IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 4, 2021, accepted March 13, 2021, date of publication March 24, 2021, date of current version April 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068358

Model-Driven Engineering Tools and Languages for Cyber-Physical Systems–A Systematic Literature Review

MUSTAFA ABSHIR MOHAMED[1], GEYLANI KARDAS[1], AND MOHARRAM CHALLENGER[2,3], (Member, IEEE)

[1]International Computer Institute, Ege University, 35100 İzmir, Turkey
[2]Department of Computer Science, University of Antwerp, 2020 Antwerp, Belgium
[3]AnSyMo/CoSys Core Lab, Flanders Make, 3001 Heverlee, Belgium

Corresponding author: Moharram Challenger (moharram.challenger@uantwerpen.be)

This work was supported in part by the Special Research Fund (BOF) at the University of Antwerp under Project 43169.

ABSTRACT The development of Cyber-physical Systems (CPS) draws more interest from both researchers and industrial practitioners considering the opportunities they offer in almost all areas of industry. However, the engineering and management of CPS are challenging tasks due to their inherent heterogeneity and complexity characteristics. Regarding the development of CPS, there currently exists no standard methodology owing to the complexity of the domain. One of the key approaches to reduce the development complexity for CPS is Model-driven Engineering (MDE), which is frequently used in many industrial domains for software development to increase the level of platform abstraction. Nevertheless, it is always almost challenging, especially for the new researchers in this field, to determine the appropriate tools and languages to perform a particular MDE activity during CPS development. To the best of our knowledge, there is no guideline that demonstrates which language(s)/tool(s) to use for the various MDE techniques/phases for the development of CPS. This paper presents a Systematic Literature Review (SLR) study that focuses on identifying and classifying the recent research practices pertaining to CPS development by applying MDE approaches. With the objective of providing a general overview of the field, the study evaluates 140 research papers published during 2010–2018. Accordingly, a precise view of the various MDE tools and languages used in the development life-cycle of CPS, addressed MDE techniques/activities, and targeted CPS components is presented. We believe that the conducted study will guide researchers and practitioners to identify appropriate tools and languages according to the system requirements. It may also help in getting an overall understanding of the research trends for further research and development on the MDE of CPS.

INDEX TERMS Cyber-physical systems (CPS), model-driven engineering (MDE), systematic literature review (SLR), MDE tools and languages.

I. INTRODUCTION

Significant challenges come across developers of Cyber-physical systems (CPS) due to their heterogeneous nature, such as the need for knowledge and skills from multiple academic disciplines, the integration of the artifacts of these disciplines, and the difficulty of the maintenance activities of such heterogeneous artifacts. In order to address these challenges and reduce the complexity of CPS development, one of the key approaches is Model-driven Engineering (MDE), which is frequently used in many industrial domains for software development [1].

The abundance of different hardware platforms available for CPS makes their development complex [2]–[4]. There is a need for a methodology that permits efficient raise of the abstraction level to overcome issues of heterogeneity induced by the multidisciplinary nature of such systems. Towards this goal, many researchers believe that MDE is a better alternative solution to overcome challenges such as development complexity, heterogeneity, adaptability, and reuse and they propose various applications of MDE for CPS development (e.g. [5]–[9]). However, to the best of our knowledge, no secondary study, highlighting both previous

The associate editor coordinating the review of this manuscript and approving it for publication was Claudia Raibulet.

researches and current research efforts related to applying MDE approaches for CPS, has been performed yet. This overview would be helpful to both researchers and industrial practitioners for discovering the pros and cons for applying MDE in CPS and for identifying interesting research directions. Without such a secondary study, it may be cumbersome to determine what was proposed, what has been successfully completed and what rather has failed.

In this paper, we present a Systematic Literature Review (SLR) of the studies which used MDE techniques such as domain-specific modeling, metamodeling, validation & verification, model transformation, simulation, code generation and analysis for CPS. An evaluation within the context of a set of research questions and analysis of the proposed toolchains in the primary studies is performed within the scope of this study. As a result, numerous tools and languages have been identified for modeling, model transformation, simulation, code generation, analysis, verification and validation activities for CPS.

The aim of this study is not to compare any of the languages/tools developed by/used in the primary studies for the MDE of CPS since such comparisons are beyond the scope of the SLR studies. Our aim is rather to report on the languages/tools developed/used in the primary studies and present them to the readers in a systematic manner, e.g. by identifying meaningful correlations between the results achieved from the evaluation of the introduced research questions which is the objective of the SLR studies [10].

The results of this study may help the researchers and practitioners to easily reach the desired class of studies and related publications considering the tools and languages, technologies, and best practices used. This study also enables researchers avoid unnecessary duplication of the trials and errors which may encounter during MDE of CPS. Finally, it identifies and determines languages and tools in each and every MDE technique/phase as well as the addressed CPS components.

The remainder of the paper is organized as follows: A brief introduction to CPS and MDE are given in section 2. Section 3 discusses the related work. Section 4 describes the research methodology and protocol definition to carry out the SLR. Section 5 shows the achieved results. The discussion of the results and the conclusions are presented in sections 6 and 7 respectively.

II. BACKGROUND

In the following subsections CPS and MDE with including their structures and construction phases are briefly discussed before giving the details of the conducted SLR.

A. CYBER-PHYSICAL SYSTEMS

A CPS is a system whose computational and communication components control and monitor physical phenomena [11], [12]. In CPS, sensors monitor and measure certain physical phenomena, like pressure, temperature, light, touch, etc. The measured data are transferred to the controllers/software through communication elements (i.e. wired/wireless network). The controllers/software make decisions/actions based on the received data from the sensors and send them through communication elements to actuators which in return make changes to the physical phenomena [12], [13], the feedback loop. The overall architecture of a CPS is depicted in Figure 1.

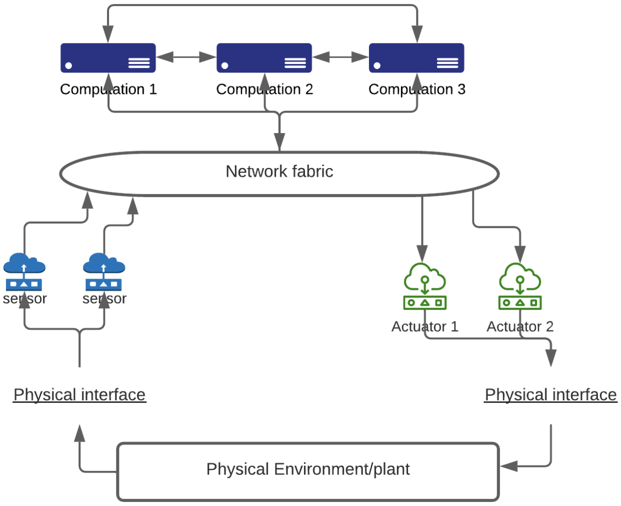

FIGURE 1. Example structure of a CPS (adapted from [12]).

The design phases of a CPS could be divided into three main phases, i.e., modeling, design, analysis. Firstly, modeling is the mechanism of imitating and reflecting the system properties so as to obtain a deeper understanding of the system under study. This stage includes activities such as modeling the dynamic behaviour, that is, modeling the continuous and discrete dynamics of the system. Secondly, designing and building the CPS computational and physical processes such as the processors, memory architectures, I/O mechanisms, operating systems, multitasking and scheduling capabilities, sensors, actuators, and the network. Finally, analysis, which is to ensure that the system meets certain requirements and specifications, takes place. Various techniques, such as using temporal logic, exist to specify the desired and undesired behaviors and to check that the implementation complies with its specification. [12]

Cyber-Physical Systems vs. Embedded Systems: According to Heath [14]; ''An embedded system is a microprocessor-based system that is built to control a function or range of functions and is not designed to be programmed by the end user in the same way that a PC is''. Embedded systems are designed to undertake one function, but with different options and choices [14]. Examples of embedded systems include, but are not limited to, MP3 players, mobile phones, video game consoles, digital cameras, DVD players, and GPS.

On the other hand, as also indicated in [12], a CPS is an integration of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. That is to say, CPSs are realtime, distributed feedback systems, interconnecting embedded systems, networking systems, physical systems and human users [15], where a centralized system monitors and controls the physical process(es), mostly with feedback loops [16]. Examples of CPS include smart factories, Industry 4.0/Cyber-physical production systems (CPPS), Unmanned Aerial Vehicles, Safety-critical systems, and smart cities.

In contrast to the conventional embedded systems, a comprehensive CPS is usually conceived as a network of interacting elements with physical input and output rather than as standalone devices. While CPSs tend to emphasize the interconnection between the cyber/computational and physical elements, embedded systems happen to focus on the computational elements and they are often referred to as ''closed boxes'' due to the fact that these systems do not expose their computational capabilities to the outside [16]. A CPS may consist of a group of interconnected embedded systems, sometimes functioning as the main brain of the whole system, with sensors and actuators. Therefore, one can deduce that embedded systems compose a subset of CPS.

B. MODEL-DRIVEN ENGINEERING

MDE advocates for the use of models as the backbone of the software development process, with the aim of automating the process from design stage to implementation stage. It tends to raise the abstraction level from computing concepts to an abstraction closer to the understanding of both the developers and the other stakeholders [17]. Models are used in MDE as the basic building blocks for the development of software artifacts. MDE paradigm raises the abstraction level of software/system development from low-level artifacts to a higher-level of models and bridges the gap between problem identification and software implementation phases. This can be done by thoroughly/partially generating either software implementations (C++, Java, and C#) or deployment artifacts (such as XML-based configuration) from models that describe the system at multiple levels of abstraction, and from a variety of perspectives [1], [18], [19].

Like any other paradigm, MDE has pros and cons [19]. Taking into consideration its pros, for instance, it helps to reduce the time/effort required for the system/software development. As the code is partially generated automatically, the task of the developers is to complete the parts that he/she intended to generate automatically or could not be generated by the used tool. Moreover, MDE helps increasing the quality, productivity and maintainability of the final artifacts. With regard to the cons, one of the major drawbacks of MDE could be that more time is to be spent on system analysis in order to correctly model the system and therefore generate the correct and optimized code. Researchers and practitioners may also face with various MDE challenges within the aspects of e.g. its sometimes uncertain and bidirectional foundation, complexity of the application domains and lack of good tooling [20].

Taking a glimpse at the literature, it is observable that MDE is used for the various software development lifecycle (SDLC) activities [19], which is to say, it is possible to support different SDLC activities using MDE:

- System design: designing is the main activity of MDE, in which a fully functional DSL and its tool are used. This may consist of more than one MDE technique, such as modeling, transformation and code generation.
- Modeling: Presentation of a graphical or textual representation of the system that may be used for tasks such as modeling system features, functionality or behavior, and modeling time constraints or system threats, etc.
- Transformations: transformations are regarded as an integral part of MDE as it helps developers define mappings between the models.
- Code generation: the aim here is to generate an architectural code out of the designed models. The code is generated from the designed models by means of model-to-text transformation or rule-based template engine.
- Validation and Verification (V&V): This could be seen as testing or analyzing MDE techniques, such as the verification of the designed models, the analysis of the accuracy of the transformations and their rules, and also the validation of the final artifacts. Ensuring that the right system is built is called validation. One important method for validation is simulation of system behavior. However, assuring that the system is behaving according to the specifications is called verification. There exist various methods of verifying a system in MDE such as requirement analysis, performance analysis, failure analysis, formal methods and model checking [21].
- Simulation: Simulation can be done using general-purpose languages (such as Python, C++, Java, etc) or simulation specific programming languages like SIMULA, in which both use a compiler or interpreter to run the simulation. However, utilizing MDE tools and techniques like modeling and model transformation will make performing simulations cost-effective and effort-saving [22].
- Requirement analysis: Requirement engineering/analysis (RE) combines tasks such as defining, modeling, validating, and reaching agreement on the system requirements. As suggested by [23], MDE can be used as a technical solution for requirement analysis, as it is possible with MDE to integrate the tasks of RE.

MDE can also be further used for system analysis activities e.g. safety analysis, security analysis, operational analysis and dependability analysis.

III. RELATED WORK

Since the scope of this study is to present an SLR on the state-of-the-art of MDE tools and languages for CPS, the related secondary studies (surveys, systematic mapping (SM), SLR, and Tertiary Studies) are addressed in this section as the related work. Although, to the best of our knowledge, there

TABLE 1. Summarized Related work.

Study	Methodology	Targeted Systems	Paradigm/ Approach	# of Papers	Studies b/w years
[24]	SLR	CPS	MPM	265	2006-2017
[25]	SLR	Embedded Systems	MBSE	61	2008-2014
[26]	SLR	Safety-critical embedded systems	PLE+MDE	19	N/A
[27]	SM	Mobile robot systems	MDE	69	After 2000
[28]	Survey	Embedded systems	MBE	N/A	N/A
[29]	Survey	Embedded software	UML+MDA	N/A	N/A
[30]	Survey	N/A	Co-simulation	84	2011-2016
[31]	Survey	Industrial-IoT	N/A	N/A	N/A
[32]	Survey	CPS	N/A	N/A	N/A
[33]	SLR	Embedded systems	MBSE	58	2008-2015
[34]	SLR	CPS	MDE (only security aspect)	69	N/A
[35]	Survey	CPS	MDE	N/A	N/A
This study	SLR	CPS	MDE	140	2010-2018

is no secondary study exactly on this topic, the following studies are relevant to the conducted SLR. Table 1 presents a summary of these related work.

In [24], an SLR on multi-paradigm modeling for cyber-physical systems is presented, where the objective of the study is to investigate studies promoting multi-modeling, multi-view, and multi-formalism approaches for the development of CPS. 265 research papers published during the period 2006–2017 are evaluated. The study reported used approaches and tools for multi-paradigm modeling as well as indicating the type of formalism presented, and which tool and/or language is used for implementing it. Their focus is mainly addressing the domain distribution and generality of the approaches. Furthermore, they report on the actors and stakeholders involved in the modeling process and their background knowledge. On the other hand, the objective of our study is mainly to investigate the reported MDE approaches and tools used for the development of CPS. This is to say when discussing modeling approaches in our study, we focus on the presented MDE activities/techniques (i.e. metamodeling, model transformation, code generation, analysis, etc.) and the languages/tools used for each of these MDE activities/techniques, as well as presenting the addressed CPS components which are not taken into account in [24].

The authors of [25] conducted SLR of the development of embedded systems using model-based system engineering (MBSE) approach. The study reviewed 61 research papers published in one of the four renowned scientific databases (IEEE, SPRINGER, ELSEVIER, and ACM) during the years 2008-2014. Subsequently, primary studies are grouped into six categories according to their relevance to the corresponding MBSE activity namely general category, modeling category, model transformation category, model verification category, simulation category, and property specification category. As a result, the study presents 28 tools which support modeling, model transformation, validation, and verification activities. The study examined the utilization of UML and SysML/MARTE profiles, and it also analyzed the application of both model-to-model and model-to-text transformations.

The SLR study in [33] covered 52 studies published during 2008-2015, with the objective of investigating the research trends in the representation of constraints and use or customization of UML together with its SysML/MARTE profiles for requirement specification of embedded systems. The study thus looked at the practices of UML profiles/diagrams for design specifications, and the trends to specify properties/constraints for design verification. As a result, the study found that the combination of UML and MARTE profiles is the major trend, that the class/block diagram is commonly used to specify structural aspects, whereas the state machine/activity diagram is commonly used to specify behavioral aspects. While the main focus of the work in [33] is to investigate how UML and its profiles SysML/MARTE where utilized for requirement specification of embedded systems, in our work, we aim at reporting the various MDE tools and languages used in the development life-cycle of CPS, addressed MDE techniques/activities, and targeted CPS components.

Another SLR is presented in [26] where the authors investigate studies combining Product Line Engineering (PLE) and MDE for the development of safety-critical embedded systems. This study further examined whether there are empirical studies applying the aforementioned techniques in the development process of safety-critical embedded systems. The study exposes that in recent years, use of MDE combined with PLE techniques to build safety-critical embedded systems is gradually growing. The study also states that the proposed approaches in the primary studies are not compared with any other related studies, besides, these approaches do not explicitly differentiate between the software and hardware variabilities.

The study in [34] concentrated mainly on the security and safety aspects of CPS, where the authors conducted an SLR study to identify the existing model-driven approaches for secure CPS that specifically cover both software and platform description. As a result, the analysis found out 17 model-driven security approaches, 7 of which were specific to CPS and were investigated more closely.

An SM study is presented in [27]. This study investigates the implementations of MDE in the field of mobile robot systems (MRS). In this study, 69 research papers were selected, and as a result, the authors found out that many domain-specific modeling languages (DSMLs) are supported with tools which are mostly built ad-hoc. Also, they reported that the solutions based on UML and using Eclipse-based tools were less preferred in this field.

A survey, presented in [28], collected the quantitative data from 113 subjects to provide the current state-of-practice (SoP) and challenges faced by the domain of embedded systems due to weaknesses in model-based engineering (MBE). The survey has two research questions, the first question is related to capturing the state of MBE practice in the embedded systems domain, how much activities concern MBE compared to non-MBE, and understanding the pros and cons of adopting and deploying MBE. The second question is about estimating whether there are important variations in the SoP between different groups in the embedded systems domain. As a result, the study provides information about the used methods and tools, purposes of models, effects of using it, and weaknesses of MBE. Furthermore, answers to the survey shown that most of the participants believe the positive outcomes of MBE distinctly exceeded the negative outcomes. Nonetheless, survey participants mentioned weaknesses such as the interoperability challenges amongst existing tools, and high efforts to train the developers for MDE.

