*Science World Journal Vol 13(No 4) 2018*
*www.scienceworldjournal.org*
*ISSN 1597-6343*
*Published by Faculty of Science, Kaduna State University*

**Full Length Research Article**

# AUTOMATIC CODE GENERATION FROM UML DIAGRAMS: THE STATE-OF-THE-ART

[1]Maryam I. Mukhtar, [2]Bashir S. Galadanci

[1,2]Department of Software Engineering, Bayero University, Kano.

Authors Email Addresses: [1]mimukhtar.se@buk.edu.ng, [2]bashirgaladanci@yahoo.com

## ABSTRACT

The emergence of the Unified Modeling Language (UML) as the de-facto standard for modeling software systems has encouraged the development of automated software tools that facilitate automatic code generation. UML diagrams are used to diagrammatically model and specify the static structure as well as the dynamic behavior of object-oriented systems and the software tools then go ahead and automatically produce code from the given diagrams. In the last two decades substantial work has been done in this area of automatic code generation. This paper is aimed at identifying and classifying this work pertaining to automatic code generation from UML diagrams, restricting the search neither to a specific context nor to a particular programming language. A Systematic literature review (SLR) using the keywords “automatic code generation”, “MDE”, “code generation” and “UML” is used to identify 40 research papers published during the years 2000–2016 which are broadly classified into three groups: Approaches, Frameworks and Tools. For each paper, an analysis is made of the achievements and the gaps, the UML diagrams used the programming languages and the platform. This analysis helps to answer the main questions that the paper addresses including what techniques or implementation methods have been used for automatic code generation from UML Diagrams, what are the achievements and gaps in the field of automatic code generation from UML diagrams, which UML diagram is most used for automatic code generation from UML diagrams, which programming language source code is mostly automatically generated from the design models and which is the most used target platform? The answers provided in this paper will assist researchers, practitioners and developers to know the current state-of-the-art in automatic code generation from UML diagrams.

**Keywords:** Automatic Code Generation (ACG); Unified Modeling Language (UML); Model Driven Engineering (MDE)

## INTRODUCTION

Model Driven Engineering (MDE) was introduced as an engineering approach that uses models directly as primary artifacts throughout the software development life cycle (SDLC) rather than using source code (Klein *et al.*,2015) The best known MDE initiative is Model Driven Architecture (MDA) which was proposed by the Object Oriented Group (OMG) in 2001 as a new software development paradigm (Summerville,2009) that uses a subset of UML diagrams such as class diagrams, sequence diagrams and state diagrams. It was started because there was the necessity to decrease development efforts, create and use analysis and design models at each stage of the software development process and automate code generation from models (Eveleens & Verhoef, 2009). Automatic code generation has gained a lot of attention in Software Engineering because it has many benefits including reuse, being less error prone than writing code manually, maintainability and accuracy. Furthermore, the advantages of high-level modeling and analysis are significantly enhanced if code can be generated automatically from a model such that the correspondence between the model and code is precisely understood (Niaz, 2005). While automatic code generation appears to be more popular than visual programming, it is interesting to note that they have similar objectives of reducing the learning and production effort of programmers and software developers.

The need of automated tools to generate source code from visual models has been increasing over time and a lot of work has been done on the automatic transformation of UML models to source code (Thongmak & Muenchaisri, 2002). Taking this into account, the objective of this systematic literature review (SLR) is to identify, review and analyze the research done in the field of automatic code generation from UML diagrams.

### Related Work

Numerous studies such as those of (Loniewski *et.al.*,2010; Nguyen *et.al.*,2013; Nguyen *et.al.*,2015; Graciano *et.al.*,2014; Queiroz & Braga,2014; Heineck *et.al.*,2016; Boussaïd *et.al.,*2017; Essaadi *et.al.,*2017*)* have concentrated on Model Driven Engineering as a whole. Boussaïd *et.al.*(2017) conducted a survey on search-based techniques and model-driven engineering and found out that among the trends that focus on managing software complexity through automation, Search based Software Engineering has gained major recognition and has been applied successfully in many areas; Queiroz & Braga(2014) did a survey on the “Development of Critical Embedded Systems Using Model-driven and Product Lines Techniques” and found out that, until recently, there were limited studies that develop safety critical embedded systems using Model Driven Engineering; Loniewski *et.al.*(2010) conducted a “Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development” and found out that at the requirements level most Model Driven Development approaches use only partially defined requirements models or even natural language; Graciano *et.al.*(2014) conducted a systematic literature review of the state-of-the-art in the field of Model Driven Development for System of Systems (SoS) and found out that Model Based Software Systems have not been sufficiently used for modeling and synthesis of System of Systems; Essaadi *et.al.*(2017) proposed a systematic mapping study of Wireless Sensor Networks(WSNs) MDE-based modeling languages and revealed that few studies take into account rigorous and exhaustive verification in analysis; Heineck *et.al.*(2016) conducted a systematic literature review on

Model Driven Development in Robotics Domain approaches and proved that the existing approaches mostly use a component-based development paradigm to provide interesting capabilities.; Nguyen *et.al.*(2013) carried out a systematic review to identify, classify and evaluate different Model Driven Security approaches and their results revealed that most approaches focus on authorization and confidentiality with a few focusing on security concerns like integrity, availability, and authentication; Nguyen *et.al.*(2015) conducted an "Extensive Systematic Review on Model-Driven Development of Secure Systems" and found out that developing domain-specific languages are a major factor in Model Driven Security approaches.

Others such as those of (Dias *et.al.*,2007; Mohi-aldeen *et.al.*,2014; Maheshwari & Prasanna,2015; Salman & Hashim,2016; Shah *et.al.*,2016) have focused on automatic test case generation; Shah *et.al.*(2016) carried out a review on various approaches that use class diagrams to support test case generation activities. Their results showed that while 76% of the techniques used intermediate forms for generating test cases, 23% of them did not. So also 65% of the studies developed a tool for their frameworks but 35% did not. Maheshwari & Prasanna (2015) did a review of literature on automatic test case generation and found out that the optimization approach gave the most efficient test suite for the given problem model; Dias *et.al.*(2007) conducted a systematic review performed on model-based testing (MBT) approaches. Their results showed that MBT approaches are rarely integrated in software development processes and are not evaluated empirically and/or not transferred to the industrial environment; Salman & Hashim (2016) did a review on Automatic Test Case Generation from UML State Chart Diagrams. Their results revealed that most of the works on state chart test case generation methods are on the Depth First Search DFS algorithm and that some of the test case generation do not work for maximum test coverage, with some producing significant numbers of tests with less test coverage; Mohi-aldeen *et.al.*(2014) carried out a "Systematic Mapping Study in Automatic Test Case" and found out that most researchers focus on code coverage, the Genetic Algorithm (GA) had the most published studies and research interest in reduction of test case generation topped the list as the dominant research concern.

A few studies such as those of (Mehmood & Jawawi, 2013; Rashid *et.al.* 2015; Rosales-Morales, *et.al.* 2015; Rashid &Anwar, 2016; Syriani *et.al.* 2018) have focused on areas relating to automatic code generation; Rosales-Morales, *et.al.* (2015) carried out an analysis and evaluation of CASE tools, IDEs tools and frameworks for automated software development and automatic code generation in order to determine whether they met a set of quality metrics. Their results revealed that most tools provided a user friendly interface, academic tools were mostly used as prototypes unlike commercial tools that supported modularity principles and functions with well- defined purposes, and almost all automatic code generation tools provided good documentation for users; Rashid &Anwar(2016)carried out a "Systematic Investigation of Tools in Model Based System Engineering for Embedded Systems" and their results revealed that Eclipse was the leading platform, providing a variety of MBSE tools for embedded systems while Topcased, Rhapsody, Papyrus and Magic Draw were the most popular and frequently used modelling tools and SIMULINK was the leading simulation tool; Mehmood & Jawawi (2013) conducted a review on Aspect-oriented Model Driven code generation. Their results showed that the most examined aspect-oriented model driven code generation topics are modeling notations and process, model composition and interaction management, aspect-oriented code generation, code generation from specification of non-functional requirements, and applicability of aspect-oriented modeling and code generation approaches. So also 52% of the research on Aspect-oriented Model Driven code generation are in conferences proceedings, 21% in workshop and 27 % in journals. Syriani *et.al. (*2018) carried out a review on Template-based Code Generation (TBGC) which is a technique that produces code from high-level specifications. Their results showed that TBGC had greatly benefited from MDE and is being used for code generation in a variety of domains for almost 15 years now and that both MDE and non-MDE tools were becoming current development resources/tools in industry; Rashid *et.al.(*2015) also did a systematic review on tools selection in Model Based System Engineering (MBSE) for Embedded Systems and found out that UML and its SYSML/MARTE construct were mostly used for specifying structural and behavioral aspects of a system; Bajovs *et.al.*(2013)gave an overview of the state-of-the-art and the practical implications of code generation from UML Models. From their review it was realized that there were some good quality MDA based systems; however, there were not many of them all over the world. This means to date majority of the programming is done by human specialists rather than code generators and even though there were tools that could convert from design models to code, those tools were rarely used in the industry. This is probably as a result of the tools being too complex to use and having no documentation about the platforms they could work on.

The research reported in Domí *et.al.*(*2012*) conducted a systematic literature review of code generation proposals from state machines and found out that the State Design pattern was the most used for implementing state machine specifications, the elements of state machines considered by most proposals included context class, the current state, the state-transition process, guards, simple states, events, and actions and only a few papers had considered qualitative aspects in Software Engineering in their implementation strategies.

From the review, it is clear that there is a dearth of literature regarding papers that have tried to analyze the work done so far on automatic code generation from UML Diagrams. This review seeks to fill this void.

## METHODOLOGY

A systematic literature review has been used to carry out this research (Kitchenham, 2004). A systematic review is a type of literature review that collects and critically analyzes multiple research studies or papers in a particular subject of interest. The research aims to address the following questions.

RQ1: What techniques or implementation methods have been used for automatic code generation from UML diagrams?

