge et al [120]).
57 **2011**	Razieh et al [57]	This paper proposes ExSAM profile for embedded systems.	Case studies	Modeling	This paper proposes ExSAM profile for embedded systems by integrating SYSML and AADL. The combination of SYSML and AADL leads to manage diverse modeling requirements for embedded systems.
58 **20**	Luciano et al [58]	This paper MADES approach for model verification and validation of embedded	Case studies	General	This paper present model verification and validation approach which is used in

1 **3**		systems			MADES project. Various notations of UML and MARTE profiles are combined together in a systematic way to specify the requirements of embedded systems. These notations are further used for formal verification of model. Simulation capabilities are also incorporated in MADES using Modelica.
59 **2** **0** **1** **2**	Xiaopu et al [59]	This paper proposes a novel methodology to improve the model verification by integrating SYSML state machine diagram and MARTE notations.	Case Study (TS)	General	This paper introduces extension of SYSML state machine diagram by integrating MARTE time notations to improve the modeling of system behavioral aspects. Furthermore, a novel algorithm is introduced to transform extended SYSML state machine diagram into timed automata. The model is then verified through Uppaal [123].
60 **2** **0** **1** **4**	Minh et al [60]	This paper introduces novel approach to model timing aspects of embedded systems by integrating UML diagrams with MARTE notations.	Example (pacemaker)	General	This paper proposes methodology to improve the modeling of embedded systems timing aspects by making use of UML sequence and state machine diagrams with MARTE notations. The TMC [124] tool is developed which can be used with papyrus and

					Visual paradigm [126] eclipse plug-ins.
61 **2012**	Erwan et al. [61]	This paper presents novel model verification approach by specifying safety critical systems requirements in model through SYSML, OCL and Alf.	Case study (Railway)	General	This paper first introduces SYSML block definition and state machine diagrams to specify requirements where OCL is used to specify constraints and Alf [127] is used to specify behavior. Thereafter, the developed model is transformed to B method to perform potential verification and validation activities through SYSML to B translator tool [125].

3.4 Results of Literature Review in the Context of MODEVES

The quantitative and qualitative analysis is performed on significant data extraction (Table II) to conclude the modeling (Table IV) and transformation (Table V) results. This leads us to conclude some important aspects of MODEVES project against each MBSE activity as follows:

Modeling Embedded Systems Requirements: It is analyzed that modeling activity is the core of MBSE approach because all other activities (i.e. model transformation, verification and simulation) are tightly coupled with this activity. Therefore, embedded systems requirements are modeled by considering various important verification and validation aspects. UML and its SYSML/MARTE profiles are frequently used in different combination to specify structural and behavior aspects because it is challenging to specify the complete requirements for complex systems through a single profile individually. On the other hand, there are certain issues while integrating UML and its SYSML/MARTE profiles [30][34]. Another big challenge is to select the appropriate property specification approach while specifying various properties and constraints in the models. MARTE profile provides rich support for timing constraints but that is not sufficient enough to specify all behavioral and structural aspects of embedded systems. Therefore, MARTE profile is rarely utilize alone (Table IV). It is mostly used in different combinations with UML and SYSML. Another interesting fact is that SYSML is the profile that provides good capabilities to model both structural and behavioral aspects alone. Block definition, parametric, activity and state machine diagrams of SYSML are frequently used in contemporary researches [3-7]. Finally, it can be concluded that the best approach to specify the wide-ranging requirements of complex and large embedded systems should be based on the systematic combination of UML and its SYSML/MARTE profiles. *Hence, we will develop the modeling methodology of MODEVES by combining different diagrams / notations of UML and its SYSML/MARTE profiles.*

Model Transformation: It is the key activity for further verification and validation of the system. It is used to perform various operations; however, the most important outcome of this activity is the executable source code in the target domain for validation (Simulation) of the system. Another important outcome of this activity is the transformation of given model into domain specific model for formal verification of the system. M2M and M2T transformations are usually used to attain the desired outcomes. M2M is more frequently used as compared to M2T (Table V) due to its accuracy of model transformation as M2M approach significantly reduces the transformation errors. On the other hand, M2T approach is highly flexible as it can easily be customized according to the target domain. Consequently, the simultaneous use of both M2M and M2T approaches is very common (Table V) for the development of large and complex embedded systems. *Hence, we will use both M2M and M2T approaches in MODEVES, however, exact decision will be made in latter stages of the project*

Model Verification: Model verification activity is performed to ensure the correctness of the given model. Model transformation is performed to convert the given model into specific model that support formal verification technique to verify the behavioral and temporal aspects of the model / system. However, the capability of formal verification approaches is questionable regarding the complete verification of all behavioral aspects of the given model. Finally, it can be concluded that model verification activity speedup and ease the development of embedded systems but it may not completely verify all behavioral aspects of the large and complex embedded systems. As the model verification in MODEVES is done through System Verilog Assertions (SVA), any other formal verification approach is optional for the project.

Simulation: Validation of the system is performed through simulation by making use of the generated source code. Researchers usually utilize available simulation tools for the validation of the system [4][25]. However, some researchers develop their own simulation mechanism / tool for validation [1]. Consequently, the most important activity of simulation is the selection or development of appropriate simulation tool because different simulation mechanism / tool are required for the simulation of source code of different languages. Another important aspect of the simulation activity is the knowledge of simulation environment / variable to perform accurate simulation. This normally requires specifying some simulation information within the developed models. Finally, it can be concluded that simulation of the model is highly dependent on the type of automatically generated executable source code. It may also require specifying some simulation aspects in the models. *As we intend to generate System Verilog source code along* *with assertions in our project (MODEVES), we will look for available simulation tool that support simulation of System Verilog Assertions and can easily be integrated into given working environment of our project.*

3.5 Limitation of Research

Although, we have completely followed the guidelines of SLR [64] and strictly observed the developed review protocol (Section 2.2), there are still certain limitations:

- We have used the appropriate search terms and thoroughly scanned the search results. However, few search terms returned thousands of results and cannot be scanned exhaustively. Furthermore, we have rejected number of researches on the basis of its title and there is a possibility that contents of the research are not properly defined in the title. Consequently, we do not claim the exhaustiveness of our research in this article.
- We have used four renowned scientific databases i.e. IEEE, ELSEIVER, ACM and SPRINGER. These databases provide large amount of journal and conference publications. However, a lot of research work is provided by other databases. Therefore, there is a fair chance that we have missed latest relevant researches from other databases. However, we believe that ultimate findings of this SLR are not affected much because selected scientific databases provide high quality latest research literature.

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$^{\text{Simple}}$-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