Another survey is presented in [29]. The study introduces statistical findings about the use of UML modeling and model-driven approaches for the design of embedded software in Brazil. The goal of this study is to identify gaps in the knowledge of how exactly UML and MDE or Model-driven Architecture (MDA) are used in industry, and to provide an understanding of how social and organizational factors impact the use of UML and MDE/MDA. Model-driven techniques and tools for CPS were surveyed in [35] including the provided features and solutions to the challenges. However, the given analysis is limited to evaluating only 10 specific representative techniques and tools.

The survey in [30] focused on summarizing and classifying the many state-of-the-art co-simulation approaches. The survey covers the key challenges in enabling co-simulation. Initially, two major co-simulation domains were addressed separately, namely continuous-time and discrete-event-based co-simulation. Then, the challenges which occur when the two domains are merged were discussed. The proposed taxonomy is used for the classification of the works related to co-simulation. 84 papers published during the period 2011-2016 were read and classified.

The work in [31] carried out a survey on the industrial Internet of Things (I-IoT), from a CPS viewpoint, concentrating mainly on control, networking, and computing systems perspectives. The contribution of this research revolves primarily around three points: Firstly, presenting I-IoT architecture and identifying I-IoT performance requirements from the communications point of view (e.g. latency and reliability). Secondly, conducting an I-IoT survey from the three critical perspectives referred to above. Finally, presenting the challenges and future research of control, networking and computing systems in the I-IoT. Similarly, in [32], a survey in CPS is carried out, in order to summarize CPS features. The study gives the research progress from different perspectives like energy control, security control, transmission and management, control technique, system resource allocation, and model-based software design. Subsequently, the study incorporates three classic applications so as to prove that the prospects of CPSs are alluring. Ultimately, the study demonstrates research challenges and outlines some suggestions for future work.

Furthermore, [36] discussed CPS challenges. The authors discussed the maturation of embedded systems into CPS and the design and operating challenges that resulted from it. The paper argues that the maturation of embedded systems has been growing by leap and bound as the networking technology became available. The research focused primarily on the functional aspects of CPS. Authors linked general software modeling challenges to the specific challenges of CPSs. The study posed many challenges related to maintainability, dependability, security, resilience, certification and policy. Similarly, we addressed the challenges of CPS in an accompanying study [37], where our objective was to report the approaches followed when applying MDE for CPS, application domains, presented case studies and addressed CPS challenges by conducting a systematic mapping study. Results related to the challenges addressed by the examined studies have shown that numerous CPS challenges were addressed, namely, complexity, dependability, flexibility, interoperability, latency, predictability, reliability, remote monitoring, security and sustainability. Complexity and interoperability seems to be the most addressed.

Unlike the work presented in [25], [26], [35] for embedded systems only, our study focuses on conducting an SLR on the primary studies concerning the development of CPS using the MDE paradigm/approach. Our SLR also differentiates from the above mentioned studies [30]–[32], [34] especially within the context of the application domain. As briefly discussed above, the work in [30] focused on co-simulation approaches regardless of the specific domains of the system, i.e. it provided a remarkable classification of the studies using continuous-time and discrete-event-based co-simulations however application domains such as CPS, IoT, Embedded system and Industry 4.0 are not in the scope of the work. Finally, the surveys in [31], [32] addressed the I-IoT and CPS domains, respectively. However, they did not consider any development paradigm/approach applied for the construction of these systems while we present an SLR of the tools/techniques used during the MDE of CPS.

IV. METHODOLOGY

SLR is a widely adopted approach when conducting a comprehensive review relevant to specific research questions. Several studies present guidelines [10], [38], [39] and approaches [40] to be followed for SLR studies.

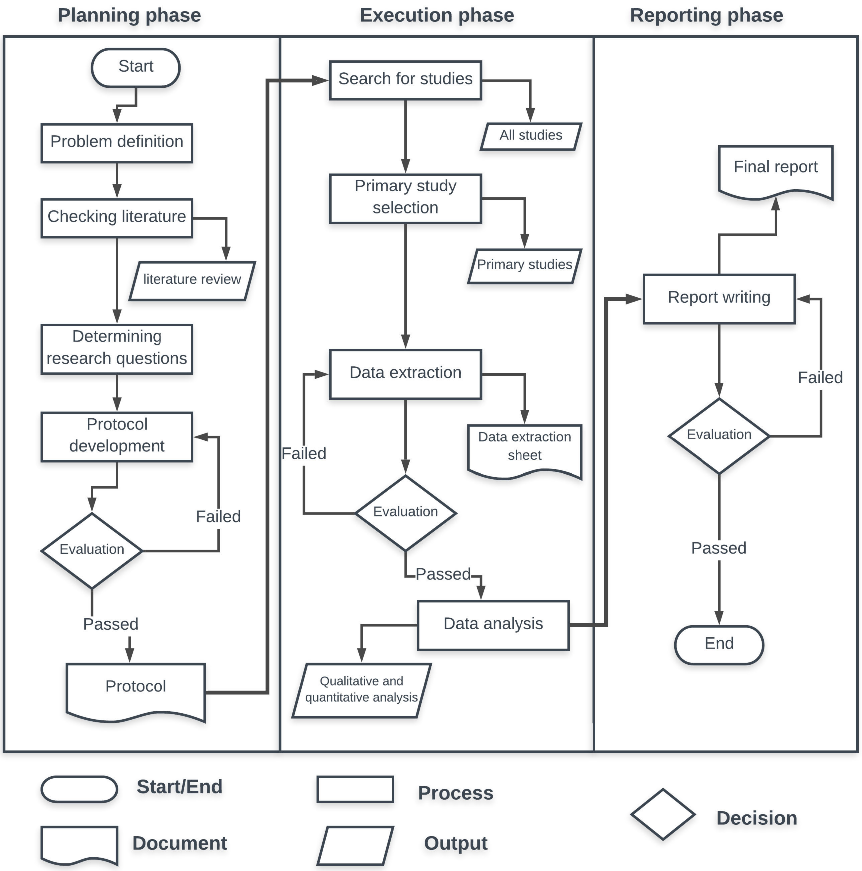

FIGURE 2. Overview of the systematic literature review process.

In this section, the applied methodology for the conducted SLR is discussed. Initially, the process followed during this work is described. Then, research questions are defined and primary study selection strategy is discussed. This is followed by specifying inclusion and exclusion criteria in addition to the quality assessment and self-assessment criteria, and finally determining the data extraction procedure.

A. PROCESS

We developed the procedure of the systematic review in this study by following the guidelines defined in [10]. Figure 2 shows an overview of the followed process. The SLR composes three main phases; review planning, review execution, and review reporting [38]. The protocol and research questions of the SLR are prepared in the planning phase while collecting the studies, extracting data from them and analyzing these data are performed in the executing phase. Finally, reporting the achieved results takes place. In every ''Evaluation'' process shown in Figure 2, all researchers participating in the SLR decide whether or not the previous SLR process has been successfully completed. If yes, which is denoted as ''Passed'' in the diagram, the researchers move to the next process; otherwise, which is denoted as ''Failed'' in the diagram, the previous process will be revisited until all reviewers agree.

B. RESEARCH QUESTIONS

Research questions were identified by following the PICOC criteria [10] (see Table 2). The research questions of this study are determined as below:

- **RQ1:** Which activity(s) of the system development is/are addressed in the study (using MDE)?
  Motivation: The motivation behind the inclusion of this research question is to report on the different activities of which MDE techniques have been applied in primary studies. The results for this question are expected to retrieve the information on the extent to which the investigated paper covered MDE activities/techniques, e.g. modeling, model transformation, simulation, code generation, analysis, verification and validation. As a result, the outcome of this question will help readers to identify which studies applied which MDE activities/techniques.
- **RQ2:** Is/Are there any tool and/or DSL developed for MDE of CPS by the study?
  Motivation: The objective of this question is to find out the tool(s) developed by the investigated studies, their availability, and the used languages while developing these tools. The results of this question will help readers identifying recently developed tools as well as the ongoing work in this field. Moreover, the results of this question are to be further correlated with RQ1 results. The

TABLE 2. Keywords definition based on PICOC criteria.

Population	"cyber-physical system*" OR "cyber physical system*" OR "smart system*" OR "cyberphysical systems" OR "cps"
Intervention	MDE OR MDD OR MDA OR "model-driven *" OR "model driven *" OR "code generation" OR "generative approach*" OR "model-based approach*" OR "domain specific model*" OR metamodel* OR "meta-model*" OR "meta model*" OR "modeling approach*"
Comparison	Not applicable
Outcome	Report on the current state-of-the-art approaches, languages and tools of MDE for CPS.
Context	Peer-reviewed publications.

outcome of this correlation will lead determining which tools were developed for which MDE phase/technique. Corresponding sub-questions are listed as follows:

- **RQ2.1:** If any tool and/or language is developed in the study, is it reported?
- **RQ2.2:** Is/Are the developed tool(s) and/or languages available and/or accessible?
- **RQ2.3:** What is/are the framework(s) or programming language(s) for its/their development?

- **RQ3:** Is/Are there any tool(s) and/or Language(s) used to apply MDE in/for CPS in the study?
  Motivation: With answering this question, the used languages/tools by the study under examination are reported. The results of this question will provide readers with an overall view of the most widely used languages/tools in this domain of application of MDE on CPS. Furthermore, the results of this question are to be correlated with the results of RQ1, in order to present which language(s)/tool(s) is/are used in which MDE activity/phase. Consequently, this will help readers to reach a meaningful understanding of all the languages/tools used in all the examined studies. This question has the following sub-question:
  - **RQ3.1:** Which tool(s) and/or language(s) is/are presented/used in each phase of the system development?
- **RQ4:** What is/are the CPS component(s) addressed in the study?
  Motivation: This question was shaped with the motivation of getting deeper insights into the modeled CPS components, that is to say, reporting on the CPS components such as sensors, cyber components, physical components, actuators which are modeled by the study under investigation. The results of this question are to be correlated with the results of RQ1. This correlation allows readers to get a clearer picture of which MDE activity(s) is/are carried out for which CPS component. In addition, the results of this correlation will help readers Identifying where the current research is more condensed, and easily reaches the desired set of studies. Results of this correlation are shown in Table 13

C. SEARCH AND SELECTION STRATEGY

This stage can be considered as one of the most important and critical stages when conducting a secondary study (i.e. in this case, SLR). Therefore, it should be carefully defined since the search of primary studies should ensure the comprehensive coverage of the topic under consideration. For a search strategy to be optimal, it needs to simultaneously include utmost relevant primary studies (recall) and exclude irrelevant ones (precision). One can deduce that an optimal search strategy must have 100% recall and/or 100% precision. Nevertheless, it is unlikely that a search strategy gives 100% in both/either recall and/or precision. Accordingly, one should come up with a gratifying trade-off search strategy (i.e. good enough), that results in not many relevant studies missed, and a manageable quantity of irrelevant studies included [41].

The search strategy, developed in this study, composes four stages. Firstly, an automatic search over the most relevant scientific digital libraries was performed. Secondly, all duplicate papers were removed. Thirdly, following predetermined criteria of inclusion, only papers related to the topic were considered. Eventually, further studies were searched by forward snowballing [42] (see Section IV-C4 for the detailed description of the forward snowballing). The composition of the search and selection strategy followed in this work is shown in Figure 3.

It is worth indicating that due to the ambiguity between the terms such as CPS, Internet of Things (IoT), embedded systems and self-adapting systems, considering all these terms and their variations in the search keyword of an SLR, will result in an immense number of studies, ultimately leading to a lack of the credibility, validity, and reproducibility of this study and making the study of an undefined scope. It is therefore decided to consider only those studies that identify themselves as CPS-related studies in this SLR. This means that if a primary study contains any of the keywords (i.e. ''cyber-physical system*'' OR ''cyber physical system*'' OR ''smart system*'' OR ''cyberphysical systems'' OR ''cps'') in its title, abstract, introduction or conclusion, it will be included in our initial pool of studies.

1) PERFORMING AUTOMATIC SEARCH

One of the key issues with SLR studies is getting all the relevant studies on the examined topic [43]. To get as many related primary studies as possible, an automatic search was performed on the following digital libraries: ACM, IEEE Xplore, ScienceDirect, DBLP, Scopus and Web of Science. The initial pool included all indexed studies found in the aforementioned digital libraries, using predefined search keywords.

Initially, digital libraries, were manually searched. However, a massive number of studies (on average, above 5000 results from each search engine) were achieved.

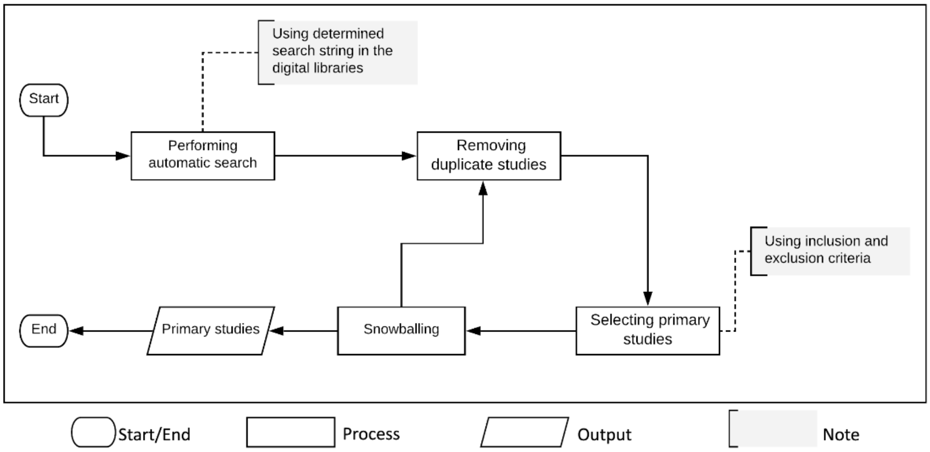
Using determined search string in the digital libraries
Start
Performing automatic search
Removing duplicate studies
Using inclusion and exclusion criteria
End
Primary studies
Snowballing
Selecting primary studies
Start/End
Process
Output
Note

FIGURE 3. Search and selection strategy.

Therefore, it is decided to perform an automatic search as it is advised during conducting SLR in [10]. PICOC criteria [10] were used to define the keywords, shown in Table 2, which leads to form "good enough" search strings. Table 3 shows the searched digital libraries and the corresponding search string(s) used. After completing the automatic search, 646 studies were obtained.

Many challenges were encountered while using the digital libraries: One of the main challenges of using these digital libraries is the lack of guidelines explaining how to use the advanced search features of these digital libraries. Also, the number of allowed terms of the search string is limited in digital libraries like ScienceDirect and DBLP, which causes splitting the search string into multiple search strings. Also, wild cards are not supported in ScienceDirect. Another challenge is that the digital libraries like ACM and IEEE do not provide the capability to restrict the search to more than one specific area at once, e.g. title, abstract, and keywords combined.

2) REMOVING DUPLICATE STUDIES

Initially, the pool of primary studies was kept in Mendeley reference manager.[1] Mendeley was used to facilitate the process of determining duplicate studies. The process of duplicate checking goes until further stages (i.e., forward snowballing). The eliminated duplicate papers were 113 studies, so, 533 studies remained. Two papers are considered as duplicate if:

- their title, author(s), publication date and venue are the same. In case of different versions of the same paper, the most recent is kept.
- the same paper is published in different venues, one of them is selected (the most recent).
- the same study has both journal and conference publications, the journal publication is considered as it contains the extended study and provides more information.

3) SELECTING PRIMARY STUDIES

In this stage, primary studies are selected following predefined inclusion and exclusion criteria (see Section IV-D). Only those studies matching the criteria are included in the final pool of the research. The criteria were applied considering the reading of Title, Abstract, Keywords, and Introduction sections, however, if it is not enough for reaching a decision, other parts like Methodology and Conclusion are considered. The process of selecting primary studies is shown in Figure 4. The inclusion or exclusion of studies are performed in several iterations:

- **Iteration 1:** The primary reviewer went through each study reading its title and abstract, and checking the general content (figure, models, tables, etc.). Studies which meet the inclusion and exclusion criteria passed to the next iteration (278 studies were removed in this iteration).
- **Iteration 2:** All the studies which passed iteration 1 were read in more detail by further reading the related paper's introduction and conclusion sections and if necessary other sections (e.g. methodology and case study). This iteration resulted in including 88 papers and excluding 82 papers. 85 papers left undecided "to be reviewed".
- **Iteration 3:** The 85 undecided papers in iteration 2 were again reviewed with a secondary reviewer. In this stage, both reviewers agreed on either including or excluding the paper. As a result, 34 papers were later included, whereas 51 papers were later excluded.

To sum up, 88 papers were included from iteration 2 and 34 papers were included from iteration 3, forming a pool of 122 primary studies.

4) FORWARD SNOWBALLING

To assure that no potential primary studies get ignored, studies that might not have been reached on the basis of automatic searching were also searched. It is critical to obtain a good

[1] https://www.mendeley.com

TABLE 3. Search strings.