RQ2: What are the achievements and gaps in the field of automatic code generation from UML diagrams?

RQ3: Which UML diagram is the most used for automatic code generation?

RQ4: Which programming language source code is mostly automatically generated from the design models?

RQ5: Which is the most used target platform?

The SLR has been performed systematically to provide an answer to each of the given research questions. Taking into account RQ1, we have analyzed the general context of each paper in terms of the implementation technique it proposes. We have classified these papers depending on whether they propose approaches, frameworks or tools. Considering RQ2, we have analyzed what achievements have been made in each paper and what gaps are remaining. Regarding question RQ3 and RQ4, we have analyzed which UML diagrams (Class Diagrams, State Charts, Sequence Diagrams, Activity Diagrams, Others) are used in each paper and what type of code (Java/Java Android, C/Objective C/ System C/C++/ C#, Unspecified, Others). Considering which platform, we have classified them into four namely Object Oriented Platforms, Embedded Platforms, More Than One Platform and Unspecified Platform. The methodology depicted in Figure 1 was followed during this SLR in order to answer the research questions.

This section explains the different stages we considered to undertake this review.

**Classification Definition**: We have defined three classification categories in order to organize the selected researches. These classifications enabled us to easily provide answers to our research questions. The three classifications are:
Approaches: This comprises of strategies that have been proposed to tackle automatic code generation from UML diagrams.
Frameworks: This consists of frameworks that are used for automatic code generation from UML diagrams.
Tools: This entails software tools that perform automatic code generation from UML diagrams.

**Search Keywords**:
Two sets of search keywords were used in this SLR. They are:
1. "Automatic Code Generation", "Model Driven Engineering"
2. "Code Generation", "UML"
The search was made using the Google Scholar academic database. Papers retrieved included those that were published in IEEE, Cite Seer, ACM, Elsevier Science Direct, Research Gate, ProQuest and some other publishers.

**Inclusion and Exclusion Criteria**
Starting from an initial list of 1980 papers, some papers were excluded based on the criteria explained below:
1. Relevance: Select the research only if it is relevant to one of the three predefined classifications (Section 2.1). Reject all other researches that do not belong to any of the predefined categories.
2. 2000–2016: Selected research work must be published from 2000 to 2016. Reject all researches that are published before 2000.
3. Type: All papers that are published in journal articles and conference proceedings are included. Reject all other researches that are in books, theses, book series and web sites.
4. Code generation: Selected research work must offer code generation, implementation of pseudo code or code schemas from UML diagrams. Reject all other researches that generate other things (not codes) from the UML diagrams.

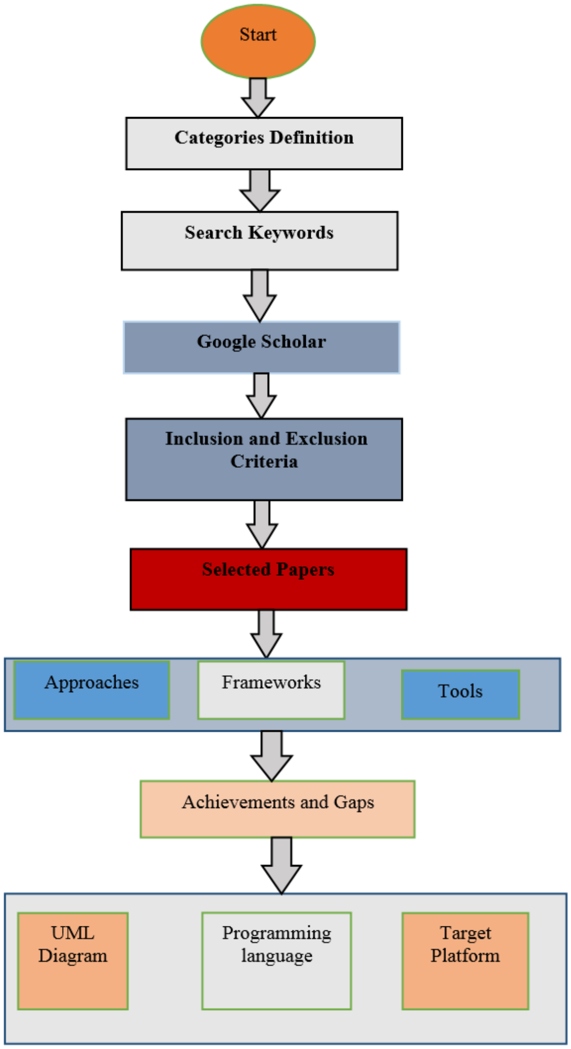
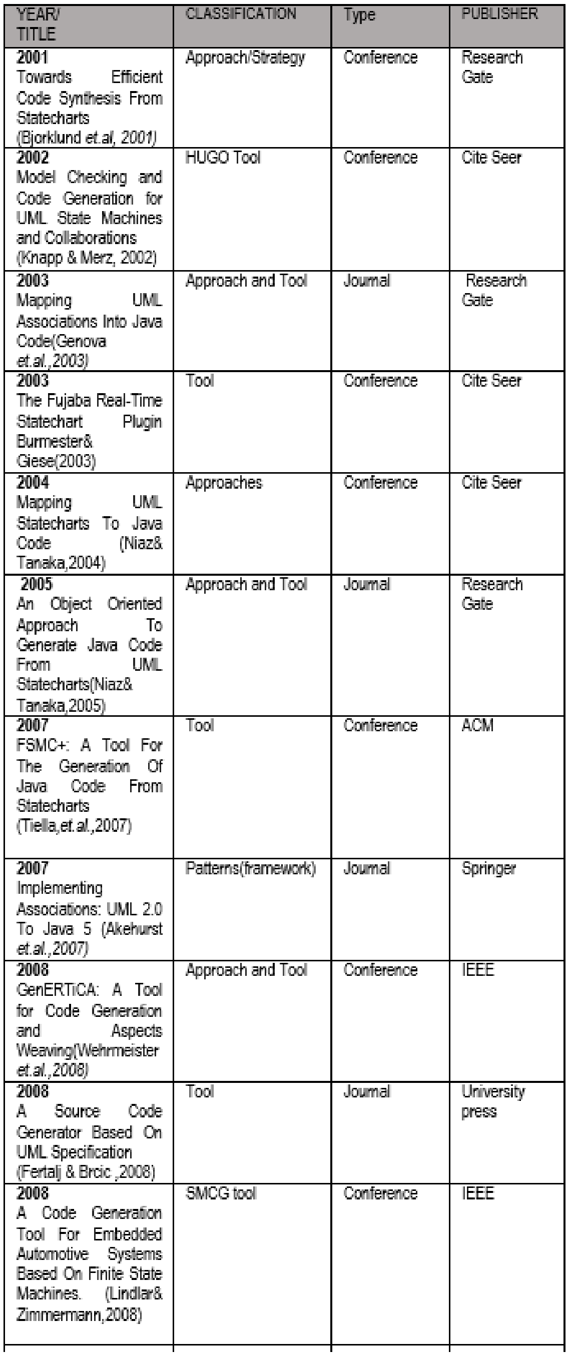


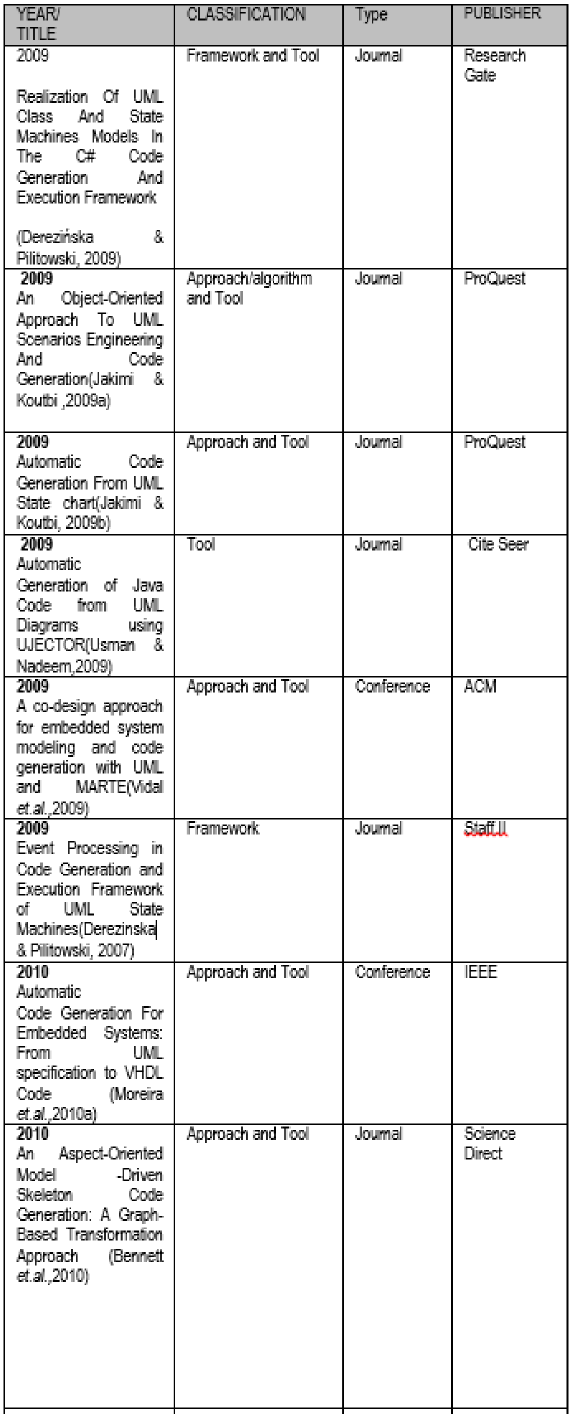
**Figure 1:** Research Methodology

## RESULTS

The SLR yielded 40 papers which were broken down into approaches, developed tools and frameworks (see Table 1).

