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

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Name of Tool / Framework</th>
<th>Corresponding MBSE Activities</th>
<th>Relevant Researches</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Topcased [62]</td>
<td>Modeling</td>
<td>[3][8][9][52]</td>
</tr>
<tr>
<td>2</td>
<td>Modelio Editor [68]</td>
<td>Modeling</td>
<td>[12]</td>
</tr>
<tr>
<td>4</td>
<td>Eclipse GEF [89]</td>
<td>Modeling</td>
<td>[17]</td>
</tr>
<tr>
<td>5</td>
<td>Rhapsody [132]</td>
<td>Modeling</td>
<td>[4][19][22][57]</td>
</tr>
<tr>
<td>6</td>
<td>PapyrusMDT [98]</td>
<td>Modeling</td>
<td>[21][44][60]</td>
</tr>
<tr>
<td>7</td>
<td>EA MDG [85]</td>
<td>Modeling</td>
<td>[27]</td>
</tr>
<tr>
<td>8</td>
<td>Visual paradigm [126]</td>
<td>Modeling</td>
<td>[60]</td>
</tr>
<tr>
<td>10</td>
<td>ATL [63]</td>
<td>Model Transformation</td>
<td>[3][8][39][44][46][50][59]</td>
</tr>
<tr>
<td>12</td>
<td>Acceleo [67]</td>
<td>Model Transformation</td>
<td>[9][21][42][44]</td>
</tr>
<tr>
<td>15</td>
<td>Apache Velocity [86]</td>
<td>Model Transformation</td>
<td>[16][18]</td>
</tr>
<tr>
<td>16</td>
<td>Eclipse EMF</td>
<td>Model</td>
<td>[21][41][37]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transformation</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>GenERTiCA tool [94]</td>
<td>Model Transformation [18]</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>MDworkbench [95]</td>
<td>Model Transformation [19][22]</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>CatapultC [116]</td>
<td>Model Transformation [22]</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>QVTO [117]</td>
<td>Model Transformation [14]</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>AGG [109]</td>
<td>Model Transformation [38][39][40]</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>MOFScript [112]</td>
<td>Model Transformation [29][39][42][43]</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>MeTHAGeM [50]</td>
<td>Model Transformation [50]</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Epsilon [133]</td>
<td>Model Transformation [12][58]</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>TMC [124]</td>
<td>Model Transformation [60]</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>SYSML to B translator tool [125]</td>
<td>Model Transformation [61]</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>PRISM [66]</td>
<td>Model Verification [7][52]</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>UPPAAL [123]</td>
<td>Model Verification [55][59]</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>SIMULINK [65]</td>
<td>Simulation [4][25]</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>ActiveHDL [129]</td>
<td>Simulation [14]</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>QEMU [130]</td>
<td>Simulation [15]</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>GTKWave [72]</td>
<td>Simulation [36]</td>
<td></td>
</tr>
</tbody>
</table>
| 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] |
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

<table>
<thead>
<tr>
<th></th>
<th>Tool</th>
<th>MBSE Activities</th>
<th>Reference</th>
</tr>
</thead>
</table>
| 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] |
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

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Tool Name</th>
<th>General Characteristics</th>
<th>Activity Specific Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>License Type</td>
<td>Customizability</td>
</tr>
<tr>
<td>1</td>
<td>Magic Draw [131]</td>
<td>Proprietary (No Magic)</td>
<td>Partial</td>
</tr>
<tr>
<td>2</td>
<td>Modelio Editor [68]</td>
<td>GPL</td>
<td>Full</td>
</tr>
<tr>
<td>3</td>
<td>Rhapsody [132]</td>
<td>Proprietary (IBM)</td>
<td>Partial</td>
</tr>
<tr>
<td>4</td>
<td>PapyrusMDT [98]</td>
<td>EPL</td>
<td>Full</td>
</tr>
<tr>
<td>5</td>
<td>Artisan Studio [114]</td>
<td>Proprietary (Atego)</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>Visual paradigm [126]</td>
<td>Proprietary (Visual paradigm)</td>
<td>No</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Tool Name</th>
<th>General Characteristics</th>
<th>Activity Specific Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>License Type</td>
<td>Customizability</td>
</tr>
<tr>
<td>1</td>
<td>MediniQVT [121]</td>
<td>EPL</td>
<td>Full</td>
</tr>
<tr>
<td>2</td>
<td>ATL [63]</td>
<td>EPL</td>
<td>Full</td>
</tr>
<tr>
<td>3</td>
<td>Xpand [97]</td>
<td>EPL</td>
<td>Full</td>
</tr>
<tr>
<td>4</td>
<td>Acceleo [67]</td>
<td>EPL</td>
<td>Full</td>
</tr>
<tr>
<td>5</td>
<td>Apache</td>
<td>ASL</td>
<td>Full</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Tool Name</th>
<th>General Characteristics</th>
<th>Activity Specific Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>License Type</td>
<td>Customizability</td>
<td>Supported Platforms</td>
</tr>
<tr>
<td>1</td>
<td>PRISM [66]</td>
<td>GPL</td>
<td>Full</td>
</tr>
<tr>
<td>2</td>
<td>Zot [118]</td>
<td>GPL</td>
<td>Full</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Tool Name</th>
<th>License Type</th>
<th>Customizability</th>
<th>Supported Platforms</th>
<th>Integration Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIMULINK [65]</td>
<td>Proprietary (Math Works)</td>
<td>Partial</td>
<td>Windows Linux</td>
<td>High</td>
</tr>
<tr>
<td>2</td>
<td>ActiveHDL [129]</td>
<td>Proprietary (AlDEC)</td>
<td>Partial</td>
<td>Windows</td>
<td>Low</td>
</tr>
<tr>
<td>3</td>
<td>QEMU [130]</td>
<td>GPL</td>
<td>Full</td>
<td>Windows Linux</td>
<td>High</td>
</tr>
<tr>
<td>4</td>
<td>Catapult C [116]</td>
<td>Proprietary (Calypto)</td>
<td>Partial</td>
<td>Windows Linux</td>
<td>Low</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Name</th>
<th>License Type</th>
<th>Supported Platforms</th>
<th>Activity Specific characteristics Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EMF</td>
<td>EPL</td>
<td>Windows</td>
<td>Framework</td>
</tr>
</tbody>
</table>
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 Graphics Questa Sim to be used in our project.
References


[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


[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/DAT.2009.5090662


[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


[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


[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


[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


[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


[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


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


http://www.mentor.com/products/fv/questa/


https://projects.eclipse.org/projects/modeling.emft.henshin/


http://www.topcased.org/index.php/content/view/32/49/