Digital Library	Results	Search query
IEEE	164	("model-driven development" OR "model-driven engineering" OR "model-driven architecture" OR "code generation" OR "generative approach" OR "model-based approach" OR "model-driven approach" OR "domain specific model*" OR metamodel OR "meta-model" OR "meta model" OR "modeling approach") AND ("cyber-physical system*" OR "cyber physical system*" OR "smart system*" OR "cyberphysical systems" OR "cps")
ACM	55	recordAbstract:(("model-driven development" OR "model-driven engineering" OR "model-driven architecture" OR "code generation" OR "generative approach" OR "model-based approach" OR "model-driven approach" OR "domain specific model*" OR metamodel OR "meta-model" OR "meta model" OR "modeling approach") AND ("cyber-physical system*" OR "cyber physical system*" OR "smart system*" OR "cyberphysical systems" OR "cps"))
Web of Science	16	TI=(("model-driven development" OR "model-driven engineering" OR "model-driven architecture" OR "code generation" OR "generative approach" OR "model-based approach" OR "model-driven approach" OR "domain specific model*" OR metamodel OR "meta-model" OR "meta model" OR "modeling approach") AND ("cyber-physical system*" OR "cyber physical system*" OR "smart system*" OR "cyberphysical systems" OR "cps"))) AND LANGUAGE: (English)
Scopus	363	The related search string is too long to fit in this paper. It is available on IEEE Data Port (in protocol sheet): https://dx.doi.org/10.21227/zbkz-6461
Science Direct	23	("code generation" OR "generative approach" OR "domain specific modelling" OR "modelling approach") AND ("cyber-physical systems" OR "cyber physical systems" OR "smart systems" OR cps OR "cyberphysical systems")
	12	("model-driven development" OR "model-driven engineering" OR "model-driven architecture" OR "model-based approach" OR "model-driven approach") AND ("cyber-physical systems" OR "cyber physical systems" OR "smart systems" OR "cyberphysical systems")
	9	(metamodel OR "meta-model" OR "meta model") AND ("cyber-physical systems" OR "cyber physical systems" OR "smart systems" OR "cyberphysical systems")
DBLP	4	(metamodel | "meta-model" | "meta model") ("cyber-physical systems" | "cyber physical systems" | "smart systems" | "cyberphysical systems")
	0	("model-driven development" | "model-driven engineering" | "model-driven architecture" | "model-based approach" | "model-driven approach") ("cyber-physical systems" | "cyber physical systems" | "smart systems" | "cyberphysical systems")
	0	("code generation" | "generative approach" | "domain specific modelling" | "modelling approach") ("cyber-physical systems" | "cyber physical systems" | "smart systems" | cps | "cyberphysical systems")

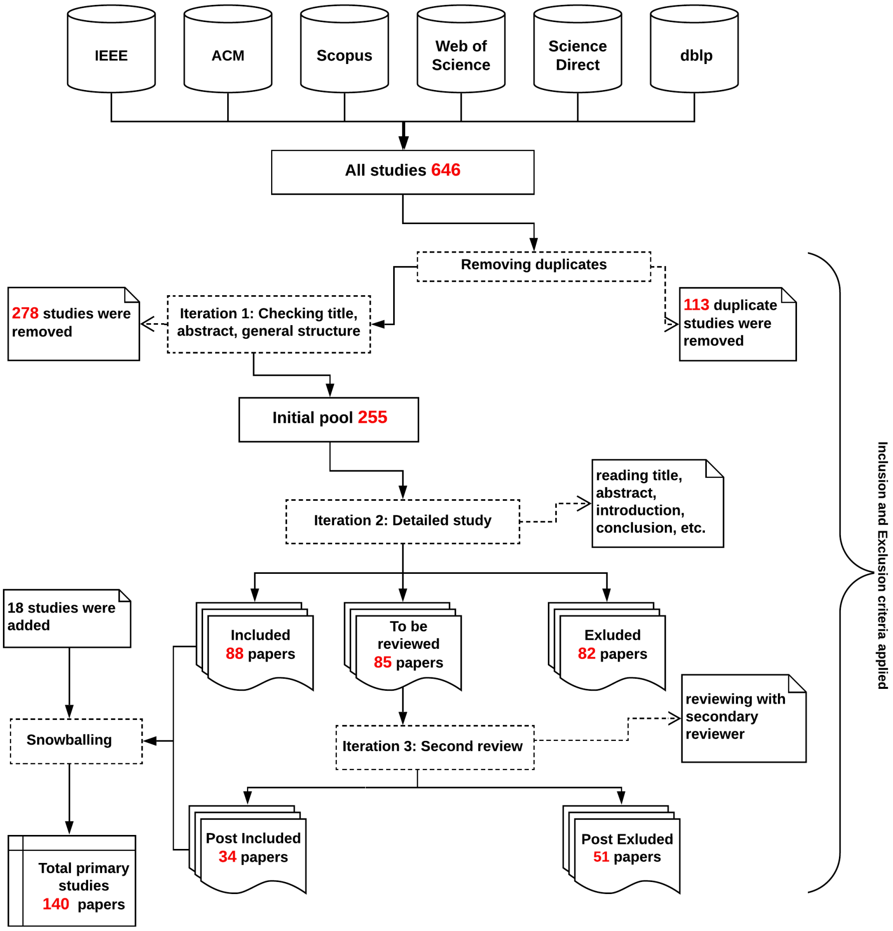
IEEE
ACM
Scopus
Web of Science
Science Direct
dblp
All studies 646
Removing duplicates
278 studies were removed
Iteration 1: Checking title, abstract, general structure
113 duplicate studies were removed
Initial pool 255
reading title, abstract, introduction, conclusion, etc.
Iteration 2: Detailed study
18 studies were added
Included 88 papers
To be reviewed 85 papers
Exluded 82 papers
reviewing with secondary reviewer
Snowballing
Iteration 3: Second review
Post Included 34 papers
Post Exluded 51 papers
Total primary studies 140 papers
Inclusion and Exclusion criteria applied

FIGURE 4. Search and selection process.

sample of primary studies [44], [45] and various approaches including snowballing [42], quasi-gold standard [46], random sampling and margin of error [47] exist to facilitate the identification of the related primary studies. It is also possible to combine these approaches. Conforming to the snowballing guidelines given in [42], the forward snowballing process was

TABLE 4. Data extraction form.

#	Study data	Description	RQ
1	Study ID	unique identifier for the study	-
2	Bibliometric information	Authors' name, Title of the study, Year of publication, Authors affiliated country, number of citations	-
3	Source	IEEE Xplore, ACM, Scopus, Science Direct etc.	-
4	Article type	Conference, Journal, Workshop etc.	-
5	MDE phase	The MDE phase (i.e. system design, simulation, transformation, V&V) the study addressed	RQ 1
6	Tools and Languages	used tools and languages in the study	RQ 2
7	Tools and Languages	developed tools and languages by the study	RQ 3
8	CPS component	The CPS component (e.g. physical component, cyber component, sensor, actuator) the study addressed	RQ 4
9	Quality assessment	problem statement, contribution, future work	QA 1, QA 2, QA 3
10	Self-assessment	Reviewer's level of understanding of the reviewed paper	SA 1

accomplished in this study by determining other papers citing any of the primary studies. We used Google Scholar to find those studies.

Forward snowballing was conducted during the study selection phase. Two iterations during forward snowballing were performed. In the first iteration, we obtained 15 studies after applying the criteria for inclusion and exclusion, and removing the duplicates. Then we made the second iteration on the studies obtained at the end of the first iteration. After applying the same process with the first iteration, the second iteration produced 3 new studies. This resulted in the inclusion of 18 papers to the pool of the primary studies, raising the total of primary studies to **140** papers.

D. INCLUSION & EXCLUSION CRITERIA (SELECTION CRITERIA)

Once all potentially relevant papers are gathered, their relevance must be assessed. Selection criteria are intended for the purpose of identifying those papers (primary studies) directly related to the research questions as suggested in [10]. The inclusion and exclusion criteria must be based on the research questions. These criteria are applied when selecting the primary studies and when performing forward snowballing. To reduce the potentiality of a bias to occur, these criteria should be documented in the protocol definition stage. The selection criteria might be revised during the search process. Inclusion and exclusion criteria are applied to a paper/study by reading sections like title, abstract, introduction, and conclusion.

- a paper is included in the primary studies pool only if it meets all the inclusion criteria and none of the exclusion criteria.

1) INCLUSION CRITERIA

- **IC1:** Study must propose at least one of the MDE approaches or techniques for CPS.
- **IC2:** Study must target CPS or at least one of its application domains.
- **IC3:** Study must be peer-reviewed journal papers, workshop papers or conference papers.
- **IC4:** Models presented by the study must not be used only for documentation and design purposes.
- **IC5:** Paper publication period must be between 2010 and 2018.
- **IC6:** Study must be available in full-text in any of the predefined digital libraries.

2) EXCLUSION CRITERIA

- **EC1:** Study is a secondary study (survey, systematic mapping, systematic review, etc.).
- **EC2:** Study is irrelevant to CPS or any of its application domains and the field of software engineering.
- **EC3:** The study is a summarized version of a complete work already in the SLR pool.
- **EC4:** Study is a kind of educational, editorial, tutorial, or other material (i.e., not a scientific paper).
- **EC5:** Study was written in other languages than English.

E. DATA EXTRACTION

Initially, the final pool of the primary studies is stored in Mendeley. Next, a Google sheet is used for the data extraction stage. The final version of the extracted data is available on IEEE Data Port.[2] In the data extraction sheet, research questions are represented in columns, whereas, primary studies are presented in rows. The process of data extraction in this study goes through 3 phases. Data extraction form is shown in Table 4.

- **Phase 1:** The primary reviewer starts extracting data from the primary studies (answering research questions). Extracted data for each study is represented in a row where each row has a key that refers to the study in Mendeley. Data extraction of each paper is followed by answering quality and self-assessment questions.
- **Phase 2:** The secondary reviewer starts reviewing primary studies with self-assessment score below 50%. After evaluating the study, if the secondary reviewer agrees with the answers given by the primary reviewer, the study is marked as agreed on, else, it goes through phase 3.
- **Phase 3:** In this phase, primary and secondary reviewers discuss the paper disagreed upon in an effort to reach a common ground.

More details on the followed methodology and the analysis of the results can be found in our technical report.[3] It is worth indicating that the technical report investigates all the studies

[2] https://dx.doi.org/10.21227/zbkz-6461

[3] https://arxiv.org/abs/2103.08644

addressing MDE for CPS in a broader perspective. However, the present paper focuses specifically on the tool and language selection in MDE for CPS. To this end, the research questions introduced in this paper take into consideration to obtain findings on MDE activites covered in the studies (RQ1), tools and languages used in MDE for CPS (RQ2), tools and languages developed to apply MDE for CPS (RQ3), and the CPS components addressed using these tools and languages (RQ4).

V. RESULTS

In this section, the research questions are analysed, so the findings obtained according to these questions are reported. Firstly, analysis of RQ1 and a correlation analysis of RQ1 with RQ2 and RQ3 are presented in section V-A, followed by answering RQ2 and RQ3 in section V-C and V-B respectively, and finally the analysis of RQ4 is presented in section V-D.

A. MDE ACTIVITIES

In this section, in addition to answering **RQ1: Which phase(s) of the system development is/are addressed in the study (using MDE)?**, a correlation analysis of **RQ1** with **RQ2** and **RQ3** is carried out to find out the used or developed tools and languages in each of the MDE activities.

Figure 5 shows the reported MDE activities and their use frequencies. As can be seen in Figure 5, the studies differ in the number of the MDE activities they addressed. 72 studies addressed 1 activity, 46 studies reported 2 activities, 17 studies reported about 3 activities, and 5 studies reported 4 activities. Discussion on all these MDE activities is given in the following where they are sorted from most reported to less reported in these studies.

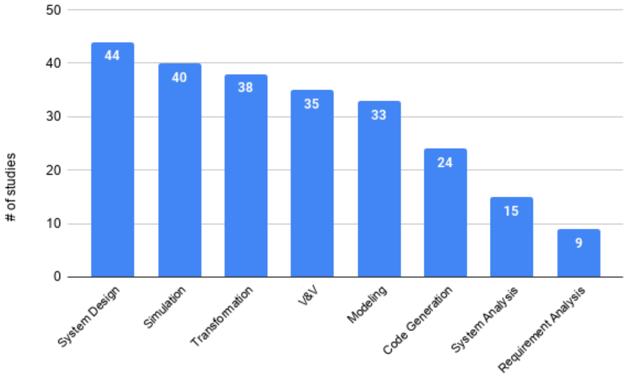

FIGURE 5. Reported MDE techniques/activities.

1) SYSTEM DESIGN

System design is the most reported MDE phase during CPS development; it is reported by 44 studies (18.41%). 15 studies presented DSLs, 14 studies developed metamodels, 4 studies developed tools, 4 studies developed extensions, and the other 7 studies either developed a new modeling approach [48], [49], or combined MDE with existing approaches for CPS development [50]. Table 5 summarizes developed tools and languages and the corresponding papers.

TABLE 5. Developed DSLs, Meta-models, tools, and extensions.

Type	Used framework/language	Relevant studies
DSML	Eclipse Modeling Framework (EMF)	[51], [55], [57], [59]
	Generic Modeling Environment (GME)	[53], [58], [73]–[76]
	Xtext	[54], [56]
	Papyrus tool	[52]
	Xtend	[54]
	Graph Rewriting and Transformation (GReAT)	[74], [75]
	Simulink	[59]
	Web Generic Modeling Environment (WebGME)	[77]
	GEMOC	[78]
	Sirius	[55], [78]
	Kermeta	[78]
Meta-model	Eclipse Modeling Framework (EMF)	[64], [67], [79]
	UML	[62], [63], [65], [66], [80]
	GEMOC	[67]
	Xtext	[67]
	Sirius	[67]
	Kevoree Modeling Framework (KMF)	[81]
	GME (MetaGME)	[61]
	ADOxx	[62]
	SysML	[82]
	ISA95 and ISA88 based	[68]
	-	[60]
	-	[83]
Tool	MetaEdit+	[84]
	ADOxx	[69]
	openArchitectureWare (oAW)	[70]
	-	[85]
Extension	-	[48], [71], [72], [86]

It is worth noting that, some of the studies are just presenting a graphical or textual representation of the system, while another group of studies present a fully functional DSML and its tool, which consist of more than one MDE technique/phase, such as modeling, transformation and code generation. We consider this latter process as the system design. Therefore, in this SLR, we divide the primary studies in those defining only models and the other studies presenting complete design.

Studies which developed DSLs are given as follows: [51] proposed a DSML called CyPhEF that supports the development and validation of self-adaptive CPS. [52] developed a simple graphical DSML for CPS while a DSML for irrigation networks was developed in [53]. In [54], authors developed a DSL that helps in quick construction of co-simulations for CPS, the grammar of the DSL was implemented in Xtext, while the code generation implementation was defined in Xtend. A framework called Advanced Vessel Simulation (AVS) was developed in [55] which supports design and evaluation of racing sailboat simulations. The AVS metamodel was developed in EMF, and Sirius was used for developing the graphical editor. A textual DSL named CHARIOT was created with using Xtext in [56].

A DSL for managing different sensor configurations for a self-driving mini vehicle was developed in [57]. The domain knowledge, static semantics, and the abstract syntax of this sensor management DSL were defined with the Eclipse Modeling Framework (EMF). [58] developed a DSML for the design of networked control systems (NCS) using passivity for separating the NCS control design from uncertainties (i.e. time delays and packet loss). In [59], authors used an ecore-based meta-model to define the abstract syntax of the proposed DSML, and the concrete syntax was implemented as an extension of Simulink standard blocks.

Reference [60] developed two meta-models for representing and sharing incident knowledge of CPS. Meta-models were developed as Eclipse plugins. A metamodel for a systematic analysis of CPS threat modeling was developed in [61] using MetaGME, while [62] developed a metamodel using ADOxx and UML and they used it for the description of an end-to-end communication use case. A meta-model for the development of a smart cyber-physical environment was presented in [63].

Reference [64] developed a meta-model for flexibility and dynamic reconfiguration of the automated production systems by using Eclipse Modeling Framework (EMF). [65] used UML profile to develop a meta-model for modeling cyber-physical assembly systems. Also, in [66], UML class-diagrams were used to develop a meta-model called Smart Environment Metamodel (SEM) to design smart cyber-physical environments. In [67], the authors extended some metamodels of SysML/MARTE for capturing the characteristics of CPS like continuous behavior and stochastic behavior. The approach was implemented in GEMOC. They defined the abstract syntax using EMF, graphical concrete syntax in Sirius, and the textual concrete syntax using Xtext. Meta-models conforming to ISA95 and ISA88 standards were developed in [68] for monitoring the process of an oil production industry.

In [69], they used ADOxx to develop the modeling tool Cyber-Physical Systems for Industry (CPS4I) for the connection of CPS and conceptualizations of industrial applications in integrated models. A tool, called FTOS, based on openArchitectureWare (oAW) was developed in [70], which provides code generation for designing fault-tolerant automation systems. In [71], the authors presented BPMN4CPS which is an extension of BPMN 2.0 for handling CPS features. New extensions for MechatronicUML were developed in [72].

2) SIMULATION

40 of the studies (16.74%) reported simulation. 11 studies addressed exactly the simulation process. They can be summarized as follows: Only 1 study [87] developed a simulator. 2 studies developed meta-models [88], [89], and 8 studies [90]–[97] used existing tools for modeling and simulation. Remaining 29 studies incorporating simulation addressed the other activities (i.e. system design, transformation, V&V, etc.).