**Table 1: Classification of selected papers**





| YEAR/ TITLE | CLASSIFICATION | Type | PUBLISHER |
|---|---|---|---|
| **2001**<br>Towards Efficient Code Synthesis From Statecharts (Bjorklund *et.al, 2001)* | Approach/Strategy | Conference | Research Gate |
| **2002**<br>Model Checking and Code Generation for UML State Machines and Collaborations (Knapp & Merz, 2002) | HUGO Tool | Conference | Cite Seer |
| **2003**<br>Mapping UML Associations Into Java Code(Genova *et.al.,2003)* | Approach and Tool | Journal | Research Gate |
| **2003**<br>The Fujaba Real-Time Statechart Plugin Burmester& Giese(2003) | Tool | Conference | Cite Seer |
| **2004**<br>Mapping UML Statecharts To Java Code (Niaz& Tanaka,2004) | Approaches | Conference | Cite Seer |
| **2005**<br>An Object Oriented Approach To Generate Java Code From UML Statecharts(Niaz& Tanaka,2005) | Approach and Tool | Journal | Research Gate |
| **2007**<br>FSMC+: A Tool For The Generation Of Java Code From Statecharts (Tiella,*et.al.*,2007) | Tool | Conference | ACM |
| **2007**<br>Implementing Associations: UML 2.0 To Java 5 (Akehurst *et.al.,2007)* | Patterns(framework) | Journal | Springer |
| **2008**<br>GenERTiCA: A Tool for Code Generation and Aspects Weaving(Wehrmeister *et.al.,2008)* | Approach and Tool | Conference | IEEE |
| **2008**<br>A Source Code Generator Based On UML Specification (Fertalj & Brcic ,2008) | Tool | Journal | University press |
| **2008**<br>A Code Generation Tool For Embedded Automotive Systems Based On Finite State Machines. (Lindlar& Zimmermann,2008) | SMCG tool | Conference | IEEE |
| 2009<br>Realization Of UML Class And State Machines Models In The C# Code Generation And Execution Framework<br>(Derezińska & Pilitowski, 2009) | Framework and Tool | Journal | Research Gate |
| **2009**<br>An Object-Oriented Approach To UML Scenarios Engineering And Code Generation(Jakimi & Koutbi ,2009a) | Approach/algorithm and Tool | Journal | ProQuest |
| **2009**<br>Automatic Code Generation From UML State chart(Jakimi & Koutbi, 2009b) | Approach and Tool | Journal | ProQuest |
| **2009**<br>Automatic Generation of Java Code from UML Diagrams using UJECTOR(Usman & Nadeem,2009) | Tool | Journal | Cite Seer |
| **2009**<br>A co-design approach for embedded system modeling and code generation with UML and MARTE(Vidal *et.al.*,2009) | Approach and Tool | Conference | ACM |
| **2009**<br>Event Processing in Code Generation and Execution Framework of UML State Machines(Derezinska & Pilitowski, 2007) | Framework | Journal | Staff.II |
| **2010**<br>Automatic Code Generation For Embedded Systems: From UML specification to VHDL Code (Moreira *et.al.*,2010a) | Approach and Tool | Conference | IEEE |
| **2010**<br>An Aspect-Oriented Model -Driven Skeleton Code Generation: A Graph-Based Transformation Approach (Bennett *et.al.*,2010) | Approach and Tool | Journal | Science Direct |

| YEAR/ TITLE | CLASSIFICATION | Type | PUBLISHER |
|---|---|---|---|
| **2010**<br>Automatic SystemC Code Generation From UML Models At Early Stages Of Systems On Chip Design. (Boutekkouk, 2010) | Approach | Journal | Cite Seer |
| **2010**<br>Tool Support for Code Generation from a UMLsec Property Montrieux *et.al.*,2010) | Tool | Conference | ACM |
| **2010**<br>Generating VHDL Source Code from UML models of Embedded System(GenERTiCA) Moreira *et.al.*,2010b) | Approach and Tool | Journal | |
| **2011**<br>A Platform Independent Framework For Statecharts Code Generation(Andolfato *et.al.*,2011) | Framework and Tool | Conference | accelconf |
| **2011**<br>SystemC Code Generation From UML For Wireless Sensor Networks Design (Villa *et.al.*,2011) | Approach | Conference | Cite Seer |
| **2011**<br>Generating Java Code From UML Class And Sequence Diagrams (Parada *et.al.*,2011) | Approach and Tool | Conference | IEEE |
| **2012**<br>Generating Methods Signatures From Transition State Diagram: A Model Transformation Approach(ElBeggar *et.al.*,2012) | Approach | Conference | IEEE |
| **2012**<br>Full Code Generation From UML Model For Complex Embedded Systems (Ciccozzi *et.al.*, 2012) | Approach | Conference | Research Gate |
| **2012**<br>Enhancing The Generation Of Correct-By-Construction Code From Design Models For Complex Embedded Systems(Ciccozzi & Sjödin, 2012) | Approach | Conference | Research Gate |
| **2012**<br>Model Checking and Code Generation for UML Diagrams Using Graph Transformation(Chama *et.al.*, 2012) | Approach and Tool | Journal | ProQuest |
| **2012**<br>Automatic Code Generation By Model Transformation From Sequence Diagram Of System's Internal Behavior (Omar *et.al.*, 2012) | Approach | Journal | Research Gate |
| **2012**<br>A Simple Implementation Of UML Sequence Diagram To Code Generation Through XMI Representation (Vadakkumcheril *et.al.*, 2013) | Tool | Journal | Cite Seer |
| **2013**<br>A Framework For Model-Based Code Generation From A Flowchart (Hussein & Salah,2013) | Framework | Conference | Cite Seer |
| **2013**<br>Generation Of Efficient Embedded C Code From UML/MARTE Models (Lennis & Aedo,2013) | Approach | Conference | ProQuest |
| **2013**<br>An MDE Approach For Automatic Code Generation From UML/MARTE To OpenCL (Rodrigues *et.al.*, 2013) | Approach | Journal | IEEE |
| **2013**<br>Automatic Code Generation From Unified modelling language Sequence Diagrams (Kundu *et.al.*, 2013) | Approach | Journal | IEEE |
| **2014**<br>An Approach to Code Generation from UML Diagrams (Gurad & Mahalle, 2014) | Approach | Journal | IJESRT |
| **2014**<br>A Model Transformation Approach for Code Generation From State Machine Diagram (Bousetta *et.al.*, 2014) | Approach | Journal | Research Gate |
| **2015**<br>HDL Code Generation from UML/MARTE Sequence Diagrams for Verifications and Synthesis (Ebeid *et.al.*, 2015) | Approach | Journal | Springer |
| **2015**<br>Software Modelling And Automatic Code Generation Based On Reactive State Diagram (Qu *et.al.*, 2015) | Framework | Conference | Atlantis-Press |
| **2015**<br>Automating Mobile Application Development: UML-Based Code Generation For Android And Windows Phone (Parada *et.al.*, 2015) | Approach and Tool | Journal | Cite Seer |

The classification of the papers showed that majority of the papers for automatic code generation from UML diagrams proposed approaches and tools to implement the approaches. So also majority of papers selected were published in 2009, 2010 and 2012. (See Figure 2 and Figure 3)

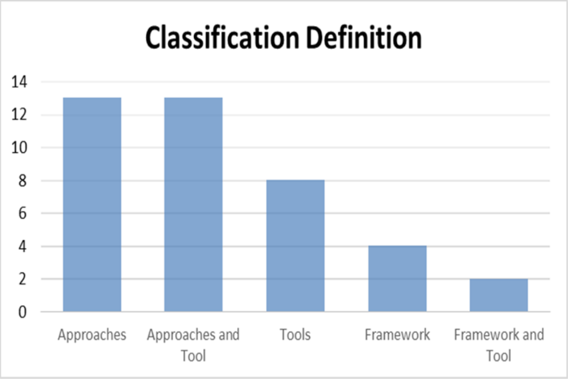


**Figure 2:** A chart showing the number of approaches, tools and frameworks

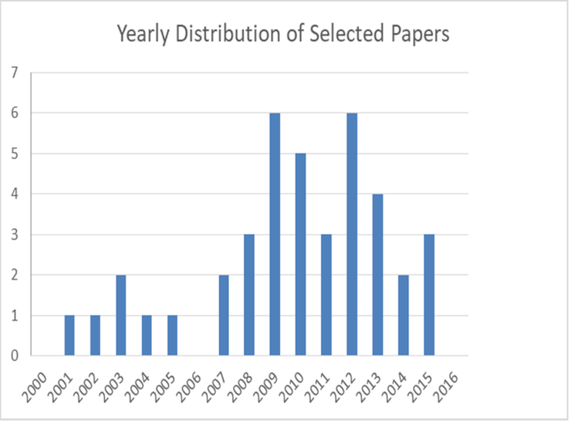
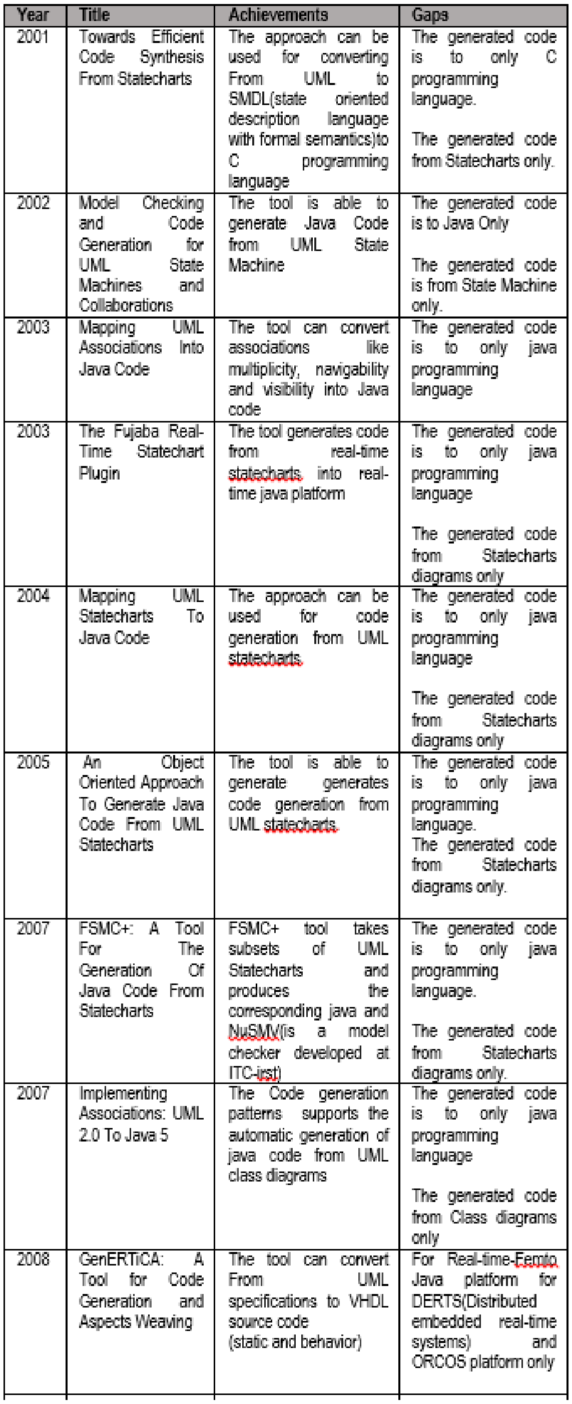


**Figure 3**: A chart showing yearly distribution of papers

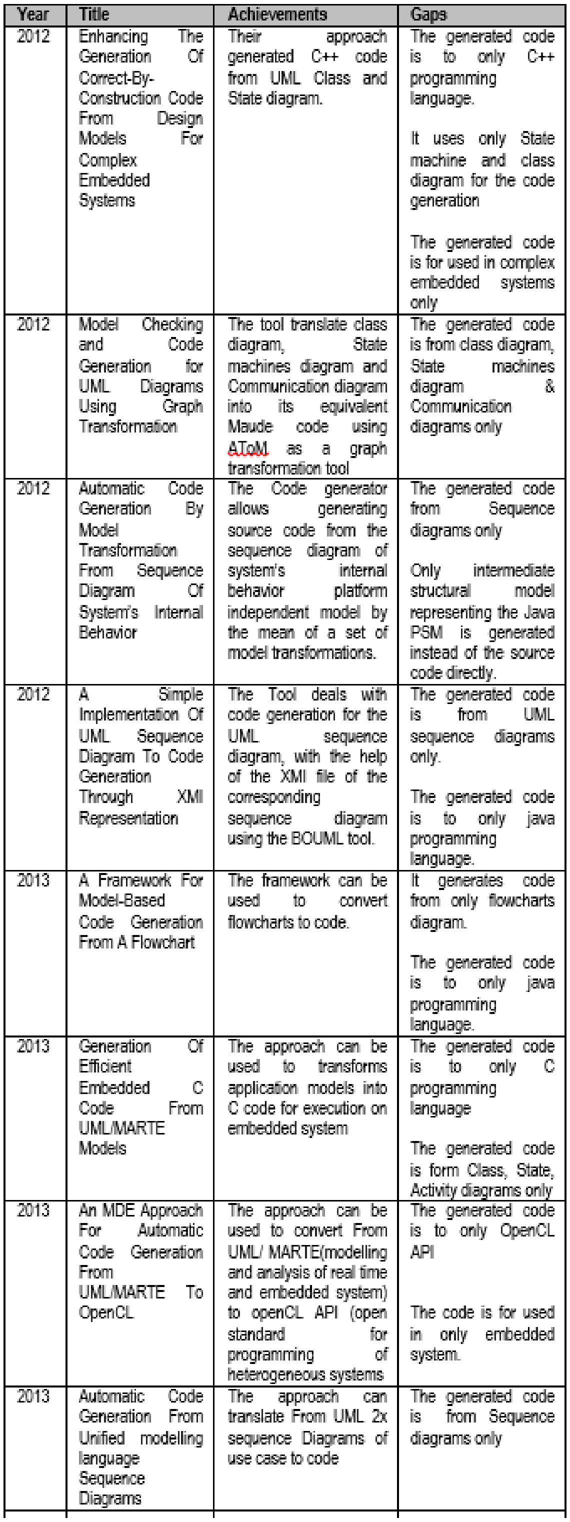
Each paper was analyzed in terms of its achievements and gaps. Table 2 gives a summary of the achievements and gaps of each paper.

**Table 1**: Summary of achievements and Gaps

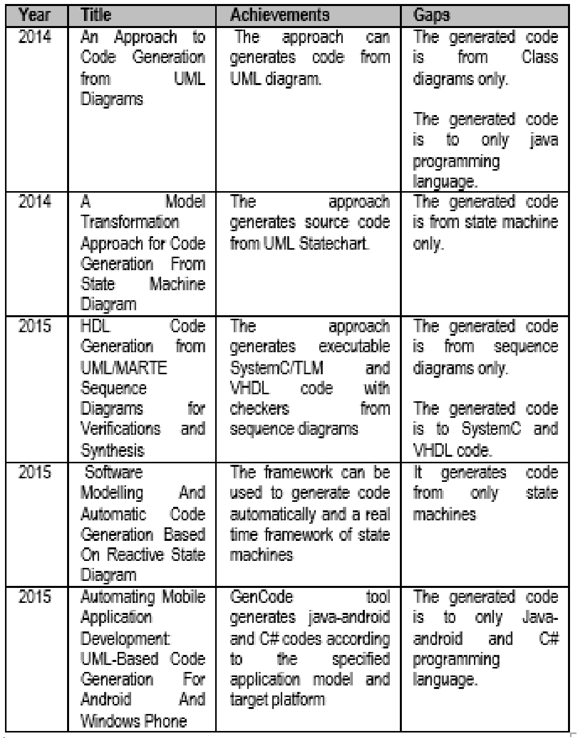


| Year | Title | Achievements | Gaps |
|---|---|---|---|
| 2001 | Towards Efficient Code Synthesis From Statecharts | The approach can be used for converting From UML to SMDL(state oriented description language with formal semantics)to C programming language | The generated code is to only C programming language.<br>The generated code from Statecharts only. |
| 2002 | Model Checking and Code Generation for UML State Machines and Collaborations | The tool is able to generate Java Code from UML State Machine | The generated code is to Java Only<br>The generated code is from State Machine only. |
| 2003 | Mapping UML Associations Into Java Code | The tool can convert associations like multiplicity, navigability and visibility into Java code | The generated code is to only java programming language |
| 2003 | The Fujaba Real-Time Statechart Plugin | The tool generates code from real-time statecharts into real-time java platform | The generated code is to only java programming language<br>The generated code from Statecharts diagrams only |
| 2004 | Mapping UML Statecharts To Java Code | The approach can be used for code generation from UML statecharts. | The generated code is to only java programming language<br>The generated code from Statecharts diagrams only |
| 2005 | An Object Oriented Approach To Generate Java Code From UML Statecharts | The tool is able to generate generates code generation from UML statecharts. | The generated code is to only java programming language.<br>The generated code from Statecharts diagrams only. |
| 2007 | FSMC+: A Tool For The Generation Of Java Code From Statecharts | FSMC+ tool takes subsets of UML Statecharts and produces the corresponding java and NuSMV(is a model checker developed at ITC-irst) | The generated code is to only java programming language.<br>The generated code from Statecharts diagrams only. |
| 2007 | Implementing Associations: UML 2.0 To Java 5 | The Code generation patterns supports the automatic generation of java code from UML class diagrams | The generated code is to only java programming language<br>The generated code from Class diagrams only |
| 2008 | GenERTiCA: A Tool for Code Generation and Aspects Weaving | The tool can convert From UML specifications to VHDL source code (static and behavior) | For Real-time-Femto Java platform for DERTS(Distributed embedded real-time systems) and ORCOS platform only |