Table 6 shows the tools and languages used for the simulation activities. Studies presented different reasons for using simulation, e.g. [51] presented simulation as a feature of the developed DSL and used it for efficiency and time analysis via MECSYCO co-simulation engine. Also [53] presented the simulation as a feature of the developed DSL and used it for performance analysis via MATLAB and EPANET. In [54], authors developed a DSL for constructing HLA-based co-simulations. Reference [58] used Simulink for time and network delay analysis. Reference [74] benefited from Robocode simulator to simulate a reconfigurable conveyor system's behavior and run it in the background (used it as a background simulation) to output time information and the coordinate for the generation of Java animation. Similar to [63], [98] used Simulink for time performance analysis. Reference [99] is another example for utilizing simulations for analysis purposes in which CPS Safety Analysis and simulation Platform (CP-SAP) was developed. Simulations were also used for security experimentation purposes like in [91], [97], [100].

TABLE 6. Reported simulation tools and languages.

Existing tool	Developed tool	Relevant studies
Simulink/Stateflow	-	[5], [58], [76], [93], [95], [97], [98], [100]–[103]
MATLAB	-	[53]
Modelica	-	[104]
ModelicaML	-	[105]
C2WT		[90], [96]
Ptolemy-II	-	[99], [106], [107]
EPANET	-	[53], [92]
-	Meta-model	[88]
MECSYCO	-	[51]
POOSL	-	[54]
Robocode	-	[74]
Embedded Systems Modeling Language (ESMoL)	-	[108]
IOPT-Flow simulator	-	[109]
SystemC	-	[110]
CPGAME	-	[94]
Verilog-AMS	-	[111]
-	Smart Grid Simulator (SGS)	[87]
-	Meta-model	[89]
CPAL	-	[112]
DEVS-Suite Simulator	-	[6]
JSBSim	-	[91]
ScicosLab	-	[91]
CIF	-	[113]

3) TRANSFORMATION

38 studies (15.90%) presented transformations (listed in Table 7). 30 studies covered one transformation type (either M2M or M2T), 3 studies considered two transformation types, while 1 study [114] showed 3 different transformation types namely M2M, M2T, T2M. the transformation types presented by the other 4 studies was not clarified. Therefore, 39 transformations were presented in total and they are as follow: 28 M2M transformations, 10 M2T transformations, and one T2M transformation. Studies implemented M2M transformations can be categorized into two:

First category covers the studies using existing tools and languages. [115] used Y2U tool to transform Statechart models to UPPAAL timed automata model. [116] presented M2M transformation by transforming Simulink simulation models to AADL architectural models using Assisted Transformation of Models engine. AADL and Modelica were used in [117], [118], where both Modelica and AADL were transformed to each other. In [119], authors used Critical Infrastructure

TABLE 7. Reported model transformations, and used transformation tools and languages.

Transformation	Existing tool	Developed tool	Relevant studies
M2M	-	metamodel	[124]
	Y2U tool	-	[115]
	AST Engine	-	[116]
	QVT	-	[98], [114], [121]
	AADL,Modelica	-	[117], [118]
	CIP VAM UML	-	[119]
	EXTEND	-	[70]
	GreAT	-	[74], [75], [123]
	UML4IoT	-	[65]
	-	UPP2SF	[101]
	ME+ tool	-	[130]
	-	STU	[5]
	GenERTiCA	-	[50]
	-	ECPS Verifier	[125]
	Xtext	-	[122]
	GraphML	-	[120]
	-	AS2T	[126]
	-	Modana	[127]
	Other	-	[59], [131]–[134]
M2T	Acceleo	-	[57], [98]
	Xtend	-	[114], [122]
	ATL	metamodel	[129]
	IOPT tools	-	[128]
	GenERTiCA	-	[50]
	Other	-	[48], [64], [110]
T2M	Xtext	-	[114]
Other			[6], [135], [136]

Protection - Vulnerability Analysis and Modeling (CIP VAM) UML profile to transform UML models to Bayesian Network (BN) models. In [50], the authors transformed UML models to Distributed Embedded Real-time Compact Specification (DERCS) models with using GenERTiCA. In [120], the authors transformed a SysML model to a graph by employing GraphML. Other studies include: [98], [114], [121] used QVT, [122] implemented M2M transformation using Xtext, [74], [75], [123] used Graph Rewriting and Transformation (GReAT) for M2M transformation, while EXTEND is used for the M2M transformation in [70].

Studies which developed metamodel, tool, or language for the M2M, M2T, and T2M transformations, constitute the second category. In [124], they proposed a transformation method that transforms Simulink model to ECML model by designing metamodels for both Simulink and ECML. [101] developed the model translation tool UPP2SF that transforms UPPAAL timed automata models to Simulink/Stateflow. On the contrary, [5] developed a tool named STU that translates Simulink/Stateflow model into UPPAAL timed automata model. In [125], the authors developed a tool named ECPS Verifier that was used for the transformation of AADL models to UPPAAL timed automata. In [126], they presented a tool named Simulink/AADL Translator Tool (AS2T) that automates the transformation of the simulation models of Simulink to AADL models. Reference [127] presented a framework called Modana that helps transforming SysML and MARTE models into Reactive Modules Language (RML) and Modelica models.

The remaining studies presented the implementations of M2T transformations using existing tools like Acceleo [57], [98], Xtend [114], [122], IOPT tools [128], and GenERTiCA [50], except for one study [129] that presented meta-models of HybridUML and Quantified Hybrid Program (QHP) and then they used ATLAS Transformation Language (ATL) for defining the transformation rules. Reference [114] was the only study presenting T2M transformation using Xtext.

4) VALIDATION AND VERIFICATION (V&V)

35 studies (14.64%) reported V&V activity. Only 2 studies developed a tool [110], [136], and 1 study developed an ontology [137]. Table 8 presents studies which implemented V&V as part of their work.

Studies using UPPAAL for verification include the followings: [115] used UPPAAL to formally verify the safety properties of a medical guideline. A Domain-specific model checking (DSMC) for MECHATRONICUML using UPPAAL model checker was presented in [114]. A pacemaker was modeled and verified using UPPAAL in [101]. In [5], the authors used the UPPAAL tool for the verification of SIMULINK/STATEFLOW models after being transformed to UPPAAL timed automata. In [125], UPPAAL was used for the formal verification (i.e. model checking) of AADL models.

Other tools and languages used for model checking for verification include the following: Simple Promela Interpreter (SPIN) model checker was used in [75] to verify the Promela code. Also, in [138], SPIN was used as a model checker to verify the PrT net models after translating it to a Promela code. In [139], they used a probabilistic model checker called PRISM. The authors in [140] verified their protocols via timed model checking MECHATRONICUML.

Simulink/Stateflow was used in [102] to verify supervisory controllers for hierarchical systems. Simulink Design Verifier (SLDV) was used for the verification of the simulation models in [141]. In [142], they used SLDV to verify the behavioral models developed in Simulink in order to meet the requirements modeled. Furthermore, Object Constraint Language (OCL) was used for the verification of the static semantics of a meta-model presented in [57]. Also, in [51], OCL was used for defining and validating metamodel constraints. In [129], a verification of KeYmaera-QHP code in KeYmaera, a hybrid verification tool, was presented. In [108], the authors used FORMULA for metamodel analysis and verification. Frama-C was used in [143] to prove and verify a developed C code library. In [142], Assume Guarantee REasoning Environment (AGREE) tool was utilized to verify that the AADL architectural models satisfy the system requirements.

Studies, presenting validation, are summarized as follows: [69] developed a modeling tool (CPS4I) and a modeling

TABLE 8. Reported V&V tools/languages which are used or developed in primary studies.

Used Tool/Language	Developed Tool/Language	Relevant studies
UPPAAL	-	[5], [101], [114], [115], [125], [145]
Simulink Design Verifier (SLDV)	-	[141], [142], [146]
SPIN (Simple Promela Interpreter)	-	[75], [138]
Object Constraint Language (OCL)	-	[51], [57]
Clock Constraint Specification Language (CCSL)	-	[147]
Open Models Laboratory (OMiLAB)	-	[69]
KeYmaera	-	[129]
CHECK	-	[70]
-	Ontology	[137]
Linux Driver Verification tool (LDV)	-	[148]
FORMULA	-	[108]
EAST-ADL	-	[84]
-	SHARC	[110]
Web Generic Modeling Environment (WebGME)	-	[77]
Protégé	-	[13]
Simulink	-	[102]
PIPE+	-	[138]
Frama-C	-	[143]
PRISM	-	[139]
MECHATRONICUML	-	[140]
AGREE	-	[142]
-	BPMN4CPS Tool	[136]

method (SeRoIn) then validated them using Open Models Laboratory (OMiLAB). CHECK validation language was used in [70] to formulate tests for the detection of semantic design errors in the developed models. A generated code in [144] was tested and analyzed using Frama-C. In [113], the authors implemented simulation-based validation.

A tool, named Simulation and Verification of Hierarchical Embedded Systems (SHARC), was developed in [110] for the verification of the behavior of automotive safety-critical systems. In [137], the authors developed an ontology and used it as the validation mechanism.

5) MODELING

33 studies (13.81%) reported about modeling. This category encompasses studies which used existing languages/tools for modeling, wherein studies which developed a language/tool for modeling were included in the group of system design. Only 4 studies [148]–[151] did not report any tool, instead, they either proposed an approach for modeling CPSs or used equational models. Table 9 shows various languages/tools used for modeling by the studies.

In [107], [141], authors used Ptolemy II to model Medical CPS, while in [82] the behavioral model of a production nominal resource was modeled using Ptolemy II. In [106], authors modeled a Holter Monitor. They used Ptolemy II to model the device's functionality and UPPAAL for modeling system's state space and the transitions between them.

TABLE 9. Reported tools/languages used for modeling and the studies used them.

Tool/Language	Relevant studies
Ptolemy II	[82], [106], [107], [141]
Petri nets	[132], [138], [152], [153]
AADL	[105], [117], [142], [154]
YAKINDU Statechart Tools	[115], [145], [155]
Papyrus tool	[98], [121], [156]
UML	[13], [154]
Modelica	[105], [154]
Simulink	[142], [146]
finite state machine (FSM)	[112], [157]
SysML	[67], [98]
HybridUML	[129]
MARTE	[67]
UPPAAL	[106]
GME (Generic Modeling Environment)	[158]
MetaEdit+	[130]
web ontology language (OWL)	[160]
ADVISE Meta tool	[161]

A modeling approach called time-constrained aspect-oriented Petri net was presented in [152]. The approach combines discrete/continuous Petri nets and aspect-orientation for modeling CPS. In the work presented in [132], colored Petri nets were extended to probabilistic colored Petri nets for modeling and analyzing CPS attacks. Petri nets were also used in [153] for modeling smart grid threats.

View oriented approach was adopted in [117] for the description of different aspects of an aerospace CPS. Modelica was used for modeling the overall architecture of a lunar rover robot and the lunar rover robot's body structure model while AADL was used for modeling the navigation system of the lunar rover. Similarly, authors in [154] integrated AADL, UML and Modelica to model the requirements of a vehicular ad-hoc network. Further, in [105], AADL and Modelica were integrated to model big data-driven CPS. In [142], Simulink/Stateflow was adopted to model a generic patient-controlled analgesia infusion pump system for analyzing logical requirements and behaviors, while AADL was used for developing the architectural model of the system.

YAKINDU statechart tools was adopted in [115] to model and simulate a stroke statechart model. Likewise, in [145], they used Yakindu statecharts for the modeling of a simplified cardiac arrest. The study in [155], created a UML statechart model for an envisioned CPS scenario using YAKINDU statechart modeling tool again. In [121], Papyrus tool was used for creating the UML models. The authors in [156] presented a methodology for knowledge representation of CPS using the modeling tool Papyrus. Reference [13] used UML for defining the dependability analysis models. Implementations of modeling CPS using HybridUML was presented in [129].

Finite state machine (FSM) was adopted in [157] to model the behavior of automation components. [112] used FSM to describe the logic of a servo tester. GME was used in [158] to build Lathe CNC System models and export models' data as an XML file. ASLan++ was used in [159] for modeling water treatment plant and attack model. Reference [150] presented a new formalism named Stochastic Occurrence Hybrid

TABLE 10. Reported code generation languages/tools and studies used them.

Existing Tool/Language	Developed Tool/Language	Relevant studies
Simulink coder	-	[5], [101], [142]
-	I2C4IOPT	[163]
Acceleo	-	[55]
Clock Constraint Specification Language (CCSL)	-	[147]
Xtend	-	[54]
-	Code generator	[58], [144], [158]
XPand	-	[70]
Scenario Modeling Language (SML)	-	[162]
-	ISA88 editor	[64]
Embedded Systems Modeling Language (ESMoL)	-	[108]
IOPT-Flow tool framework	-	[109]
-	Interpreter	[75]
Generation of Embedded Real-Time Code based on Aspects (GenERTiCA)	-	[50]
KMF	-	[81]
Rhapsody code generator	-	[103]
HA-SPIRAL	-	[164]
Compositional Interchange Format (CIF)	-	[113]
GeneAuto	-	[143]

Automata and a modeling approach to model the stochastic behavior in CPSs.

6) CODE GENERATION

24 studies (10.04%) reported about language/tools used or developed for code generation purposes. Table 10 lists used and developed languages/tools for code generation. These studies are categorized here into studies which used existing tools for code generation, and studies which developed new code generation tools.

In [101], they used Simulink Real-Time Workshop Embedded Coder (RTWEC) to generate C code from a pacemaker Stateflow chart. Likewise, C code and VHDL code were generated in [5], [142] from the Stateflow models using Simulink coder. Moreover, a tool named GeneAuto was presented in [143] that generates C or ADA code from Simulink models. In [108], built-in code generator for Embedded Systems Modeling Language (ESMoL) was used to generate functional C code. IOPT-Flow tool framework was used in [109] to generate C and Javascript code or VHDL hardware descriptions.

Programmable Logic Controllers (PLCs) Code Generation was presented in [162] using Scenario Modeling Language (SML). In the same manner, implementation of PLC code generation was presented in [113] using Compositional Interchange Format (CIF). Clock Constraint Specification Language (CCSL) constraints were utilized in [147] for code generation purposes. The authors in [54] used Xtend for the support of code generation for OpenRTI. Kevoree Modeling Framework (KMF) was used in [81] for the generation of Java API in order to create and manipulate the runtime models.

In [163], they developed a tool named I2C4IOPT for automatic code generation of globally asynchronous and locally synchronous systems (GALS) - supported by Arduino boards. An ISA88 editor was implemented in EMF in [64] to generate a programmable logic controllers (PLC) control code. A code generator was developed in [58] for the generation of Simulink models and network-scripts. In [144], a model-based code generator for medical CPS was presented. An interpreter was developed in [75] to translate finite-state machine (FSM) models and constraints into Promela code.

7) SYSTEM ANALYSIS

15 studies (6.28%) reported language/tools which used or was developed for system analysis reasons. Table 11 lists the used and developed languages/tools for system analysis. Studies can be categorized into ones directly using existing tools for system analysis, and others developing new tools for system analysis purposes.

In [165], meta-models for operational analysis and system analysis were developed. They also used TTool for safety analysis. A knowledge-based approach using Failure Models, Effects and Criticality Analysis (FMECA) techniques was presented in [166]. The authors first modeled FMECA using UML class diagram, then the FMECA metamodel was expressed in Protégé, which was then used to build an ontology-based KB. A metamodel was developed to enable the management of application requirements and business constraints for CPS in [167]. CPS meta-model for knowledge formalization was presented in [168], where they also implemented formal concept analysis in their work. CPS Safety Analysis and Simulation Platform (CP-SAP) was developed in [99] for Human-machine interaction (HMI) safety analysis of CPS. A framework called Modana was presented in [127] that aims to model and analyze the non-functional aspect (i.e. time, energy, etc) of Energy-Aware CPS.

In [152], a modeling approach based on discrete/continuous Petri nets was proposed for schedulability analysis. Also, a modeling approach was presented in [132] that supports both qualitative and quantitative analysis of CPS attacks using probabilistic colored Petri nets. CPS dependability analysis was presented in [169] using Stochastic Petri Net (SPN). An approach for specification and analysis of automotive CPS was presented in [154] where Modelica was used for analyzing engine model and AADL was used for End to End Delay Analysis from brake-pedal to throttle actuator. Security analysis tool CL-AtSe was used in [159] for analyzing and discovering potential attacks on Industrial Control Systems.

B. DEVELOPED TOOLS/LANGUAGES FOR APPLYING MDE ON CPS

In this section, the results and findings of **RQ2: Is/Are there any tool and/or DSL developed for MDE of CPS by the study?** and its following sub-questions are presented.

TABLE 11. Reported System analysis languages/tools.

Existing Tool/Language	Developed Tool/Language	Relevant studies
EAST-ADL	-	[130], [170]
Petri nets	-	[132], [152], [169]
AADL	-	[154]
Modelica	-	[154]
-	Metamodel	[165]–[168], [171]
-	CP-SAP	[99]
-	Modana	[127]
CL-AtSe	-	[159]
TTool	-	[165]
HiP-HOPS tool	-	[130]

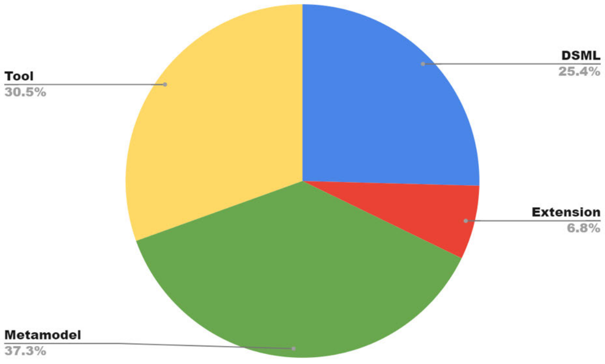

FIGURE 6. Frequency and type of the developed tool/language.