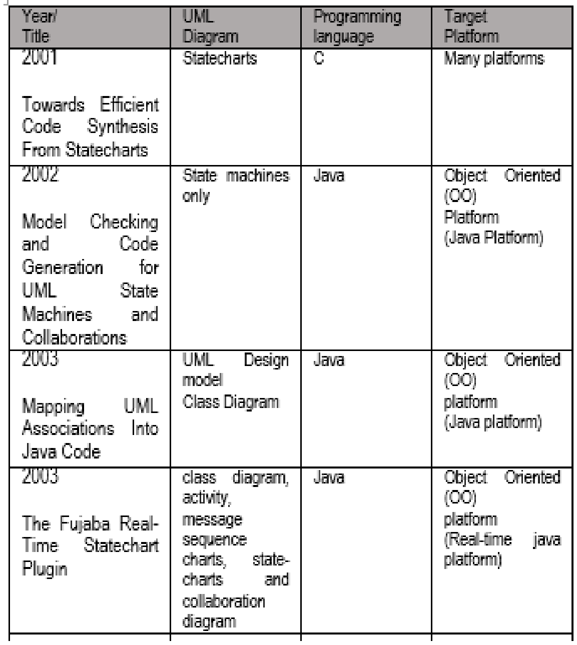
| Year | Title | Achievements | Gaps |
|---|---|---|---|
| 2008 | A Source Code Generator Based On UML Specification | The tool is capable of generating the source code in various programming languages from the same specification. | The tool does not work on different platforms |
| 2008 | A Code Generation Tool For Embedded Automotive Systems Based On Finite State Machines. | The tool convert state machines to code. | The generated code is for used in only embedded automotive systems<br>The generated code from State Machines only |
| 2009 | Realization Of UML Class And State Machines Models In The C# Code Generation And Execution Framework | The tool transforms UML models into C# source code and supports execution of the application reflecting the behavioral model. | The generated code is to only C programming language<br>The generated code from class and state machines diagrams only |
| 2009 | An Object-Oriented Approach To UML Scenarios Engineering And Code Generation | The tool automatically generate Implementation code from the UML Sequence Diagrams in an object-oriented programming language such as Java. | The generated code is to only object oriented programming languages like java.<br>The generated code from Sequence diagrams only. |
| 2009 | Automatic Code Generation From UML State chart | The tool generates low-level Java code directly from multiple UML statecharts. | The generated code is to only java programming language<br>The generated code from Statecharts only |
| 2009 | Automatic Generation of Java Code from UML Diagrams using UJECTOR | The tool converts UML Class , sequence and activity diagrams to java code | The generated code is to only java programming language<br>The generated code from Class , sequence and activity diagrams only |
| 2009 | A co-design approach for embedded system modeling and code generation with UML and MARTE | The tool converts UML Model to VHDL code | The generated code is to only VHDL code |
| 2009 | Event Processing in Code Generation and Execution Framework of UML State Machines | Their framework support converting of UML class and state machine. | The generated code is only to c# programming.<br>The generated code is from Class and State machine only. |
| 2010 | Automatic Code Generation For Embedded Systems: From UML specification to VHDL Code | The tool converts UML specifications to VHDL source code | The generated code is to only VHDL code<br>The generated code is for used in only embedded systems |
| 2010 | An Aspect-Oriented Model -Driven Skeleton Code Generation: A Graph-Based Transformation Approach | The tool translate extended UML Class Diagram to Java and AspectJ code | The generated code is to only AspectJ and Java code<br>The generated code from class diagrams only |
| 2010 | Automatic SystemC Code Generation From UML Models At Early Stages Of Systems On Chip Design | The approach can be used for automatic SystemC code generation from UML models at early stages of Systems On Chip (SOC) design | The generated code is to only SystemC programming language<br>The generated code is for used in only SOC design<br>The generated code is from only Sequence diagrams and activity diagrams |
| 2010 | Tool Support for Code Generation from a UMLsec Property | The tool can generate code from UML model activity diagram. | The generated code is to only java and AspectJ code. |
| 2010 | Generating VHDL Source Code from UML models of Embedded System(GenERTiCA | The tool translates from UML specifications to VHDL source code. | The generated code is VHDL code which is for used in only embedded systems. |
| 2011 | A Platform Independent Framework For Statecharts Code Generation | The tool translates UML model to SCXML and to Java application. | The generated code is to only Java programming language |
| 2011 | SystemC Code Generation From UML For Wireless Sensor Networks Design | Their approach can be used for converting UML models to automatic SystemC executable models | The generated code is to only SystemC.<br>-It generates code from package diagram, class diagram, composite structure diagram, state diagram, activity diagrams. |
| 2011 | Generating Java Code From UML Class And Sequence Diagrams | GenCode tool convert UML Class and sequence to structural and behavioral code | The generated code is to only Java programming language<br>It generates the code from only class and sequence diagrams. |
| 2012 | Generating Methods Signatures From Transition State Diagram: A Model Transformation Approach | The approach can be used diagram to generate intermediate structural model for the java platform for system's complex classes From Domain class diagram and transition state | It uses only class diagram and transition state diagram for code generation.<br>The generated code is to only Java programming language. |
| 2012 | Full Code Generation From UML Model For Complex Embedded Systems | The approach can be used to generates C++ structural and behavioral code from design models | The generated code is to only C++ programming language.<br>It uses only State machine and class diagram for the code generation<br>The generated code is for used in complex embedded systems only |



| Year | Title | Achievements | Gaps |
|---|---|---|---|
| 2012 | Enhancing The Generation Of Correct-By-Construction Code From Design Models For Complex Embedded Systems | Their approach generated C++ code from UML Class and State diagram. | The generated code is to only C++ programming language.<br>It uses only State machine and class diagram for the code generation<br>The generated code is for used in complex embedded systems only |
| 2012 | Model Checking and Code Generation for UML Diagrams Using Graph Transformation | The tool translate class diagram, State machines diagram and Communication diagram into its equivalent Maude code using AToM as a graph transformation tool | The generated code is from class diagram, State machines diagram & Communication diagrams only |
| 2012 | Automatic Code Generation By Model Transformation From Sequence Diagram Of System's Internal Behavior | The Code generator allows generating source code from the sequence diagram of system's internal behavior platform independent model by the mean of a set of model transformations. | The generated code from Sequence diagrams only<br>Only intermediate structural model representing the Java PSM is generated instead of the source code directly. |
| 2012 | A Simple Implementation Of UML Sequence Diagram To Code Generation Through XMI Representation | The Tool deals with code generation for the UML sequence diagram, with the help of the XMI file of the corresponding sequence diagram using the BOUML tool. | The generated code is from UML sequence diagrams only.<br>The generated code is to only java programming language. |
| 2013 | A Framework For Model-Based Code Generation From A Flowchart | The framework can be used to convert flowcharts to code. | It generates code from only flowcharts diagram.<br>The generated code is to only java programming language. |
| 2013 | Generation Of Efficient Embedded C Code From UML/MARTE Models | The approach can be used to transforms application models into C code for execution on embedded system | The generated code is to only C programming language<br>The generated code is form Class, State, Activity diagrams only |
| 2013 | An MDE Approach For Automatic Code Generation From UML/MARTE To OpenCL | The approach can be used to convert From UML/ MARTE(modelling and analysis of real time and embedded system) to openCL API (open standard for programming of heterogeneous systems | The generated code is to only OpenCL API<br>The code is for used in only embedded system. |
| 2013 | Automatic Code Generation From Unified modelling language Sequence Diagrams | The approach can translate From UML 2x sequence Diagrams of use case to code | The generated code is from Sequence diagrams only |

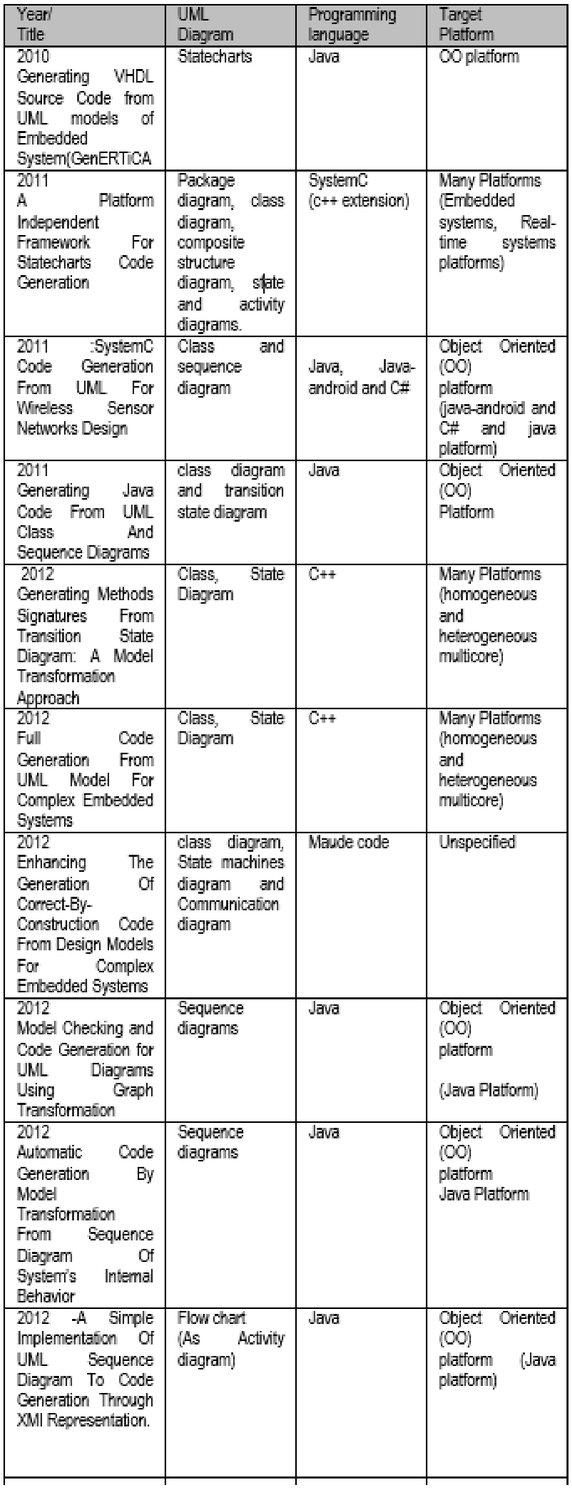


| Year | Title | Achievements | Gaps |
|---|---|---|---|
| 2014 | An Approach to Code Generation from UML Diagrams | The approach can generates code from UML diagram. | The generated code is from Class diagrams only.<br>The generated code is to only java programming language. |
| 2014 | A Model Transformation Approach for Code Generation From State Machine Diagram | The approach generates source code from UML Statechart. | The generated code is from state machine only. |
| 2015 | HDL Code Generation from UML/MARTE Sequence Diagrams for Verifications and Synthesis | The approach generates executable SystemC/TLM and VHDL code with checkers from sequence diagrams | The generated code is from sequence diagrams only.<br>The generated code is to SystemC and VHDL code. |
| 2015 | Software Modelling And Automatic Code Generation Based On Reactive State Diagram | The framework can be used to generate code automatically and a real time framework of state machines | It generates code from only state machines |
| 2015 | Automating Mobile Application Development: UML-Based Code Generation For Android And Windows Phone | GenCode tool generates java-android and C# codes according to the specified application model and target platform | The generated code is to only Java-android and C# programming language. |