RQ2.1: If any tool and/or language is developed in the study, is it reported?

59 studies developed DSLs/DSMLs, metamodels, tools, or their extensions. 22 of these studies developed metamodels, 15 studies developed DSLs/DSMLs, 18 studies developed tools (including 2 frameworks and 1 platform), and 4 studies presented extensions for other tools/languages as shown in Figure 6. These developed tools/languages were addressed in a detailed way in the correlation analysis performed between RQ1 with RQ2 and RQ3 which is presented previously in section V-A.

RQ2.2: Is/Are the developed tool(s) and/or languages available and/or accessible?

From the abovementioned 59 studies which consider the development of a tool/language, only 10 studies [5], [51], [60], [67], [75], [110], [116], [125], [136], [163] provided public access (mostly with a web link) to the developed tool/language.

RQ2.3: What is/are the framework(s) or programming language(s) for its/their development?

As can be seen in Figure 7, UML is the most used language, followed by EMF and GME. Figure 7 shows the correlation between RQ3.1 and RQ3.3. It is clear that UML is mostly used for building metamodels, where EMF is used for building both metamodels and DSLs, and GME is mostly used for building DSLs. Other presented tools are; GreAT that is used alongside with GME [74], [75], Sirius is used for building the graphical concrete syntax [55], [67], [78] and finally, Xtext is used for developing DSL grammar in [54], and for building the textual concrete syntax as in [67]. These tools/languages are listed in Table 5 and discussed in detail in section V-A.

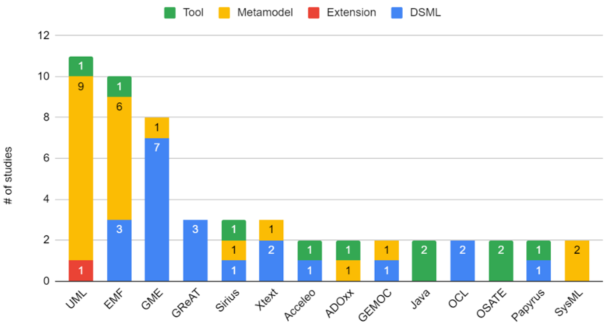

FIGURE 7. Used languages/frameworks for developing DSMLs, metamodels and/or tools.

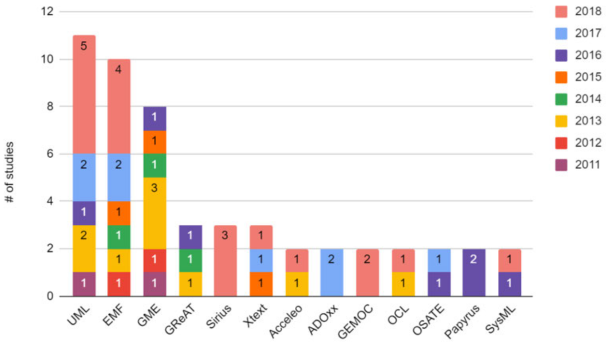

FIGURE 8. Distribution of languages/frameworks used for developing DSMLs, metamodels and/or tools over publication years.

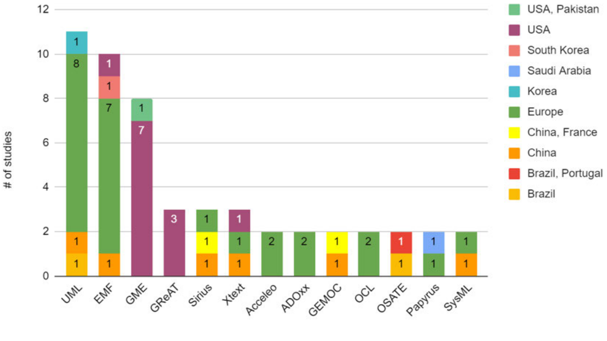

FIGURE 9. Distribution of languages/frameworks used for developing DSMLs, metamodels and/or tools over author affiliation countries.

GME seems to be the third most used framework for building DSLs and metamodels. However, it is worth mentioning that the results of distributing RQ3.3 over the publication years (see Figure 8) show that GME is not used for the last 2 years (2017 and 2018) of the examined period by any of the primary studies. Further, the results of RQ3.3 were distributed over authors' country of affiliation as depicted in Figure 9. The study found out that GME and its tool GReAT were only used by authors/researchers affiliated to the USA, where on the other hand, UML and EMF were mostly used by authors/researchers affiliated to Europe.

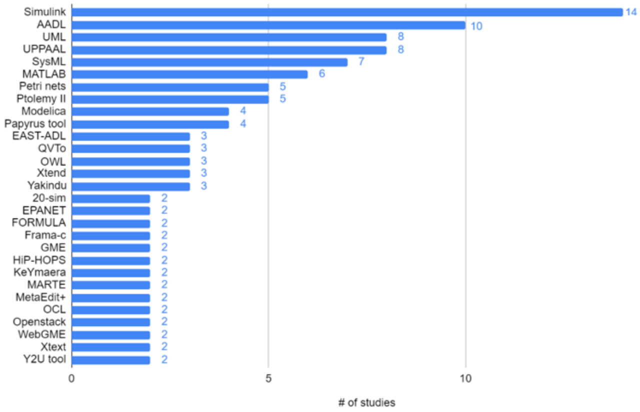

FIGURE 10. Most used tools/languages which are at least reported by 2 studies.

C. USED TOOLS/LANGUAGES FOR APPLYING MDE ON CPS

In this section, the results and findings of **"RQ3: Is/Are there any tool(s) and/or Language(s) used to apply MDE in/for CPS in the study?"** and its following sub-question are presented.

RQ3.1: Which tool(s) and/or language(s) is/are presented/used in each phase of the system development?

Figure 10 shows tools and languages used by the primary studies. It is worth indicating that a correlation analysis between RQ1 with RQ2 and RQ3 has been already presented in section V-A for better understanding of the tools/languages, and to give the reader a clear idea about the MDE phase/activity in which the tool/language was used for. Therefore, in this section, only the most used tool/languages are briefly discussed.

Our study found that Simulink is the most used tool. Majority of the reviewed studies used Simulink for simulation purposes, listed in (Table 6). Simulink was also used for modeling [142], [146]. Simulink coder was used for code generation purposes in [5], [101], [142] while Simulink Design Verifier (SLDV) was adopted for the verification of models [141], [142], [146]. AADL follows Simulink as the second most used tool. It was used for modeling the cyber part of the systems [117], developing architectural models [142], or for system analysis [154].

UML is used by various studies for building metamodels, listed in Table 5. It was also used for modeling activities like defining dependability analysis models [13]. The vast majority of the studies used UPPAAL for verification, see Table 8. For instance, [106] used UPPAAL for modeling system's state space and the transitions between them.

D. ADDRESSED CPS COMPONENTS

In this section, the results and findings for **"RQ4: What is/are the CPS component(s) addressed in the study?"**, and a correlation analysis of RQ1 with RQ4 are presented.

According to [172], a CPS mainly consists of 5 components, which are, Physical components, Cyber components, Sensors, Actuators, and Network. Amongst the 140 primary studies covered in this SLR, only 6 papers were left undetermined (the addressed CPS component by these papers could not be determined) and 9 studies addressed more than 1 CPS component. Figure 11 shows the categories of CPS components. The full list of studies and their supporting CPS components is given in Table 12.

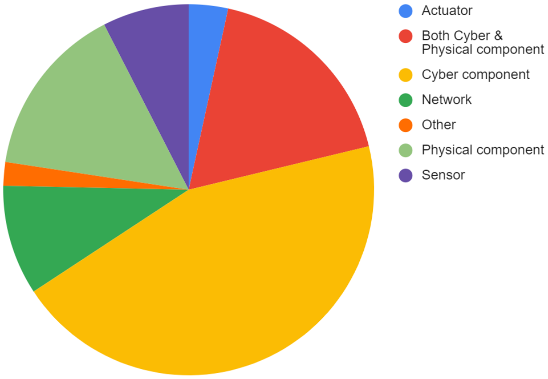

FIGURE 11. Reported CPS component(s).

- Cyber Component: 65 studies (44.5%) addressed this component, i.e. the software aspect of the system. Examples are Controllers (e.g. [52], [163]), Development Artifacts related with Transformation [54], Simulation [124], System verification [129], System behavior covering timing behavior [115], System safety properties [147], and System requirements [169], [173].
- Physical Component: Reported by 22 studies (15.1%). These studies addressed the physical and hardware components of the system, e.g. Physical Dynamics (environment behavior) [80], [150], Power plant [132], [174], [175], and Hardware [105], [130].
- Both Cyber & Physical components: Reported by 26 studies (17.8%). This category contains the studies discussing modeling both cyber and physical aspects of the system. [176] reported about modeling a controller (cyber component), and a plant (physical component). Another example is [117] where the authors modeled a lunar rover robot's body (physical component) and its navigation system (cyber component).
- Network: Reported by 14 studies (9.6%). Studies in this group addressed issues like; sensor networks [75], network security [62], physical attacks [161], security requirements and attacks [121].
- Sensors: 11 studies (7.5%) reported this component. Studies in this group addressed the different operations of sensors, like sensor design [116], sensor management [57], sensor data analysis [144], and sensor failures [177].
- Actuators: Reported by only 5 studies (3.4%). This component is less addressed one compared to the other CPS components. For instance, [116] covered actuator modeling and design, while [13] discussed actuator failure.
- Other: Studies, which do not fit any of the above categories, are grouped under this category. They consider

TABLE 12. Addressed CPS components and the corresponding studies.

Category	# of studies	Relevant studies
Cyber component	65	[5], [6], [50], [52], [54]–[56], [59], [69], [70], [73], [74], [78], [82], [83], [85], [90], [92], [93], [95], [97], [99], [104], [107]–[109], [112]–[115], [118], [120], [122]–[126], [128], [129], [131], [133], [134], [136], [137], [140], [142], [143], [146]–[149], [155], [157], [158], [160], [162]–[164], [169], [170], [173], [179]–[182]
Both Cyber & Physical components	26	[49], [61], [64], [67], [71], [76], [81], [91], [96], [98], [117], [127], [135], [138], [139], [141], [151], [153], [154], [156], [166]–[168], [171], [176], [183]
Physical component	22	[51], [65], [68], [77], [80], [84], [89], [101], [102], [105], [106], [110], [111], [130], [132], [145], [150], [159], [174], [175], [184], [185]
Network	14	[53], [58], [60], [62], [63], [66], [75], [87], [92], [94], [100], [119], [121], [161]
Sensor	11	[13], [53], [57], [63], [66], [73], [75], [116], [122], [144], [177]
Actuator	5	[13], [63], [66], [116], [122]
Other	3	[79], [86], [178]

business processes [86], workflows (process) [79], and data [178].

Further, in this SLR, a correlation analysis of the MDE activities and CPS components is scrutinized so as to provide an understanding of the addressed MDE activities in each CPS component development, (see Table 13). Despite the fact that the correlation analysis cannot indicate the CPS domain wholly, one can see that, for instance considering the Cyber component, most research works concentrated on Transformation (22 studies), V&V (18 studies), Simulation (17 studies), Code generation and System design (16 studies each), while Requirement analysis and System analysis were addressed only by 4 and 3 studies, respectively.

Similarly, for the Physical component, the research work converged on Simulation (10 studies), System design and V&V (8 studies each), and Modeling (7 studies) while System analysis and Requirement analysis are again less addressed (3 studies and 1 study respectively). In terms of the sensor component, most research works concentrated on System design (6 studies), however, System analysis for sensors was not addressed by any study. Regarding the actuator component, which is the least addressed CPS component, it is interesting to note that Code generation, Simulation, Requirement analysis, and System analysis were not addressed by any study.

Finally, taking into consideration the CPS components, one can also deduce from above results that the Actuators are currently the least studied components during MDE of CPS. Furthermore, Requirement analysis and System analysis are the least addressed activities for developing all CPS components.

VI. DISCUSSION AND THREATS TO THE VALIDITY

In addition to the detailed assessments given in the previous section for the results of the conducted SLR, this section includes a more general discussion of the achieved findings along with its implications. Threats to the validity of the study are also discussed in this section.

Regarding MDE activities/techniques addressed (RQ 1), the most considered MDE activity was system design. Researchers developed DSLs (15 studies), metamodels (22 studies) and tools (18 studies) for this purpose. The results show that the total number of studies developed DSML is quite low (10.71%, only 15 studies out of 140) for a wide and complex domain like CPS. Furthermore, considering the fact that though general-purpose modeling languages such as UML and SysML have rich tool support, they tend to be too general to consider domain-specific aspects, and they also lack the detailed formal semantics needed for formal analysis. On the contrary, DSLs are specialized modeling languages that are developed according to the needs of a specific application domain. Consequently, they constitute a good foundation for domain-specific formal analysis and automated tool support. Further, DSLs tend to use notations domain experts are familiar with, which in return makes them mostly accepted. [52], [186]. Therefore, one can deduce that there is an opportunity for conducting more research to design DSLs to address different aspects of CPS development life-cycle and to cover as much application domains as possible. DSLs can provide a higher level of abstraction for complex systems such as CPS which may lead to increase the performance and to decrease the time and the cost of CPS development. Simulation was the second most reported MDE activity (40 studies, 16.81%). Apart from 1 study that developed a tool [87] and 2 studies that developed metamodels [88], [89] for simulation purposes, the rest of the studies (37 studies) used existing simulation tools and languages.

Furthermore, analysis of the results for RQ 1 showed that M2M transformation gains more attention in terms of the existing/developed tools and languages in comparison with the other transformation types, M2T and T2M. In addition, it is observed that languages like GenERTiCA and Xtext were used for the implementation of more than one transformation type. Also, it is worth mentioning that tools like UPP2SF and STU can be used as complementary tools for M2M transformation. Other complementary languages for modeling CPS are Modelica and AADL, where Modelica is used for modeling the physical world and AADL for modeling the cyber components, and the transformation between these two languages do not require any third-party tool or language [117], [118]. V&V was reported by 35 studies (14.71%,). However, apart from 2 studies [110], [136] which developed a new tool and one study that developed an ontology [137] specific for CPS, the rest of the studies preferred using the existing general purpose V&V tools.

TABLE 13. MDE activities addressed for the development of each CPS component.

	Code generation	Modeling	Requirement Analysis	Simulation	System Analysis	System Design	Transformation	V&V
Actuators		[13]				[63], [66]	[116], [122]	[13]
Both Cyber & Physical component	[64], [81]	[98], [117], [138], [141], [151], [153], [154], [156]	[139], [183]	[76], [91], [96], [98], [176]	[127], [154], [166]–[168], [171], [183]	[49], [61], [64], [67], [71], [76], [81]	[64], [98], [117], [127], [135]	[138], [139], [141], [176]
Cyber component	[5], [50], [54], [55], [70], [108], [109], [113], [142], [143], [147], [155], [158], [162]–[164]	[82], [107], [112], [115], [129], [142], [146], [148], [149], [155], [157], [158], [160]	[142], [173], [181], [182]	[5], [6], [54], [74], [90], [92], [93], [95], [97], [99], [104], [107]–[109], [112], [113], [140]	[99], [169], [170]	[50], [52], [54]–[56], [59], [69], [70], [73], [74], [78], [82], [83], [85], [179], [180]	[5], [6], [50], [59], [70], [74], [104], [114], [115], [118], [120], [122]–[126], [128], [129], [131], [133], [134], [136]	[5], [69], [70], [107], [108], [113]–[115], [125], [129], [136], [137], [140], [142], [143], [146]–[148]
Network	[58], [75]	[121], [161]	[121]	[53], [58], [87], [92], [94], [100]		[53], [58], [60], [62], [63], [66], [75], [100]	[75], [119], [121]	[75]
Physical component	[101], [111]	[105], [106], [130], [132], [145], [150], [159]	[175]	[51], [89], [101], [102], [105], [106], [110], [111], [175], [185]	[130], [132], [159]	[51], [65], [68], [77], [80], [84], [174], [184]	[65], [101], [110], [130], [132]	[51], [77], [84], [101], [102], [110], [145], [185]
Sensors	[75], [144]	[13]	[177]	[53]		[53], [57], [63], [66], [73], [75]	[57], [75], [116], [122]	[13], [57], [75], [144]

Results for R.Q 2.1 showed that the top 10 most used tools and languages in the field of applying MDE paradigm on CPS are; Simulink, AADL, UML, UPPAAL, SysML, MATLAB, Petri nets, Ptolemy II, Modelica and Papyrus. Investigation for RQ 3.1 revealed that 59 studies out of 140 developed DSL, metamodel or tool. However, only 10 of the 59 studies reported the availability of these developed tools and languages (i.e. tools can be downloaded via the links given in their papers) according to the results for RQ 3.2. The CPS research community can consider increasing the number of the publicly available MDE tools to facilitate the dissemination of these tools and provide the adoption of them in various industry fields. The results of RQ3.3 revealed that UML, EMF, and GME were the most used tools and environments since they were used in 59 different studies while developing various DSLs, metamodels, and tools. This is somehow expected due to the widespread use of UML during system modeling in different application areas as well as many popular modeling language creation workbenches are EMF-based. However, GME was not present for the last 2 years (2017 and 2018) in any study. Findings also exposed that GME was mostly used by the researchers affiliated to USA, while UML and EMF were mostly used by the researchers affiliated to Europe.