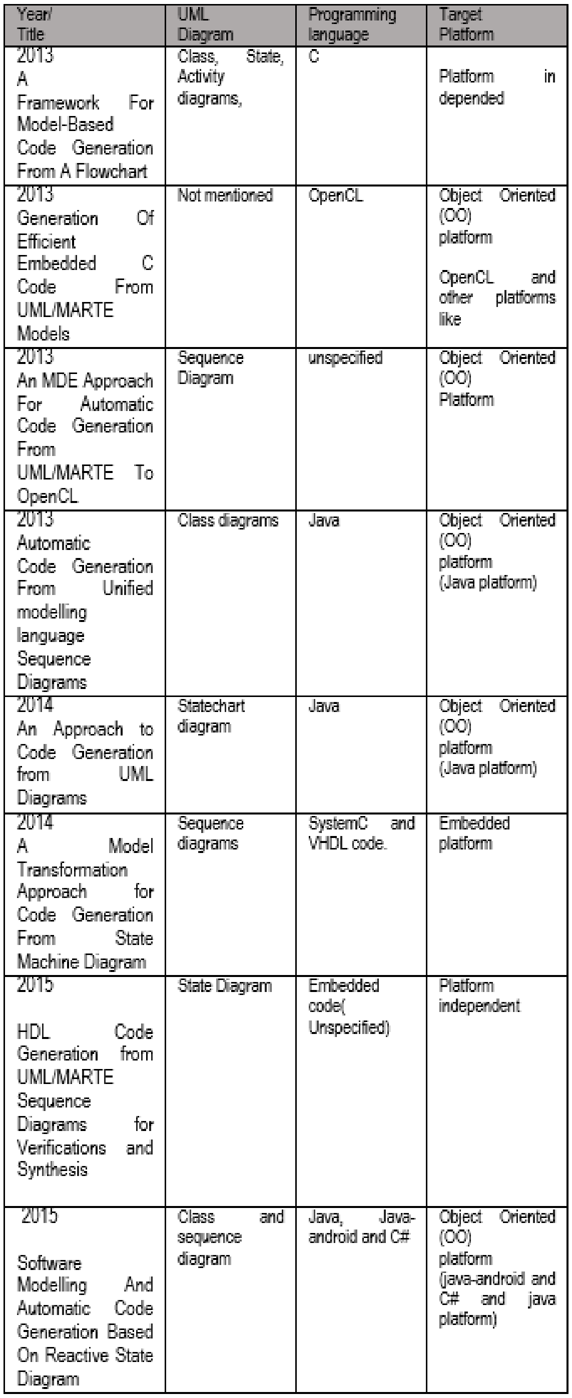


| Year/ Title | UML Diagram | Programming language | Target Platform |
|---|---|---|---|
| 2001<br>Towards Efficient Code Synthesis From Statecharts | Statecharts | C | Many platforms |
| 2002<br>Model Checking and Code Generation for UML State Machines and Collaborations | State machines only | Java | Object Oriented (OO) Platform (Java Platform) |
| 2003<br>Mapping UML Associations Into Java Code | UML Design model Class Diagram | Java | Object Oriented (OO) platform (Java platform) |
| 2003<br>The Fujaba Real-Time Statechart Plugin | class diagram, activity, message sequence charts, state-charts and collaboration diagram | Java | Object Oriented (OO) platform (Real-time java platform) |

| Year/ Title | UML Diagram | Programming language | Target Platform |
|---|---|---|---|
| 2004<br>Mapping UML Statecharts To Java Code | Statecharts | Java | OO Platforms |
| 2005<br>An Object Oriented Approach To Generate Java Code From UML Statecharts | JCode: Class and sequence diagrams | Java | Object Oriented (OO) Platform (Java Platform) |
| 2007<br>FSMC+: A Tool For The Generation Of Java Code From Statecharts | Statecharts | Java and NuSMV code | Many Platforms (Java/Linux platform) |
| 2007<br>Implementing Associations: UML 2.0 To Java 5 | Class Diagram | Java | Object Oriented (OO) platform (Java Platform) |
| 2008<br>GenERTiCA: A Tool for Code Generation and Aspects Weaving | Class, Sequence (mandatory) Activity or state diagrams, or composite structure or deployment diagrams | VHDL, Java, C++, code | Many Platforms (Real-time Femto Java platform ORCOS platform only) |
| 2008<br>A Source Code Generator Based On UML Specification | UML diagrams | unspecified Source code | Many Platforms (Platform independent/dependent) |
| 2008<br>A Code Generation Tool For Embedded Automotive Systems Based On Finite State Machines. | State machines. | Unspecified | Embedded Platform (OSEK/VDX OS for embedded system) |
| 2009<br>Realization Of UML Class And State Machines Models In The C# Code Generation And Execution Framework | Class and state machines diagrams | C | Object Oriented (OO) platform (C# platform) |
| 2009<br>An Object-Oriented Approach To UML Scenarios Engineering And Code Generation. | Sequence Diagram | Java | Platform independent |
| 2009<br>Automatic Code Generation From UML State chart | Statecharts. | Java | OO platforms |
| 2009<br>Automatic Generation of Java Code from UML Diagrams using UJECTOR | UJECTOR: Class , sequence and activity diagrams | Java | Object Oriented (OO) platform (Java Platform) |
| 2009<br>A co-design approach for embedded system modeling and code generation with UML and MARTE | Class and State Machine | C# | OO platform |
| 2009<br>Event Processing in Code Generation and Execution Framework of UML State Machines | Class, Sequence (mandatory) Activity or state diagrams, or composite structure or deployment diagrams, can also be used | VHDL, Java, C++, code | Many Platforms (Real-time Femto Java platform ORCOS platform only) |
| 2010<br>Automatic Code Generation For Embedded Systems: From UML specification to VHDL Code | Class Diagrams | AspectJ Code, Java | Many Platforms (Aspect Oriented and java platform) |
| 2010<br>An Aspect-Oriented Model - Driven Skeleton Code Generation: A Graph-Based Transformation Approach | Sequence, Activity diagrams | SystemC C++ extension | Embedded platforms |
| 2010<br>Automatic SystemC Code Generation From UML Models At Early Stages Of Systems On Chip Design | Activity Diagram | Java and AspectJ | Many Platform (Aspect Oriented and java platform) |
| 2010<br>Tool Support for Code Generation from a UMLsec Property | Class, Sequence (mandatory) Activity or state diagrams, or composite structure or deployment diagrams, can also be used | VHDL, Java, C++, code | Many Platform |



| Year/ Title | UML Diagram | Programming language | Target Platform |
|---|---|---|---|
| 2010<br>Generating VHDL Source Code from UML models of Embedded System(GenERTiCA | Statecharts | Java | OO platform |
| 2011<br>A Platform Independent Framework For Statecharts Code Generation | Package diagram, class diagram, composite structure diagram, state and activity diagrams. | SystemC (c++ extension) | Many Platforms (Embedded systems, Real-time systems platforms) |
| 2011 :SystemC Code Generation From UML For Wireless Sensor Networks Design | Class and sequence diagram | Java, Java-android and C# | Object Oriented (OO) platform (java-android and C# and java platform) |
| 2011<br>Generating Java Code From UML Class And Sequence Diagrams | class diagram and transition state diagram | Java | Object Oriented (OO) Platform |
| 2012<br>Generating Methods Signatures From Transition State Diagram: A Model Transformation Approach | Class, State Diagram | C++ | Many Platforms (homogeneous and heterogeneous multicore) |
| 2012<br>Full Code Generation From UML Model For Complex Embedded Systems | Class, State Diagram | C++ | Many Platforms (homogeneous and heterogeneous multicore) |
| 2012<br>Enhancing The Generation Of Correct-By-Construction Code From Design Models For Complex Embedded Systems | class diagram, State machines diagram and Communication diagram | Maude code | Unspecified |
| 2012<br>Model Checking and Code Generation for UML Diagrams Using Graph Transformation | Sequence diagrams | Java | Object Oriented (OO) platform<br>(Java Platform) |
| 2012<br>Automatic Code Generation By Model Transformation From Sequence Diagram Of System's Internal Behavior | Sequence diagrams | Java | Object Oriented (OO) platform<br>Java Platform |
| 2012 -A Simple Implementation Of UML Sequence Diagram To Code Generation Through XMI Representation. | Flow chart (As Activity diagram) | Java | Object Oriented (OO) platform (Java platform) |



| Year/ Title | UML Diagram | Programming language | Target Platform |
|---|---|---|---|
| 2013<br>A Framework For Model-Based Code Generation From A Flowchart | Class, State, Activity diagrams, | C | Platform in depended |
| 2013<br>Generation Of Efficient Embedded C Code From UML/MARTE Models | Not mentioned | OpenCL | Object Oriented (OO) platform<br>OpenCL and other platforms like |
| 2013<br>An MDE Approach For Automatic Code Generation From UML/MARTE To OpenCL | Sequence Diagram | unspecified | Object Oriented (OO) Platform |
| 2013<br>Automatic Code Generation From Unified modelling language Sequence Diagrams | Class diagrams | Java | Object Oriented (OO) platform (Java platform) |
| 2014<br>An Approach to Code Generation from UML Diagrams | Statechart diagram | Java | Object Oriented (OO) platform (Java platform) |
| 2014<br>A Model Transformation Approach for Code Generation From State Machine Diagram | Sequence diagrams | SystemC and VHDL code. | Embedded platform |
| 2015<br>HDL Code Generation from UML/MARTE Sequence Diagrams for Verifications and Synthesis | State Diagram | Embedded code( Unspecified) | Platform independent |
| 2015<br>Software Modelling And Automatic Code Generation Based On Reactive State Diagram | Class and sequence diagram | Java, Java-android and C# | Object Oriented (OO) platform (java-android and C# and java platform) |

In order to enable us to further analyze and answer our research questions, the charts in Figure 4, Figure 5 and Figure 6 are used to illustrate each of the three factors.

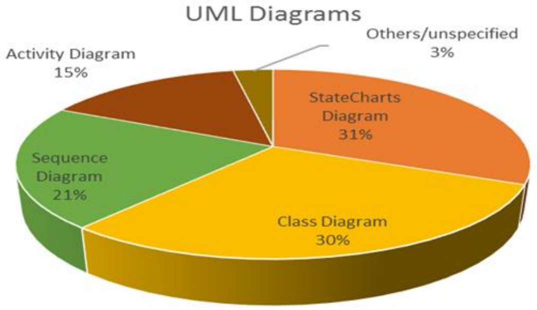


**Figure 4**: UML Diagrams Used in the Selected Papers

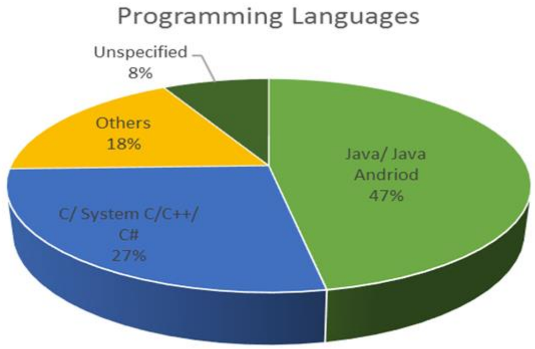


**Figure 5***:* Programming Languages Used in the Selected Papers

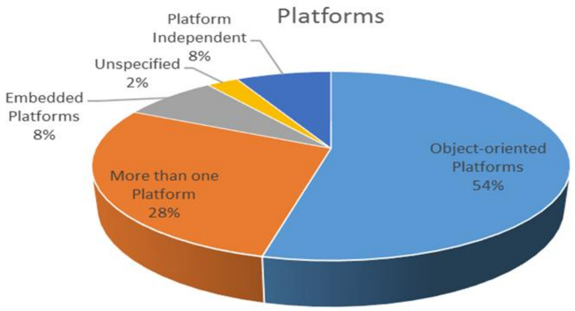


**Figure 6**: Target Platforms Used in the Selected Papers

## DISCUSSION

**Discussion on RQ1**: The techniques that are used for code generation from UML diagrams can be classified into three, namely approaches, tools and frameworks. We found out that out of the 40 selected papers, 13 papers proposed only approaches to code generation from UML diagrams, 13 papers proposed approaches and developed tools to implement their proposed approaches, 4 papers proposed frameworks for code generation from UML diagrams, 2 papers proposed a framework and developed a tool to implement the framework and 8 papers presented tools for automatic code generation from UML diagrams. From the analysis, it is clear that most papers are on approaches and tools for automatic code generation. For now there are only few developed frameworks.

**Discussion on RQ2**: MDA was established with a fundamental notion of completely automated transformation of UML diagrams to code. It can be seen from Table 2 that automatic code generation from UML diagrams to source code has been achieved in all the 40 selected papers. However, only a few papers have used up to 3 UML diagrams for code generation. In fact, half of the papers analyzed were generating source code from a single UML diagram. Also, in a majority of the papers, the UML diagrams are automatically converted to a single programming language source code. Furthermore, in most papers the source code that is being automatically generated from the UML diagrams is targeting a single platform.

**Discussion on RQ3**: State chart diagrams are the most used UML diagrams during 2000-2016. Generating source code from state chart diagrams is one of the challenging tasks due to its dynamic nature and because many state machine concepts are not supported by the object-oriented programming languages (Domí *et.al.* 2012).This is probably the reason why code generation from state charts received great attention during 2000-2016. Class diagrams are the second most used UML diagrams during 2000-2016. In early years class diagrams were the easiest to automate as there was a direct mapping between it and many OO programming languages (Jakimi & Koutbi, 2009b). In later years a new version of UML became available and generating code from class diagrams was revisited. However the code that is automatically generated from only class diagrams generates a skeleton code/framework consisting of class attributes and method signatures and thus cannot be run (Pawde & Chole, 2014). Sequence diagrams are the third most used UML diagrams followed by activity diagrams.

**Discussion on RQ4**: Java is the most used programming language in the 40 selected papers. One of the major reasons is that UML diagrams can only be implemented in a programming language that supports concepts like classes, objects, composition and inheritance (Jakimi & Koutbi, 2009a) and Java supports all these concepts. Another reason is probably because there exist mapping rules from UML diagrams onto Java source code (Niaz & Tanaka, 2004; Thongmak & Muenchaisri, 2002) and similar object oriented languages. Also Java is both a programming language and a programming environment of wide use in the context of heterogeneous and network-wide distributed applications (Marinschek, 2003). Java is followed by similar object oriented languages because of the same reasons. From the analysis, it is clear that most papers are on automatic Java code generation from UML diagrams.

**Discussion on RQ5**: Object oriented platforms are the most used target platforms used in the 40 selected papers. This is because majority of the 40 selected papers were generating Java and C source code, and both of them are OO programming languages that can be executed on object oriented platforms.

## Conclusion

This research has tried to find out the progress made so far on automatic code generation from UML diagrams. To accomplish this objective, a SLR was performed to identify and analyze a set of 40 relevant research papers. The main findings are: The result of the review showed that most papers (34 papers) are on approaches and tools for automatic code generation with only a few (6 papers) on developed frameworks. Also we found out that automatic code generation from UML diagrams to source codes has been largely achieved; however 50% of the selected papers were generating source code from a single UML diagram, 47% of the selected papers were generating java source code from the UML diagrams and 54% of the selected papers were targeting object oriented platforms. As a general conclusion, we have to say that there is still a huge gap in automatically transforming all UML diagrams to several source codes for use in a wide variety of platforms.

This research should be of value to researchers in knowing the gaps remaining in automatic code generation from UML diagrams. This will prompt them to propose satisfactory solutions for bridging these gaps and realizing the MDE dream of automatically generating source codes from UML diagrams.

This research is limited in terms of the database used which is Google Scholar. Future research should involve carrying out the review using databases such as Scopus.

## REFERENCES

Akehurst, D., Howells, G. and Mcdonald-maier, K., 2007. Implementing associations : UML 2 . 0 to Java 5 *Software & Systems Modeling*, 6(1), pp. 3–35.

Andolfato, L., Chiozzi, G., Migliorini, N. and Morales, C.,2011. A platform independent framework for statecharts code generation. In proceedings of the *13th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPS),Grenoble, France*. pp. 614–617.

Bajovs, A., Nikiforova, O. and Sejans, J.,2013. Code Generation from UML Model : State of the Art and Practical Implications. *Applied Computer Systems* 14(1), pp. 9–18.

Bennett, J. , Cooper, K. and Dai, L.,2010. Aspect-oriented Model-Driven Skeleton Code Generation: A Graph based Transformation Approach. *Science of Computer Programming,* 75(8) pp. 689–725.

Bjorklund, D., Lilius, J. and Porres, I.,2001. Towards Efficient Code Synthesis from Statecharts. In *pUML,* (pp. 29–41).

Bousetta, B., Beggar, O. E. and Gadi, T.,2014. A Model Transformation Approach For Code Generation From State Machine Diagram. *IADIS International Journal on Computer Science and Information Systems*, 9, pp. 1–15.

Boussaïd, I., Siarry, P. and Ahmed-nacer, M., 2017. A survey on search-based model-driven engineering. *Automated Software Engineering,*24(2), pp. 233–294.

Boutekkouk,F.,2010. Automatic SystemC Code Generation from UML Models at Early Stages of Systems on Chip Design. *International Journal of Computer Applications,* 8(6). pp. 10–17.

Burmester, S. and Giese,H.,2003. The Fujaba Real-Time Statechart Plugin. *Proc. of the fujaba Days*.

Chama, W., Elmansouri, R. and Chaoui, A.,2012. Model Checking and Code Generation for UML Diagrams Using Graph Transformation. *International Journal of Software Engineering & Applications,* 3(6) p. 39.

Ciccozzi, F., Cicchetti, A . and Sjödin, M.,2012. Full Code Generation from UML Models for Complex Embedded Systems. In *Second International Software Technology Exchange Workshop (STEW) November 28th, Kista, 2012.*

Ciccozzi, F. and Sjödin, M., 2012, September. Enhancing the Generation Of Correct-By-Construction Code from Design Models For Complex Embedded Systems. In *Emerging Technologies & Factory Automation(ETFA),2012 IEEE 17th IEEE*, 2012 on ( pp. 1–4). *IEEE.*

Dias Neto,A. C., Subramanyan,R., Vieira,M. and Travassos,G.H ,2007. A Survey on Model-based Testing Approaches : A Systematic Review. *In Empirical Assessment of Software Engineering Languages and Technologies: held in conjunction with 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, *1st ACM Internal Workshop on* pp. 31–36.

Derezinska A. and Pilitowski, R., 2007. Event Processing in Code Generation and Execution Framework of UML State Machines. *Software Engineering in Progress, Nakom, Poznan,* pp. 80–92.

Derezińska, A. and Pilitowski, R.,2009. Realization of UML Class and State Machine Models in the C # Code Generation and Execution Framework Related work Code generation and execution support. *Informatica(Slovenia)*, 33(4) on pp. 431–440.

Domí, E., Pérez, B. and Rubio, Á. L., 2012.A systematic review of code generation proposals from state machine specification. *Information and Software Technology*, 54(10), pp. 1045–1066.

Ebeid, E. Fummi, F. and Quaglia, D., 2015. HDL code generation from UML / MARTE sequence diagrams for verification and synthesis. *Design automation for embedded systems*, 19(3), pp. 277–299.

El Beggar, O., Bousetta, B. and Gadi, T.,2012,October. Generating Methods Signatures From Transition State Diagram: A Model Transformation Approach. In *Information Science and Technology (CIST), Colloquium*, 2012 Colloquium in (pp.4-9). *IEEE*.

Essaadi, F., Maissa, Y.B. and Dahchour, M.,2017. MDE-based Languages for Wireless Sensor Networks Modeling : A Systematic Mapping Study. In *Advances in Ubiquitous Networking 2* (pp. 331–346). *Springer, Singapore*.

Eveleens,J.L. and Verhoef,C., 2009. The Rise and Fall of the Chaos Report Figures. *IEEE Software,(1),* pp. 30–36.

Fertalj, K. and Brcic, M., 2008. A Source Code Generator Based on UML Specification. *International Journal of Computers and Communication.*2(1) pp. 10–19.

Genova, G., Del Castillo, C.R. and Llorens, J.,2003 Mapping UML Associations into Java. *Journal of Object Technology*, 2(5), pp. 135–162.