Regarding the addressed CPS components during MDE of CPS (RQ4), one of the interesting results shows that most of the primary studies consider only the cyber and physical components of CPS during MDE. There is a limited number of work also focused on the components such as sensors, actuators and networks. This may have several reasons. For instance, some of the studies (e.g. [64], [68], [102], [132], [175]) represent the sensors, actuators, network elements, etc. under the umbrella of the physical plant, hence they do not include models specialized for these components other than the physical model. Another reason can be modeling the systems with the all details covering such as sensors and actuators may lead to very large and complex models (to be used in MDE) especially when we consider big CPS and that can make the application of MDE less efficient and almost not feasible.

This study is meant to provide practitioners with insight into which languages/tools are most commonly used when applying MDE techniques for CPSs development, and also to help them easily identify how these languages/tools have been used in the various MDE techniques/activities. Moreover, based on the findings given above, the conducted SLR may guide the researchers in shaping their future work on the MDE of CPS by taking into account the following open issues:

- The distribution of MDE techniques over the targeted CPS components shown in Table 13 sheds light on the areas where the research efforts are concentrated and areas where potential research can be undertaken. For instance;
  - Modeling sensor and actuator components of the CPSs and generation of the executable artifacts from the corresponding models can be investigated since the current studies on them are so rare.
  - Similarly, the MDE research on the requirement and system analysis activities as well as simulation for creating sensors and actuators components needs to be increased which may improve the coverage and functionality of the design models for CPS.
- DSLs and domain-specific simulation and verification tools specific for CPS are currently rare. Using such languages and tools can help the developers while dealing with the complexity of CPS by working in a higher level of abstraction.
- Although the SLR in here both determined the languages/tools and revealed how they are utilized during MDE of CPS, additional work on comparing these

languages/tools possibly in a qualitative or quantitative manner may help the formalization of a detailed guide for the language/tool selection in the field of MDE for CPS.

- The reasons for the decline of Generic Modeling Environment (GME) may be further investigated since our results showed that although GME is constantly used from 2011-2016 in the primary studies, it has not been used for the last 2 years (2017 and 2018) of the examined period.

THREATS TO THE VALIDITY

Threats to the validity for this SLR study are classified according to categories proposed in [187], and hence they include four types, namely construct, internal, external and conclusion validity threats.

Construct Validity: It represents how the SLR study truly reflects the intent of the researchers, and what is asked by the research questions. To define the research questions, it is important to stress that the process proposed by [38] and [39] and guidelines defined by [10] were followed in this study.

Furthermore, another aspect of construct validity is to assure that all relevant studies on the selected topic are found adequately. The possibility of missing primary studies is a common threat to the validity of any SLR. For instance, although CPS is a well-known and widely-used concept, keywords such as ''embedded systems'' and ''real-time systems'' might be used previously to refer to the systems that are also CPS. However, we believe that omitting such keywords caused a negligible risk in this SLR especially considering the publication years covered. Thus, the terms listed in Table 3 are sufficiently good enough to be used as keywords to find the most appropriate studies as planned at the beginning of the study. Moreover, to mitigate this risk, search strings were also formed through several iterations which also enabled the adequate coverage of the literature. Another option for selecting primary studies can be starting the search directly from major venues including journals and conferences which are well-known in the CPS research. Such a search strategy may be effective in determining the significant studies in the related field at first sight which may pave the way of e.g. preparing in-depth survey of studies mostly in a more specific field. For instance, that selection approach can be used in providing a survey and evaluation of model transformations performed during MDE of CPS. However, it causes gathering relatively a small set of studies and many studies published in other venues may be neglected when we consider SLRs instead of surveys. Hence, as suggested in the SLR guidelines given in [10] and mostly applied in the current SLRs covering other research domains, we also preferred searching general publication digital libraries/databases to determine the primary studies using the search strings which leads performing the SLR both transparent and replicable. It is also worth indicating that these digital libraries already index most well-reputed publication venues and the list of publication venues included in our online repository for this SLR verifies that the coverage of the search is enough within this context. Furthermore, to improve the results, the forward snowballing sampling method was used, and it has proved to be effective.

Internal Validity: This relates to the degree to which the design and the conduct of the SLR study are likely to prevent systematic errors. Internal validity is a prerequisite for external validity [10]. Therefore, both qualitative and quantitative analysis were used to minimize threats. The use of a rigorous protocol and data extraction form mitigates this kind of threats to validity. Moreover, threats originating from personal bias or lack of understanding of the study were reduced by conducting data extraction phase iteratively. For this purpose, one researcher extracted data from the primary studies and answered quality and self-assessment questions. The other two researchers (expert in CPS and MDE) reviewed the extracted data from studies with low self-assessment rates under 50%.

External Validity: According to [187], external threats concern the generalizability of the SLR results, that is, the degree to which the primary studies is representative of the reviewed topic. In this study, the set of primary studies may not be representative of the entire set of existing studies on the topic, MDE for CPS. However, this threat was mitigated as follows; Firstly, the search strategy consisted of manual and automatic search, then followed by the forward snowballing. The forward snowballing enabled finding studies which were not captured by the search strings in the digital libraries. Secondly, the inclusion and exclusion criteria of the protocol created in this study support refining the set of primary studies which leads to include only studies which meet the topic. Only studies in English were included. Papers written in other languages concerning the same topic may exist. However, this threat is considered as having minimal effect.

Conclusion Validity: All relevant primary studies may not be identified [10]. To alleviate this threat, the research protocol of this study was designed and validated carefully to minimize the risk of excluding relevant studies. Search strings were formed in a way that only a very small number of relevant studies could be missed, and a manageable quantity of irrelevant studies could be included. Besides the automatic search, a manual search and a forward snowballing were performed. We did not apply backward snowballing in addition to forward snowballing since some references achieved by the backward snowballing would be out of our search range, i.e. it would cause access and force to examine the papers published before 2010. Elimination of these old-dated papers would have an additional cost with probably very limited benefits. We already had a large pool of papers. The protocol was rigorously defined to be reusable by other researchers for reproducing the same study, i.e. the protocol is available on IEEE Data Port.[4]

[4] https://dx.doi.org/10.21227/zbkz-6461

VII. CONCLUSION

CPS have proven to offer tremendous opportunities in almost all areas of industry and society. Due to its inherent heterogeneity and complexity, engineering and managing such systems is known to be a challenge for the developers. Thus, numerous researches were conducted and are still being conducted in this domain.

The aim of this study was to identify the current features of the use of MDE for CPS. For this purpose, an SLR of the papers in the field, published between 2010 and 2018, was performed. The initial search retrieved 646 papers of which 140 were included in this study by following the defined selection strategy through a multi-stage process. A key feature of this SLR is that it is not restricted to a particular CPS domain. This broad scope in the search gives deeper insights into the state-of-the-art of using MDE for CPS. Findings contribute new knowledge that can be used to improve CPS development using MDE.

The study points out that MDE for CPS is an active research area with an increasing number of publications over the years. Moreover, the study shows the covered areas in addition to the languages and tools which have been used/proposed. Regarding the CPS components, our study also exposed that the MDE effort was mostly put on the development of cyber and physical components, where the other components (networks, sensors, and actuators) did not get much attention. Study results revealed that solutions based on UML and Eclipse-based tools were mostly preferred.

Finally, the study also presented some open issues in MDE of CPS for researchers and practitioners to assist shaping their future work in this area. For instance, designing and implementing sensors and actuators used in CPS with MDE and developing domain-specific simulation tools require further investigation.

ACKNOWLEDGMENT

Mustafa Abshir Mohamed would like to thank Turkish Government for Türkiye Scholarships (YTB) Program. Moharram Challenger and Geylani Kardas would also like to thank the European Cooperation in Science & Technology COST Action networking mechanisms and support of COST Action IC1404: Multi-Paradigm Modeling for Cyber-Physical Systems (MPM4CPS). COST was supported by the EU Framework Programme Horizon 2020.

REFERENCES

[1] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *Proc. Future Softw. Eng. (FOSE)*. Washington, DC, USA: IEEE Computer Society, May 2007, pp. 37–54.

[2] C. Durmaz, M. Challenger, O. Dagdeviren, and G. Kardas, "Modelling contiki-based IoT systems," in *Proc. 6th Symp. Lang., Appl. Technol. (SLATE)*. Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017, pp. 5:1–5:13.

[3] H. M. Marah, R. Eslampanah, and M. Challenger, "DSML4TinyOS: Code generation for wireless devices," in *Proc. Int. Workshop Model-Driven Eng. Internet Things (MDE IoT)*, 2018, pp. 509–514.

[4] T. Z. Asici, B. Karaduman, R. Eslampanah, M. Challenger, J. Denil, and H. Vangheluwe, "Applying model driven engineering techniques to the development of contiki-based IoT systems," in *Proc. IEEE/ACM 1st Int. Workshop Softw. Eng. Res. Practices Internet Things (SERP IoT)*, May 2019, pp. 25–32.

[5] Y. Jiang, H. Song, Y. Yang, H. Liu, M. Gu, Y. Guan, J. Sun, and A. L. Sha, "Dependable model-driven development of CPS: From stateflow simulation to verified implementation," *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 1, p. 12, 2019.

[6] A. Alshareef and H. S. Sarjoughian, "Model-driven time-accurate DEVS-based approaches for CPS design," in *Proc. Int. Symp. Model-Driven Approaches Simulation Eng., (Mod4Sim), Spring Simulation Multi-Conf., (SpringSim)*. Baltimore, MD, USA: Society for Computer Simulation International, Apr. 2018, pp. 84–95.

[7] H. Vangheluwe, "Multi-paradigm modelling of cyber-physical systems," in *Proc. 4th Int. Workshop Softw. Eng. Smart Cyber-Phys. Syst.*, May 2018, p. 1.

[8] G. Kardas, B. T. Tezel, and M. Challenger, "Domain-specific modelling language for belief–desire–intention software agents," *IET Softw.*, vol. 12, no. 4, pp. 356–364, Aug. 2018.

[9] S. Arslan and G. Kardas, "DSML4DT: A domain-specific modeling language for device tree software," *Comput. Ind.*, vol. 115, Feb. 2020, Art. no. 103179.

[10] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele Univ., Keele, U.K., Tech. Rep. EBSE-2007-01, 2007.

[11] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. 11th IEEE Int. Symp. Object Compon.-Oriented Real-Time Distrib. Comput. (ISORC)*, May 2008, pp. 363–369.

[12] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. Cambridge, MA, USA: MIT Press, 2016.

[13] T. Sanislav and G. Mois, "A dependability analysis model in the context of cyber-physical systems," in *Proc. 18th Int. Carpathian Control Conf. (ICCC)*, May 2017, pp. 146–150.

[14] S. Heath, *Embedded Systems Design*. Amsterdam, The Netherlands: Elsevier, 2002.

[15] F. van den Berg, V. Garousi, B. Tekinerdogan, and B. R. Haverkort, "Designing cyber-physical systems with aDSL: A domain-specific language and tool support," in *Proc. 13th Annu. Conf. Syst. Syst. Eng. (SoSE)*, Jun. 2018, pp. 225–232.

[16] E. A. Lee, "Cyber-physical systems-are computing foundations adequate," in *Proc. Position Paper NSF Workshop Cyber-Phys. Syst., Res. Motivat., Techn. Roadmap*, vol. 2, 2006, pp. 1–9.

[17] A. R. da Silva, "Model-driven engineering: A survey supported by the unified conceptual model," *Comput. Lang., Syst. Struct.*, vol. 43, pp. 139–155, Oct. 2015.

[18] E. A. Marand, E. A. Marand, and M. Challenger, "DSML4CP: A domain-specific modeling language for concurrent programming," *Comput. Lang., Syst. Struct.*, vol. 44, pp. 319–341, Dec. 2015.

[19] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*. Morgan & Claypool: San Rafael, CA, USA, 2017.

[20] A. Bucchiarone, J. Cabot, R. F. Paige, and A. Pierantonio, "Grand challenges in model-driven engineering: An analysis of the state of the research," *Softw. Syst. Model.*, vol. 19, no. 1, pp. 5–13, Jan. 2020.

[21] L. Strandén, J. Nilsson, and H. Lönn, "Model driven software verification and validation," SP Tech. Res. Inst. Sweden, Borås, Sweden, Tech. Rep. 2007:15, 2007.

[22] D. Cetinkaya, A. Verbraeck, and M. D. Seck, "MDD4MS: A model driven development framework for modeling and simulation," in *Proc. Summer Comput. Simulation Conf.*, 2011, pp. 113–121.

[23] B. Baudry, C. Nebut, and Y. L. Traon, "Model-driven engineering for requirements analysis," in *Proc. 11th IEEE Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2007, p. 459.

[24] A. Barišic, D. Savic, R. Al-Ali, I. Ruchkin, D. Blouin, A. Cicchetti, R. Eslampanah, O. Nikiforova, M. Abshir, and M. Challenger, "Systematic literature review on multi-paradigm modelling for cyber-physical systems," Eur. Supported EU Framework Programme Horizon, Tech. Rep. WG4 (delivery 4.4) MPM4CPS IC1404 COST Action, 2020, doi: 10.5281/zenodo.2528953.

[25] M. Rashid, M. W. Anwar, and A. M. Khan, "Toward the tools selection in model based system engineering for embedded systems—A systematic literature review," *J. Syst. Softw.*, vol. 106, pp. 150–163, Aug. 2015.

[26] P. G. G. Queiroz and R. T. V. Braga, "Development of critical embedded systems using model-driven and product lines techniques: A systematic review," in *Proc. 8th Brazilian Symp. Softw. Compon., Archit. Reuse*, Sep. 2014, pp. 74–83.

[27] G. L. Casalaro and G. Cattivera, "Model-driven engineering for mobile robot systems: A systematic mapping study," Malardalen Univ., Västerås, Sweden, Tech. Rep. urn:nbn:se:mdh:diva-28261, 2015.

[28] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, "Model-based engineering in the embedded systems domain: An industrial survey on the state-of-practice," *Softw. Syst. Model.*, vol. 17, no. 1, pp. 91–113, Feb. 2018.

[29] L. T. W. Agner, I. W. Soares, P. C. Stadzisz, and J. M. Simão, "A Brazilian survey on UML and model-driven practices for embedded software development," *J. Syst. Softw.*, vol. 86, no. 4, pp. 997–1005, Apr. 2013.

[30] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: A survey," *ACM Comput. Surveys*, vol. 51, no. 3, pp. 1–33, 2018.

[31] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial Internet of Things: A cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78238–78259, 2018.

[32] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2011, pp. 1–6.

[33] M. Rashid, M. W. Anwar, F. Azam, and M. Kashif, "Model-based requirements and properties specifications trends for early design verification of embedded systems," in *Proc. 11th Syst. Syst. Eng. Conf. (SoSE)*, Jun. 2016, pp. 1–7.

[34] J. Geismann and E. Bodden, "A systematic literature review of model-driven security engineering for cyber–physical systems," *J. Syst. Softw.*, vol. 169, Nov. 2020, Art. no. 110697.

[35] B. Liu, Y.-R. Zhang, X.-L. Cao, Y. Liu, B. Gu, and T.-X. Wang, "A survey of model-driven techniques and tools for cyber-physical systems," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 11, pp. 1567–1590, Nov. 2020.

[36] P. J. Mosterman and J. Zander, "Cyber-physical systems challenges: A needs analysis for collaborating embedded software systems," *Softw. Syst. Model.*, vol. 15, no. 1, pp. 5–16, Feb. 2016.

[37] M. A. Mohamed, M. Challenger, and G. Kardas, "Applications of model-driven engineering in cyber-physical systems: A systematic mapping study," *J. Comput. Lang.*, vol. 59, Aug. 2020, Art. no. 100972.

[38] J. C. de Almeida Biolchini, P. G. Mian, A. C. C. Natali, T. U. Conte, and G. H. Travassos, "Scientific research ontology to support systematic review in software engineering," *Adv. Eng. Informat.*, vol. 21, no. 2, pp. 133–151, Apr. 2007.

[39] A. Siddaway, "What is a systematic literature review and how do i do one," Univ. Stirling, Stirling, U.K., 2014, p. 1.

[40] M. Lavallee, P.-N. Robillard, and R. Mirsalari, "Performing systematic literature reviews with novices: An iterative approach," *IEEE Trans. Educ.*, vol. 57, no. 3, pp. 175–181, Aug. 2014.

[41] M. Skoglund and P. Runeson, "Reference-based search strategies in systematic reviews," in *Proc. EASE*, 2009, pp. 1–10.

[42] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proc. 18th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2014, p. 38.

[43] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman, "Systematic literature reviews in software engineering—A tertiary study," *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805, 2010.

[44] C. Wohlin, P. Runeson, P. A. da Mota Silveira Neto, E. Engström, I. do Carmo Machado, and E. S. de Almeida, "On the reliability of mapping studies in software engineering," *J. Syst. Softw.*, vol. 86, no. 10, pp. 2594–2610, 2013.

[45] W. Afzal, S. Alone, K. Glocksien, and R. Torkar, "Software test process improvement approaches: A systematic literature review and an industrial case study," *J. Syst. Softw.*, vol. 111, pp. 1–33, Jan. 2016.

[46] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 625–637, Jun. 2011.

[47] T. Kosar, S. Bohra, and M. Mernik, "A systematic mapping study driven by the margin of error," *J. Syst. Softw.*, vol. 144, pp. 439–449, Oct. 2018.

[48] F. S. Gonçalves, G. V. Raffo, and L. B. Becker, "Managing CPS complexity: Design method for unmanned aerial vehicles," *IFAC-PapersOnLine*, vol. 49, no. 32, pp. 141–146, 2016.

[49] J. Ding, Y. Atif, S. F. Andler, B. Lindström, and M. Jeusfeld, "CPS-based threat modeling for critical infrastructure protection," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 45, no. 2, pp. 129–132, Oct. 2017.

[50] L. Ollinger, M. A. Wehrmeister, C. E. Pereira, and D. Zühlke, "An integrated concept for the model-driven engineering of distributed automation architectures on embedded systems," *IFAC Proc. Volumes*, vol. 46, no. 7, pp. 222–227, May 2013.

[51] M. D'Angelo, A. Napolitano, and M. Caporuscio, "CyPhEF: A model-driven engineering framework for self-adaptive cyber-physical systems," in *Proc. 40th Int. Conf. Softw. Eng., Companion Proceeedings*, May 2018, pp. 101–104.

[52] M. W. Aziz and M. Rashid, "Domain specific modeling language for cyber physical systems," in *Proc. Int. Conf. Inf. Syst. Eng. (ICISE)*, Apr. 2016, pp. 29–33.

[53] M. U. Tariq, H. A. Nasir, A. Muhammad, and M. Wolf, "Model-driven performance analysis of large scale irrigation networks," in *Proc. IEEE/ACM 3rd Int. Conf. Cyber-Phys. Syst.*, Apr. 2012, pp. 151–160.

[54] T. Nagele and J. Hooman, "Rapid construction of co-simulations of cyber-physical systems in HLA using a DSL," in *Proc. 43rd Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2017, pp. 247–251.

[55] E. Lavigne, G. Guillou, and J.-P. Babau, "AVS, a model-based racing sailboat simulator: Application to wind integration," *IFAC-PapersOnLine*, vol. 51, no. 10, pp. 88–94, 2018.

[56] S. M. Pradhan, A. Dubey, A. Gokhale, and M. Lehofer, "CHARIOT: A domain specific language for extensible cyber-physical systems," in *Proc. Workshop Domain-Specific Model.*, Oct. 2015, pp. 9–16.

[57] M. A. A. Mamun, C. Berger, and J. Hansson, "MDE-based sensor management and verification for a self-driving miniature vehicle," in *Proc. ACM Workshop Domain-Specific Modeling (DSM)*, 2013, pp. 1–6.

[58] X. Koutsoukos, N. Kottenstette, J. Hall, E. Eyisi, H. Leblanc, J. Porter, and J. Sztipanovits, "A passivity approach for model-based compositional design of networked control systems," *ACM Trans. Embedded Comput. Syst.*, vol. 11, no. 4, p. 75, 2012.

[59] M. U. Tariq, J. Florence, and M. Wolf, "Design specification of cyber-physical systems: Towards a domain-specific modeling language based on simulink, eclipse modeling framework, and giotto," in *Proc. ACESMB MoDELS*, Sep. 2014, pp. 6–15.

[60] F. Alrimawi, L. Pasquale, D. Mehta, and B. Nuseibeh, "I've seen this before: Sharing cyber-physical incident knowledge," in *Proc. 1st Int. Workshop Secur. Awareness Design Deployment*, May 2018, pp. 33–40.

[61] G. Martins, S. Bhatia, X. Koutsoukos, K. Stouffer, C. Tang, and R. Candell, "Towards a systematic threat modeling approach for cyber-physical systems," in *Proc. Resilience Week (RWS)*, Aug. 2015, pp. 1–6.

[62] S. Maksuti, A. Bicaku, M. Tauber, S. Palkovits-Rauter, S. Haas, and J. Delsing, "Towards flexible and secure end-to-end communication in industry 4.0," in *Proc. IEEE 15th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2017, pp. 883–888.

[63] F. Cicirelli, G. Fortino, A. Guerrieri, G. Spezzano, and A. Vinci, "A meta-model framework for the design and analysis of smart cyber-physical environments," in *Proc. IEEE 20th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2016, pp. 687–692.

[64] S. Bougouffa, K. Meßzmer, S. Cha, E. Trunzer, and B. Vogel-Heuser, "Industry 4.0 interface for dynamic reconfiguration of an open lab size automated production system to allow remote community experiments," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Dec. 2017, pp. 2058–2062.

[65] K. Thramboulidis, I. Kontou, and D. C. Vachtsevanou, "Towards an IoT-based framework for evolvable assembly systems," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 182–187, 2018.

[66] F. Cicirelli, G. Fortino, A. Guerrieri, G. Spezzano, and A. Vinci, "Metamodeling of smart environments: From design to implementation," *Adv. Eng. Informat.*, vol. 33, pp. 274–284, Aug. 2017.

[67] P. Huang, K. Jiang, C. Guan, and D. Du, "Towards modeling cyber-physical systems with SysML/MARTE/pCCSL," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 1, Jul. 2018, pp. 264–269.

[68] E. Irisarri, M. V. García, F. Pérez, E. Estevez, and M. Marcos, "A model-based approach for process monitoring in oil production industry," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–4.

[69] M. Walch, "Knowledge-driven enrichment of cyber-physical systems for industrial applications using the KbR modelling approach," in *Proc. IEEE Int. Conf. Agents (ICA)*, Jul. 2017, pp. 84–89.

[70] C. Buckl, D. Sojer, and A. Knoll, ''FTOS: Model-driven development of fault-tolerant automation systems,'' in *Proc. IEEE 15th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2010, pp. 1–8.

[71] I. Graja, S. Kallel, N. Guermouche, and A. H. Kacem, ''BPMN4CPS: A BPMN extension for modeling cyber-physical systems,'' in *Proc. IEEE 25th Int. Conf. Enabling Technol., Infrastruct. Collaborative Enterprises (WETICE)*, Jun. 2016, pp. 152–157.

[72] C. Heinzemann, S. Becker, and A. Volk, ''Transactional execution of hierarchical reconfigurations in cyber-physical systems,'' *Softw. Syst. Model.*, vol. 18, no. 1, pp. 157–189, Feb. 2019.

[73] K. Zhang and J. Sprinkle, ''Model-based software synthesis for self-reconfigurable sensor network in water monitoring,'' in *Proc. 20th IEEE Int. Conf. Workshops Eng. Comput. Based Syst. (ECBS)*, Apr. 2013, pp. 40–48.

[74] S. Bao, J. Porter, and A. Gokhale, ''Reasoning for CPS education using surrogate simulation models,'' in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 1, Jun. 2016, pp. 764–773.

[75] K. Zhang and J. Sprinkle, ''A closed-loop model-based design approach based on automatic verification and transformation,'' in *Proc. 14th Workshop Domain-Specific Modeling (DSM)*, 2014, pp. 1–6.

[76] K. An, A. Trewyn, A. Gokhale, and S. Sastry, ''Model-driven performance analysis of reconfigurable conveyor systems used in material handling applications,'' in *Proc. IEEE/ACM 2nd Int. Conf. Cyber-Phys. Syst.*, Apr. 2011, pp. 141–150.

[77] M. Bunting, Y. Zeleke, K. McKeever, and J. Sprinkle, ''A safe autonomous vehicle trajectory domain specific modeling language for non-expert development,'' in *Proc. Int. Workshop Domain-Specific Modeling*, Oct. 2016, pp. 42–48.

[78] C. Guan, Y. Ao, D. Du, and F. Mallet, ''XSHS: An executable domain-specific modeling language for modeling stochastic and hybrid behaviors of cyber-physical systems,'' in *Proc. 25th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2018, pp. 683–687.

[79] R. Seiger, C. Keller, F. Niebling, and T. Schlegel, ''Modelling complex and flexible processes for smart cyber-physical environments,'' *J. Comput. Sci.*, vol. 10, pp. 137–148, Sep. 2015.

[80] F. Cicirelli, G. Fortino, A. Guerrieri, A. Mercuri, G. Spezzano, and A. Vinci, ''A metamodel framework for edge-based smart environments,'' in *Proc. IEEE Int. Conf. Cloud Eng. (IC E)*, Apr. 2018, pp. 286–291.

[81] T. Hartmann, A. Moawad, F. Fouquet, G. Nain, J. Klein, and Y. L. Traon, ''Stream my models: Reactive peer-to-peer distributed modelsrun.Time,'' in *Proc. ACM/IEEE 18th Int. Conf. Model Driven Eng. Lang. Syst. (MODELS)*, Sep. 2015, pp. 80–89.

[82] G. Barbieri and C. Fantuzzi, ''Design of cyber-physical systems: Definition and metamodel for reusable resources,'' in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–9.

[83] V. R. S. Kumar, M. Shanmugavel, V. Ganapathy, and B. Shirinzadeh, ''Unified meta-modeling framework using bond graph grammars for conceptual modeling,'' *Robot. Auto. Syst.*, vol. 72, pp. 114–130, Oct. 2015.

[84] D. Chen, A. Maffei, J. Ferreirar, H. Akillioglu, M. R. Khabazzi, and X. Zhang, ''A virtual environment for the management and development of cyber-physical manufacturing systems,'' *IFAC-PapersOnLine*, vol. 48, no. 7, pp. 29–36, 2015.

[85] J. Jeon, I. Chun, and W. Kim, ''Metamodel-based CPS modeling tool,'' in *Embedded Multimedia Comput. Technol. Service*. Dordrecht, The Netherlands: Springer, 2012, pp. 285–291.

[86] P. Bocciarelli, A. D'Ambrogio, A. Giglio, and E. Paglia, ''A BPMN extension for modeling cyber-physical-production-systems in the context of industry 4.0,'' in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 599–604.

[87] A. Tundis, R. Egert, and M. Mühlhäuser, ''Applying a properties modeling approach for monitoring smart grids,'' in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 714–719.

[88] S. Merschak, P. Hehenberger, M. Witters, and K. Gadeyne, ''A hierarchical meta-model for the design of cyber-physical production systems,'' in *Proc. 19th Int. Conf. Res. Educ. Mechatronics (REM)*, Jun. 2018, pp. 36–41.

[89] M. Ciavotta, A. Bettoni, and G. Izzo, ''Interoperable meta model for simulation-in-the-loop,'' in *Proc. IEEE Ind. Cyber-Physical Syst. (ICPS)*, May 2018, pp. 702–707.

[90] Y. Barve, H. Neema, S. Rees, and J. Sztipanovits, ''Towards a design studio for collaborative modeling and co-simulations of mixed electrical energy systems,'' in *Proc. IEEE Int. Sci. Smart City Oper. Platforms Eng. Partnership Global City Teams Challenge (SCOPE-GCTC)*, Apr. 2018, pp. 24–29.

[91] A. Hahn, R. K. Thomas, I. Lozano, and A. Cardenas, ''A multi-layered and kill-chain based security analysis framework for cyber-physical systems,'' *Int. J. Crit. Infrastruct. Protection*, vol. 11, pp. 39–50, Dec. 2015.

[92] M. Housh and Z. Ohar, ''Model-based approach for cyber-physical attack detection in water distribution systems,'' *Water Res.*, vol. 139, pp. 132–143, Aug. 2018.

[93] R. Matinnejad, S. Nejati, L. C. Briand, and T. Bruckmann, ''Test generation and test prioritization for simulink models with dynamic behavior,'' *IEEE Trans. Softw. Eng.*, vol. 45, no. 9, pp. 919–944, Sep. 2019.

[94] Y. Mordecai, ''Conceptual modeling of cyber-physical gaps in air traffic control,'' *Procedia Comput. Sci.*, vol. 140, pp. 21–28, Jan. 2018.

[95] B. V. Acker, J. Denil, H. Vangheluwe, and P. D. Meulenaere, ''Managing heterogeneity in model-based systems engineering of cyber-physical systems,'' in *Proc. 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (PGCIC)*, Nov. 2015, pp. 617–622.

[96] H. Neema, B. Potteiger, X. Koutsoukos, G. Karsai, P. Volgyesi, and J. Sztipanovits, ''Integrated simulation testbed for security and resilience of CPS,'' in *Proc. 33rd Annu. ACM Symp. Appl. Comput.*, Apr. 2018, pp. 368–374.

[97] N. Rashid, J. Wan, G. Quiros, A. Canedo, and M. A. Al Faruque, ''Modeling and simulation of cyberattacks for resilient cyber-physical systems,'' in *Proc. 13th IEEE Conf. Autom. Sci. Eng. (CASE)*, Aug. 2017, pp. 988–993.

[98] M. Di Natale, M. Morelli, and F. Cremona, ''Matching execution architecture models with functional models to analyze the time performance of CPS systems,'' in *Proc. Int. Conf. Complex Syst. Eng. (ICCSE)*, Nov. 2015, pp. 1–6.

[99] C.-F. Fan, C.-C. Chan, H.-Y. Yu, and S. Yih, ''A simulation platform for human-machine interaction safety analysis of cyber-physical systems,'' *Int. J. Ind. Ergonom.*, vol. 68, pp. 89–100, Nov. 2018.

[100] W. Yan, Y. Xue, X. Li, J. Weng, T. Busch, and J. Sztipanovits, ''Integrated simulation and emulation platform for cyber-physical system security experimentation,'' in *Proc. 1st Int. Conf. High Confidence Netw. Syst. (HiCoNS)*, 2012, pp. 81–88.

[101] M. Pajic, Z. Jiang, I. Lee, O. Sokolsky, and R. Mangharam, ''From verification to implementation: A model translation tool and a pacemaker case study,'' in *Proc. IEEE 18th Real Time Embedded Technol. Appl. Symp.*, Apr. 2012, pp. 173–184.

[102] J. M. van de Mortel-Fronczak, R. G. M. Huisman, M. H. R. van der Heijden, and M. A. Reniers, ''Supervisor synthesis in model-based automotive systems engineering,'' in *Proc. ACM/IEEE Int. Conf. Cyber-Physical Syst. (ICCPS)*. Washington, DC, USA: IEEE Computer Society, Apr. 2014, pp. 187–198.

[103] E. Palachi, C. Cohen, and S. Takashi, ''Simulation of cyber physical models using SysML and numerical solvers,'' in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2013, pp. 671–675.

[104] G. Simko, D. Lindecker, T. Levendovszky, E. Jackson, S. Neema, and J. Sztipanovits, ''A framework for unambiguous and extensible specification of DSMLs for cyber-physical systems,'' in *Proc. 20th IEEE Int. Conf. Workshops Eng. Comput. Based Syst. (ECBS)*, Apr. 2013, pp. 30–39.

[105] L. Zhang, ''A framework to specify big data driven complex cyber physical control systems,'' in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Jul. 2014, pp. 548–553.

[106] R. Parveen, S. Pradhan, and N. Goveas, ''Design and feasibility study of health related devices using COTS components,'' in *Proc. IEEE Ind. Cyber-Physical Syst. (ICPS)*, May 2018, pp. 529–533.

[107] L. C. Silva, M. Perkusich, F. M. Bublitz, H. O. Almeida, and A. Perkusich, ''A model-based architecture for testing medical cyber-physical systems,'' in *Proc. 29th Annu. ACM Symp. Appl. Comput.*, Mar. 2014, pp. 25–30.

[108] S. Neema, G. Simko, T. Levendovszky, J. Porter, A. Agrawal, and J. Sztipanovits, ''Formalization of software models for cyber-physical systems,'' in *Proc. 2nd FME Workshop Formal Methods Softw. Eng. (FormaliSE)*, 2014, pp. 45–51.

[109] F. Pereira and L. Gomes, "The IOPT-flow framework pairing Petri nets and data-flows for embedded controller development," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 4832–4837.

[110] R. Weissnegger, M. Schuss, C. Kreiner, M. Pistauer, K. Römer, and C. Steger, "Simulation-based verification of automotive safety-critical systems based on EAST-ADL," *Procedia Comput. Sci.*, vol. 83, pp. 245–252, Jan. 2016.

[111] M. Lora, E. Fraccaroli, and F. Fummi, "Virtual prototyping of smart systems through automatic abstraction and mixed-signal scheduling," in *Proc. 22nd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 232–237.

[112] N. Navet and L. Fejoz, "CPAL: High-level abstractions for safe embedded systems," in *Proc. Int. Workshop Domain-Specific Modeling*, Oct. 2016, pp. 35–41.

[113] F. F. H. Reijnen, M. A. Goorden, J. M. van de Mortel-Fronczak, and J. E. Rooda, "Supervisory control synthesis for a waterway lock," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2017, pp. 1562–1563.

[114] C. Gerking, W. Schafer, S. Dziwok, and C. Heinzemann, "Domain-specific model checking for cyber-physical systems," in *Proc. MoDeVVa MoDELS*, 2015, pp. 18–27.

[115] C. Guo, Z. Fu, Z. Zhang, S. Ren, and L. Sha, "Model and integrate medical resource available times and relationships in verifiably correct executable medical best practice guideline models," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Physical Syst. (ICCPS)*, Apr. 2018, pp. 253–262.

[116] F. S. Gonçalves and L. B. Becker, "Model driven engineering approach to design sensing and actuation subsystems," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–8.

[117] L. Zhang, "View oriented approach to specify and model aerospace cyber-physical systems," in *Proc. IEEE 11th Int. Conf. Dependable, Autonomic Secure Comput.*, Dec. 2013, pp. 296–303.

[118] L. Zhang and S. Feng, "Integration design and model transformation for cyber physical systems," in *Proc. IEEE 5th Int. Conf. Softw. Eng. Service Sci.*, Jun. 2014, pp. 754–757.

[119] A. Drago, S. Marrone, N. Mazzocca, A. Tedesco, and V. Vittorini, "Model-driven estimation of distributed vulnerability in complex railway networks," in *Proc. IEEE 10th Int. Conf. Ubiquitous Intell. Comput., IEEE 10th Int. Conf. Autonomic Trusted Comput.*, Dec. 2013, pp. 380–387.