Graciano Neto,V.V.,Guessi, M., Oliveira,L.B.R., Oquendo, and Nakagawa, E.Y., 2014. Investigating the Model-Driven Development for Systems-of-systems.In Proceedings of the 2014 European Conference on Software Architecture workshops(p.22).ACM.

Gurad, H. D. and Mahalle, V. S, 2014. An Approach to Code Generation from UML Diagrams. *International Journal of Engineering Sciences & Research Technology*, 3(1).

Heineck,T.,Goncalves,E.,Sousa,A.,Oliveira,M and Castro, J.,2016,September. Model-Driven Development in Robotics Domain:A Systematic Literature Review. In *Software Components, Architectures and Reuse (SBCARS), 2016 X Brazilian Symposium on* (pp. 151–160). *IEEE.*

Hussein, B. M. and Salah, A.,2013. A Framework for Model-Based Code Generation from a Flowchart. *International Journal of Computing Academic Research.*, 2(5), pp. 167–181.

Jakimi A. and El Koutbi, M.,2009. An Object-Oriented Approach to UML Scenarios Engineering and Code Generation.

*International journal of Computer Theory and Engineering.*, 1(1), pp. 1793-8201.

Jakimi A. and El Koutbi,2009. Automatic Code Generation FromUML Statechart. *International journal of Engineering and Techmology,* 1(2), pp. 1793–8236.

Kitchenham, B.,2004.Procedures for Performing Systematic Reviews. Keele, UK,Keele University, 33(2004),pp.1-26.

Klein,J.,Levinson,H. and Marchetti,J.,2015. Model-Driven Engineering : Automatic Code Generation and Beyond. CARNEGIE-MELLON UNIV PITTSBURGH PA PITTSBURGH United States.

Knapp A. and Merz, S.,2002. Model Checking and Code Generation for UML State Machines and Collaborations. *Proc. 5th Wsh. Tools for System Design and Verification*, pp. 59–64.

Lennis, L. and Aedo, J.,2013, January. Generation of Efficient Embedded C Code from UML / MARTE Models. in *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*(p.1). *The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).*

Lindlar,F. and Zimmermann,A., 2008, July. A code generation tool for embedded automotive systems based on finite state machines. In *Industrial Informatics*, 2008. INDIN 20008. 6th IEEE *International Conference on* ( pp. 1539–1544).IEEE.

Loniewski, G., Insfran, E. and Abrahão, S., 2010,October. A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development. In Internation Conference on *Model Driven Engineering Languages and System*( pp. 213–227) Springer, Berlin, Heidelberg.

Maheshwari V., and Prasanna, M.,2015. Generation of Test Case using Automation in Software Systems – A Review. In *Indian Journal of Science and Technology*, 8(35).

Marinschek,M., 2003. Towards Executable UML - Code Generation From Interaction And State Chart Diagrams. na.

Mehmood A. and Jawawi, D.N.,2013. Aspect-oriented model-driven code generation : A systematic mapping study. *Information and Software Technology*, 55(2), pp. 395–411.

Mohi-aldeen, S. M., Deris, S. and Mohamad, R., 2014. Systematic Mapping Study in Automatic Test Case Generation. In SoMet (pp. 703–720).

Montrieux, L., Jürjens, J., Haley, C. B., Yu,Y.,Schobbens,P.Y. and Toussaint, H.,2010. Tool Support for Code Generation from a UMLsec Property. In *Proceedings of the IEEE/ACM international conference on Automated Software Engineering* (pp. 357–358) ACM.

Moreira, T. G., Wehrmeister, M. A. Pereira, C. E.. Petin, J. F. and Levrat, E., 2010a,July. Automatic Code Generation For Embedded Systems: From UML specification to VHDL Code. In I*ndustrial Informatics (INDIN)*, *2010 8th IEEE International Conference on (*pp. 1085–1090).IEEE.

Moreira, T. G., Wehrmeister, M. A., Pereira, C. E., Petin,J.F. and Levrat, E., 2010b. Generating VHDL source code from UML models of embedded systems. In Distributed, Parallel and Biologically Inspired Systems 2010 *(*pp. 125–136).Springer, Berlin, Heidelberg.

Nguyen, P. H., Klein, J.,Le Traon, Y. and Kramer, M. E., 2013,December. A Systematic Review of Model-Driven Security. In *Software Engineering Conference (APSEC),2013 20th Asia-Pacific.* ( Vol.1, pp. 432–441).IEEE.

Nguyen, P. H., Kramer, M., Klein, J. and Le Traon, Y.,2015.An Extensive Systematic Review on Model-Driven Development of Secure Systems.*Information and Software Technology,* 68,pp. 62–81.

Niaz, I.A., 2005. Automatic Code Generation From UML Class and Statechart Diagrams.

Niaz,I.A. and Tanaka,J., 2004, February. Mapping uml statecharts to java code. in *ASTED Conf. on Software Engineering*, pp. 111–116.

Niaz, I. A. and Tanaka, J.,2005. An Object-Oriented Approach To Generate Java Code From UML Statecharts. *International Journal Computer & Information Scence.*, 6(2), pp.83-98.

Omar, E.B., BOUSETTA, B. and Gadi, T. ,2012. Automatic code generation by model transformation from sequence diagram of system ' s internal behavior. *detail,* 1(2) pp. 129–146.

Parada, A. G. , Siegert, E. and De Brisolara, L. B. ,2011,November. Generating Java code from UML Class and Sequence Diagrams. In *on Computing System Engineering(SBESC)*, 2011 *Brazilian Symposium* on (pp. 99–101).IEEE.

Parada, A., Marques,M and De Brisolara,L.B.,2015. Automating mobile application development : UML-based code generation for Android and Windows Phone. *Revista de Informatica Teorica e Aplicade*, 22(2), pp. 31–50.

Pawde, P. R. and Chole, V., 2014. Generation of Java Code Structure from UML Class Diagram. *International Journal Innovation Science and Modern Engineering*, 2(7) pp. 2319-6386.

Qu, M.C., Meng, L.J, Wu,X.H. and Cui, N.G.,2015,June. Software Modelling and Automatic Code Generation Based on Reactive State Diagram. In *International Conference on Computer Information Systems and Industrial Applications(CISIA)*, pp. 899–901

Queiroz, P.G.G. and Braga,R.T.V.,2014,September. Development of Critical Embedded Systems Using Model-driven and Product Lines Techniques : A Systematic Review. *In Software Components, Architectures and Reuse (SBCARS), 2014 Eight Brazilian Symposium on ( pp. 74–83).* IEEE.

Rashid, M., Anwar, M.W. and Khan, A.M.,2015. Towards the Tools Selection in Model Based System Engineering for Embedded Systems - A Systematic Literature Review. *Journal of Systems and Software*, 106, pp. 150–163.

Rashid M. and Anwar, M.W.,2016,June. A systematic investigation of tools in model based System Engineering for Embedded Systems. In *System of Systems Engineering Conference (soSE),2016 11th.* (pp. 1–6). *IEEE.*

Rodrigues, A.W.O., Guyomarc,F. and Dekeyser,J.L.,2013. An MDE Approach for Automatic Code Generation from MARTE to OpenCL. *Computing in Science & Engineering*, 15(1), pp.46-55.

Rosales-Morales, V.Y., Alor.-Hernandez,G.,Garcia-Alcaraz,J.,L. Zatarain-Cabada, R. and Barron-Estrada, M.L.,2015. An analysis of tools for automatic software development and automatic code generation. *Revista. Facultad de Ingenieria Universidad de Antioquia*,(77), pp. 75–87.

Salman, Y. D. and Hashim, N. L., 2016. Automatic Test Case Generation from UML State Chart Diagram : A Survey. In *Advanced Computer and Communication Engineering Technology(pp.123-134).Springer,Cham.*

Shah, S.A.A., Shahzad, R. K., Bukhari, S.S.A., Minhas, N. M. and Humayun, M.,2016. A Review of Class Based Test Case Generation Techniques. *JSW*, 11(5), pp. 464–480.

Summerville,I.,2009. *Software Engineering*, 9th ed. PEARSON.

Syriani, E., Luhunu, L. and Sahraoui, H.,2018. Systematic Mapping Study of Template-based Code Generation.Computer Languages, Systems & Structures, 52, pp.43-62.

Thongmak,M. and Muenchaisri,P.,2002. Design of rules for transforming UML sequence diagrams into Java code. *In Software Engineering Conference,2002. Ninth Asia-Pacific (pp.485-494)*. IEEE.

Tiella,R.,Villafiorita,A.andTomasi,S.,2007,September. FSMC+. a tool for the generation of Java Code from Statecharts. In *Proceedings of the 5th International Symposium on Principles and Practice of Programming in Java*(pp. 93–102). ACM.

Usman M. and Nadeem, A.,2009. Automatic Generation of Java Code from UML Diagrams using UJECTOR. *International Journal of Software Engineering and Its Applications*, 3(2), pp. 21–37.

Vadakkumcheril, T., Mythily, M. and. Valarmathi, M. L., 2013. A Simple Implementation of UML Sequence Diagram to Java Code Generation through XMI Representation. *International Journal of Emerging Technology and Advanced Engineering,* 3, pp. 12.

Vidal, J., De Lamotte, F., Gogniat, G., Soulard, P. and Diguet, J.P.,2009,April. A co-design approach for embedded system modeling and code generation with UML and MARTE. In *Proceedings of the Conference on Design, Automation and Test in Europe (*pp. 226–231). *European Design and Automation*.

Villa, S., Serna, D. and Aedo, J. ,2011. SystemC Code Generation from UML for Wireless Sensor Networks Design. In proceedings of the *International Conference on Modelling,Simulation and Visualization Methods* (MSV) (p.1). *The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).*

Wehrmeister, M. A., Freitas, E. P., Pereira, C. E. and Rammig,F., 2008,May. GenERTiCA: A Tool for Code Generation and Aspect Weaving. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on* (pp. 234–238). IEEE.