[120] G. Bakirtzis, B. T. Carter, C. R. Elks, and C. H. Fleming, "A model-based approach to security analysis for cyber-physical systems," in *Proc. Annu. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2018, pp. 1–8.

[121] A. Motii, B. Hamid, A. Lanusse, and J.-M. Bruel, "Guiding the selection of security patterns for real-time systems," in *Proc. 21st Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Nov. 2016, pp. 155–164.

[122] J. Dell, T. Greiner, and W. Rosenstiel, "Model-based platform design and evaluation of cloud-based cyber-physical systems (CCPS)," in *Proc. 12th IEEE Int. Conf. Ind. Informat. (INDIN)*, Jul. 2014, pp. 376–381.

[123] S. Whitsitt, J. Sprinkle, and R. Lysecky, "Generating model transformations for mending dynamic constraint violations in cyber physical systems," in *Proc. 14th Workshop Domain-Specific Modeling (DSM)*, 2014, pp. 35–40.

[124] H. S. Son, W. Y. Kim, R. Y. Kim, and H.-G. Min, "Metamodel design for model transformation from simulink to ECML in cyber physical systems," in *Computer Applications for Graphics, Grid Computing, and Industrial Environment*. Berlin, Germany: Springer, 2012, pp. 56–60.

[125] F. S. Gonçalves, D. Pereira, E. Tovar, and L. B. Becker, "Formal verification of AADL models using UPPAAL," in *Proc. 7th Brazilian Symp. Comput. Syst. Eng. (SBESC)*, Nov. 2017, pp. 117–124.

[126] R. F. Passarini, L. B. Becker, and J.-M. Farines, "The assisted transformation of models: Supporting cyber-physical systems design by extracting architectural aspects and operating modes from simulink functional models," in *Proc. 3rd Brazilian Symp. Comput. Syst. Eng.*, Nov. 2013, pp. 47–52.

[127] B. Cheng, X. Wang, J. Liu, and D. Du, "Modana: An integrated framework for modeling and analysis of energy-aware CPSs," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf.*, vol. 2, Jul. 2015, pp. 127–136.

[128] I. S. Brito, J. A. P. Barros, and L. Gomes, "From requirements to code (Re2Code)—A model-based approach for controller implementation," in *Proc. IEEE 14th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2016, pp. 1224–1230.

[129] Y. Zhou, X. Gong, J. Li, and B. Li, "Verifying CPS for self-adaptability," in *Proc. IEEE/ACIS 17th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2018, pp. 166–172.

[130] D. Chen, D. V. Panfilenko, M. R. Khabbazi, and D. Sonntag, "A model-based approach to qualified process automation for anomaly detection and treatment," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–8.

[131] O. Nikiforova, N. El Marzouki, K. Gusarovs, H. Vangheluwe, T. Bures, R. Al-Ali, M. Iacono, P. O. Esquivel, and F. Leon, "The two-hemisphere modelling approach to the composition of cyber-physical systems," in *Proc. 12th Int. Conf. Softw. Technol.* Madrid, Spain: SciTePress, 2017, pp. 286–293.

[132] X. Liu, J. Zhang, and P. Zhu, "Modeling cyber-physical attacks based on probabilistic colored Petri nets and mixed-strategy game theory," *Int. J. Crit. Infrastruct. Protection*, vol. 16, pp. 13–25, Mar. 2017.

[133] N. Jarus, S. S. Sarvestani, and A. R. Hurson, "Models, metamodels, and model transformation for cyber-physical systems," in *Proc. 7th Int. Green Sustain. Comput. Conf. (IGSC)*, 2016, pp. 1–8.

[134] R. Chen, Y. Liu, and X. Ye, "Ontology based behavior verification for complex systems," in *Proc. ASME Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf.* New York, NY, USA: ASME, 2018, pp. 1–9.

[135] M. Tuo, X. Zhou, G. Yang, and N. Fu, "An approach for safety analysis of cyber-physical system based on model transformation," in *Proc. IEEE Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmartData)*, Dec. 2016, pp. 636–639.

[136] I. Graja, S. Kallel, N. Guermouche, and A. H. Kacem, "Verification of the consistency of time-aware cyber-physical processes," in *Proc. Int. Conf. Service-Oriented Comput.* Cham, Switzerland: Springer, 2017, pp. 67–79.

[137] K. Lynch, R. Ramsey, G. Ball, M. Schmit, and K. Collins, "Ontology-driven metamodel validation in cyber-physical systems," in *Information Technology: New Generations*. Cham, Switzerland: Springer, 2016, pp. 1255–1258.

[138] X. He, "Modeling and analyzing cyber physical systems using high level Petri nets," in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2018, pp. 469–476.

[139] L. Pagliari, R. Mirandola, and C. Trubiani, "Multi-modeling approach to performance engineering of cyber-physical systems design," in *Proc. 22nd Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Nov. 2017, pp. 142–145.

[140] U. Pohlmann, H. Trsek, L. Dürkop, S. Dziwok, and F. Oestersötebier, "Application of an intelligent network architecture on a cooperative cyber-physical system: An experience report," in *Proc. IEEE Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2014, pp. 1–6.

[141] L. Silva, H. Almeida, A. Perkusich, and M. Perkusich, "A model-based approach to support validation of medical cyber-physical systems," *Sensors*, vol. 15, no. 11, pp. 27625–27670, Oct. 2015.

[142] A. Murugesan, M. P. E. Heimdahl, M. W. Whalen, S. Rayadurgam, J. Komp, L. Duan, B.-G. Kim, O. Sokolsky, and I. Lee, "From requirements to code: Model based development of a medical cyber physical system," in *Software Engineering in Health Care*. Cham, Switzerland: Springer, 2014, pp. 96–112.

[143] R. Cohen, R. Jobredeaux, and E. Feron, "A credible autocoding application within a rocket and its payload," in *Proc. IEEE/AIAA 34th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2015. p. 8C4–1.

[144] A. Banerjee and S. K. S. Gupta, "Model based code generation for medical cyber physical systems," in *Proc. 1st Workshop Mobile Med. Appl. (MMA)*, 2014, pp. 22–27.

[145] C. Guo, Z. Fu, S. Ren, Y. Jiang, M. Rahmaniheris, and L. Sha, "Pattern-based statechart modeling approach for medical best practice guidelines—A case study," in *Proc. IEEE 30th Int. Symp. Comput.-Based Med. Syst. (CBMS)*, Jun. 2017, pp. 117–122.

[146] M. W. Whalen, A. Murugesan, S. Rayadurgam, and M. P. E. Heimdahl, "Structuring simulink models for verification and reuse," in *Proc. 6th Int. Workshop Modeling Softw. Eng. (MiSE)*, 2014, pp. 19–24.

[147] J. Peters, R. Wille, N. Przigoda, U. Kühne, and R. Drechsler, "A generic representation of CCSL time constraints for UML/MARTE models," in *Proc. 52nd Annu. Design Autom. Conf.*, Jun. 2015, p. 122.

[148] S. Kothari, P. Awadhutkar, A. Tamrawi, and J. Mathews, "Modeling lessons from verifying large software systems for safety and security," in *Proc. Winter Simulation Conf. (WSC)*. Las Vegas, NV, USA: IEEE Press, Dec. 2017, p. 109.

[149] H. Orojloo and M. A. Azgomi, "A game-theoretic approach to model and quantify the security of cyber-physical systems," *Comput. Ind.*, vol. 88, pp. 44–57, Jun. 2017.

[150] B. Cheng and D. Du, "Towards a stochastic occurrence-based modeling approach for stochastic CPSs," in *Proc. Theor. Aspects Softw. Eng. Conf.*, Sep. 2014, pp. 162–169.

[151] S. Xin, Q. Guo, H. Sun, B. Zhang, J. Wang, and C. Chen, "Cyber-physical modeling and cyber-contingency assessment of hierarchical control systems," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2375–2385, Sep. 2015.

[152] Z. Qian and H. Yu, "A TAOPN approach to modeling and scheduling cyber-physical systems," in *Proc. Int. Conf. Inf. Sci. Appl. (ICISA)*, Jun. 2013, pp. 1–7.

[153] T. M. Chen, J. C. Sanchez-Aarnoutse, and J. Buford, "Petri net modeling of cyber-physical attacks on smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 741–749, Dec. 2011.

[154] L. Zhang, "An integration approach to specify and model automotive cyber physical systems," in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, Dec. 2013, pp. 568–573.

[155] P. Garamvölgyi, I. Kocsis, B. Gehl, and A. Klenik, "Towards model-driven engineering of smart contracts for cyber-physical systems," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2018, pp. 134–139.

[156] M. A. H. Kacem, Z. Simeu-Abazi, E. Gascard, G. Lemasson, and A. Maisonnasse, "Application of a modeling approach on a cyber-physical system, 'RobAIR,'" *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14230–14235, 2017.

[157] B. Brandenbourger, M. Vathoopan, and A. Zoitl, "Behavior modeling of automation components using cross-domain interdependencies," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–4.

[158] S. Li, D. Li, F. Li, and N. Zhou, "CPSiCGF: A code generation framework for CPS integration modeling," *Microprocessors Microsyst.*, vol. 39, no. 8, pp. 1234–1244, Nov. 2015.

[159] M. Rocchetto and N. O. Tippenhauer, "Towards formal security analysis of industrial control systems," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 114–126.

[160] L. Petnga and M. Austin, "An ontological framework for knowledge modeling and decision support in cyber-physical systems," *Adv. Eng. Informat.*, vol. 30, no. 1, pp. 77–94, Jan. 2016.

[161] C. Cheh, K. Keefe, B. Feddersen, B. Chen, W. G. Temple, and W. H. Sanders, "Developing models for physical attacks in cyber-physical systems," in *Proc. Workshop Cyber-Phys. Syst. Secur. PrivaCy*, Nov. 2017, pp. 49–55.

[162] D. Gritzner and J. Greenyer, "Generating correct, compact, and efficient PLC code from scenario-based assume-guarantee specifications," *Procedia Manuf.*, vol. 24, pp. 153–158, Jan. 2018.

[163] A. Ataíde, J. P. Barros, I. S. Brito, and L. Gomes, "Towards automatic code generation for distributed cyber-physical systems: A first prototype for Arduino boards," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–4.

[164] T. M. Low and F. Franchetti, "High assurance code generation for cyber-physical systems," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng. (HASE)*, 2017, pp. 104–111.

[165] H. Zhao, L. Apvrille, and F. Mallet, "Multi-view design for cyber-physical systems," in *Proc. Symp. 13th Int. Conf. ICT Educ., Res., Ind. Appl.*, 2017, pp. 1–8.

[166] N. Ali and J.-E. Hong, "Failure detection and prevention for cyber-physical systems using ontology-based knowledge base," *Computers*, vol. 7, no. 4, p. 68, Dec. 2018.

[167] G. Sapienza, I. Crnkovic, and P. Potena, "Architectural decisions for HW/SW partitioning based on multiple extra-functional properties," in *Proc. IEEE/IFIP Conf. Softw. Archit.*, Apr. 2014, pp. 175–184.

[168] M. Lezoche and H. Panetto, "Cyber-physical systems, a new formal paradigm to model redundancy and resiliency," *Enterprise Inf. Syst.*, vol. 14, pp. 1–22, Sep. 2018.

[169] X. Hu, S. Liu, G. Chen, and C. Jiang, "Dependability modelling and evaluation of cyber-physical systems: A model-driven perspective," in *Proc. 1st Int. Workshop Cloud Comput. Inf. Secur.* Shanghai, China: Atlantis Press, 2013, pp. 40–44.

[170] D. Chen and Z. Lu, "A model-based approach to dynamic self-assessment for automated performance and safety awareness of cyber-physical systems," in *Proc. Int. Symp. Model-Based Saf. Assessment*. Cham, Switzerland: Springer, 2017, pp. 227–240.

[171] D. Morozov, M. Lezoche, and H. Panetto, "Multi-paradigm modelling of cyber-physical systems," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1385–1390, 2018.

[172] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A survey on concepts, applications, and challenges in cyber-physical systems," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 12, pp. 4242–4268, 2014.

[173] L. Zhang, "Aspect-oriented MDA approach for non-functional properties of distributed cyber physical systems," in *Proc. 10th Int. Symp. Distrib. Comput. Appl. Bus., Eng. Sci.*, Oct. 2011, pp. 284–288.

[174] K. Thramboulidis, D. C. Vachtsevanou, and A. Solanos, "Cyber-physical microservices: An IoT-based framework for manufacturing systems," in *Proc. IEEE Ind. Cyber-Phys. Syst. (ICPS)*, May 2018, pp. 232–239.

[175] V. Koutsoumpas, "A model-based approach for the specification of a virtual power plant operating in open context," in *Proc. IEEE/ACM 1st Int. Workshop Softw. Eng. Smart Cyber-Phys. Syst.* IEEE Press, May 2015, pp. 26–32.

[176] J. F. Broenink, P.-J.-D. Vos, Z. Lu, and M. M. Bezemer, "A co-design approach for embedded control software of cyber-physical systems," in *Proc. 11th Syst. Syst. Eng. Conf. (SoSE)*, Jun. 2016, pp. 1–5.

[177] S. Nannapaneni, S. Mahadevan, S. Pradhan, and A. Dubey, "Towards reliability-based decision making in cyber-physical systems," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2016, pp. 1–6.

[178] J. Plasse, J. Noble, and K. Myers, "An adaptive modeling framework for bivariate data streams with applications to change detection in cyber-physical systems," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 1074–1081.

[179] F. Latombe, X. Crégut, B. Combemale, J. Deantoni, and M. Pantel, "Weaving concurrency in executable domain-specific modeling languages," in *Proc. ACM SIGPLAN Int. Conf. Softw. Lang. Eng.*, Pittsburgh, PA, USA, Oct. 2015, pp. 125–136.

[180] V. Mezhuyev and R. Samet, "Geometrical meta-metamodel for cyber-physical modelling," in *Proc. Int. Conf. Cyberworlds*, Oct. 2013, pp. 89–93.

[181] F. Tan, Y. Wang, Q. Wang, L. Bu, R. Zheng, and N. Suri, "Guaranteeing proper-temporal-embedding safety rules in wireless CPS: A hybrid formal modeling approach," in *Proc. 43rd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2013, pp. 1–12.

[182] R. Sinha, C. Pang, G. S. Martínez, J. Kuronen, and V. Vyatkin, "Requirements-aided automatic test case generation for industrial cyber-physical systems," in *Proc. 20th Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Dec. 2015, pp. 198–201.

[183] J. Liu and L. Zhang, "Aspect-oriented MDA development method for non-functional properties of cyber physical systems," in *Proc. 2nd Int. Conf. Netw. Distrib. Comput.*, Sep. 2011, pp. 149–153.

[184] S. Feldmann, S. Rösch, D. Schütz, and B. Vogel-Heuser, "Model-driven engineering and semantic technologies for the design of cyber-physical systems," *IFAC Proc. Volumes*, vol. 46, no. 7, pp. 210–215, May 2013.

[185] M. Blackburn and P. Denno, "Virtual design and verification of cyber-physical systems: Industrial process plant design," *Procedia Comput. Sci.*, vol. 28, pp. 883–890, Jan. 2014.

[186] A. D. Brucker and J. Doser, "Metamodel-based UML notations for domain-specific languages," in *Proc. 4th Int. Workshop Softw. Lang. Eng. (ATEM)*, 2007.

[187] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer, 2012, p. 236.

MUSTAFA ABSHIR MOHAMED received the B.Sc. degree in information communication technology (ICT) from Gollis University, in 2015. Then, he was awarded a Scholarship from the Turkish Government under the program (YTB) for his master degree. In 2020, he completed his graduate studies at the International Computer Institute, Ege University, and received the M.Sc. degree in information technologies.

GEYLANI KARDAS received the B.Sc. degree in computer engineering and the M.Sc. and Ph.D. degrees in information technologies from Ege University, Turkey, in 2001, 2003, and 2008, respectively. He is currently an Associate Professor with the International Computer Institute (ICI), Ege University, and the Head of the Software Engineering Research Laboratory (Ege-SERLab), ICI. His research interests include agent-oriented software engineering, model-driven software development, and domain-specific (modeling) languages. He has authored or coauthored over 100 peer-reviewed articles in these research areas. He works as the principle investigator, a researcher or a consultant in various research and development projects funded by governments, agencies, and private corporations. He is also an Associate Editor of the *Journal of Computer Languages* (Elsevier).

MOHARRAM CHALLENGER (Member, IEEE) received the Ph.D. degree in IT from the International Computer Institute, Ege University, in February 2016. From 2005 to 2009, he had been a Faculty Member and a Senior Lecturer with the Computer Engineering Department, IAU-Shabestar University. From 2010 to 2013, he was a Researcher and a Team Leader of a bilateral project between Slovenia and Turkey (TUBITAK). From 2012 to 2016, he was the Research and Development Director of UNIT IT Ltd. leading a national project funded by TUBITAK and two international software-intensive projects in Europe called ITEA ModelWriter and ITEA Assume. From 2017 to 2018, he had also been a Faculty Member and an Assistant Professor with Ege University. From January 2019 to July 2020, he was a Postdoctoral Researcher with the University of Antwerp working in a Flanders Make projects called PACo and DTDesign. He is currently a Tenure-Track Assistant Professor with the Department of Computer Science, University of Antwerp. His research interests include domain-specific modeling languages, multi-agent systems, cyber-physical systems, and the Internet of Things.