Multi-Paradigm Modeling for Cyber-Physical Systems: A Systematic Mapping Review

Ankica Barisic, Ivan Ruchkin, Dušan Savić, Mustafa Abshir Mohamed, Rima Al-Ali, Letitia W Li, Hana Mkaouar, Raheleh Eslampanah, Moharram Challenger, Dominique Blouin, et al.

Multi-Paradigm Modeling for Cyber-Physical Systems: A Systematic Mapping Review

Ankica Barišić[a], Ivan Ruchkin[c], Dušan Savić[d], Mustafa Abshir Mohamed[e], Rima Al-Ali[f], Letitia W. Li[g], Hana Mkaouar[h], Raheleh Eslampanah[i], Moharram Challenger[i], Dominique Blouin[h], Oksana Nikiforova[j], Antonio Cicchetti[k]

[a] *Universite Cote d'Azur, CNRS, Inria, Sophia Antipolis, France*
[b] *NOVA LINCS, University NOVA of Lisboa, Portugal*
[c] *University of Pennsylvania, US*
[d] *Faculty of Organizational Sciences, Belgrade, Serbia*
[e] *Ege University, Izmir, Turkey*
[f] *Charles University, Prague, Czech Republic*
[g] *FAST Labs, BAE Systems, US*
[h] *LTCI, Telecom Paris, Institut Polytechnique de Paris, France*
[i] *University of Antwerp and Flanders Make, Belgium*
[j] *Riga Technical University, Riga, Latvia*
[k] *Mälardalen University, Västerås, Sweden*

Abstract

Cyber-Physical Systems (CPS) are heterogeneous and require cross-domain expertise to model. The complexity of these systems leads to questions about prevalent modeling approaches, their ability to integrate heterogeneous models, and their relevance to the application domains and stakeholders. The methodology for Multi-Paradigm Modeling (MPM) of CPS is not yet fully established and standardized, and researchers apply existing methods for modeling of complex systems and introducing their own. No systematic review has been previously performed to create an overview of the field on

*Corresponding Author

Email addresses: a.barisic@campus.fct.unl.pt (Ankica Barišić), iruchkin@cis.upenn.edu (Ivan Ruchkin), dusan.savic@fon.bg.ac.rs (Dušan Savić), mustafaxoodiye@gmail.com (Mustafa Abshir Mohamed), alali@d3s.mff.cuni.cz (Rima Al-Ali), letitia.li@baesystems.com (Letitia W. Li), hana.mkaouar@telecom-paris.fr (Hana Mkaouar), raheleh.eslampanah@uantwerpen.be (Raheleh Eslampanah), moharram.challenger@unatwerpen.be (Moharram Challenger), dominique.blouin@telecom-paris.fr (Dominique Blouin), oksana.nikiforova@rtu.lv (Oksana Nikiforova), antonio.cicchetti@mdh.se (Antonio Cicchetti)

the methods used for MPM of CPS. In this paper, we present a systematic mapping study that determines the models, formalisms, and development processes used over the last decade. Additionally, to determine the knowledge necessary for developing CPS, our review studied the background of actors involved in modeling and authors of surveyed studies. The results of the survey show a tendency to reuse multiple existing formalisms and their associated paradigms, in addition to a tendency towards applying transformations between models. These findings suggest that MPM is becoming a more popular approach to model CPS, and highlight the importance of future integration of models, standardization of development process and education.

Keywords: Cyber-Physical System, Model, Formalism, Development Process, Modeling Paradigm, Systematic Review

---

1. Motivation and Background

Cyber-Physical Systems (CPS) are systems that integrate computation, networking, and physical processes. The key characteristic of CPS is their seamless integration of both hardware and software resources for computational, communication, and control purposes, all of which are co-designed with the physically engineered components [1]. Engineering CPS requires physical models, computational models, and network models. Physical models presented as continuous-time models are typically analyzed with solvers that numerically approximate the solutions to differential equations, while computational models are typically discrete-transition models with analysis that enumerate the states (e.g. state machines, dataflow models, synchronous/reactive models, and/or discrete event models). Many challenges arise from the heterogeneity of CPS modeling, such as defining modeling languages, ensuring the determinism of models, and accurately representing the discrete events that are causally related but occur at the same time ([2]).

Furthermore, engineering CPS requires a combination of methods used in different domains (mechanical, electrical, biomedical, etc.) with methods of computer science [3]. This requirement highlights the necessity of having trans-disciplinary modeling approaches which combine different engineering disciplines. Moreover, it is also required to join the classical models with the abstractions as in the case of physical processes (e.g., graphs and differential equations). In literature, there are many efforts in that direction, but no common modeling approach yet covers all of the involved disciplines in CPS.

The act of modeling involves three distinct concepts [3]: the subject that is being modeled, the model itself, and the modeling paradigm with a *formalism*. The subject (system) of modeling is typically either present, potentially

present, or intentionally designed to be deployed in the world, while a **model** is an abstraction of the system [4], which is a simplified representation or view of that system. **Formalisms** are mathematical approaches to describe models, consisting of an abstract syntax and formal semantics [5]. Furthermore, **modeling languages** are concrete implementations of formalisms, expressing systems in a formal and precise way by using diagrams, rules, symbols, signs, letters, numerals, and so on. A modeling language may implement more than one formalism by combining their syntax and semantics [6]. Its definition requires designing its abstract syntax and its formal semantics. In this context, we note the use of **meta-models**, which allow designers to describe modeling languages more abstractly. A meta-model is also a modeling language but a more abstract one; it has its own meta-model (i.e., a meta-meta-model). For model processing and analysis, many **tools** were created such as parsers, simulators, model-checkers, and code generators.

Selection of the appropriate modeling formalisms and tools is a crucial phase in order to handle different aspects of a CPS, as well as the development processes. Specifically, the development activities for modern complex systems, and in particular CPS, encompass multiple technical domains and teams, where each team uses its own set of modeling languages for the aspects of the system that are relevant to that team. Thus, it is necessary to properly integrate these languages in customizable ways.

In this paper, we study the current approaches used for Multi-Paradigm Modeling for CPSs (MPM4CPS) based on a corpus of 153 research studies related to MPM4CPS from the past decade (selected by rigorously reviewing 614 studies). Our aim is to identify, analyze and characterize MPM4CPS approaches and their applications for modeling of CPS. Thus, we investigate the models used in current CPS, and their formalisms, modeling languages, and tools. When a project uses multiple models, it is also important to explore the type of operators used between the models, which directly impacts the deployment aspect. Moreover, the development process plays a big role in the applicability of the approach. In addition, we survey the application domains in which CPS modeling was applied, as well as the roles of stakeholders and skills required for CPS engineers. Based on the analysis of the above data, we identify future research questions for modeling of CPS and related educational programs.

The paper is organized as follows: Section 2 introduce the definition of MPM4CPS; Section 3 describes the method used to perform this study, i.e., the planning and execution details of this; Section 4 presents multi-paradigm modeling approaches for CPS; Section 5 analyse how the approaches were developed; Section 6 presents application domains; Section 7 presents profiles of stakeholders involved in development of CPS; and Section 8 presents the

quality of the papers surveyed. The results are summarized in Section 9. Then, we discuss threats to validity in Section 10, related work in Section 11 and finally, Section 12 ends the paper with a set of conclusions and perspectives for future work.

2. Multi-Paradigm Modeling for Cyber-Physical Systems

The above considerations lead to **Multi-Paradigm Modeling (MPM)** for CPS (MPM4CPS) — a school of thought that advocates the combination of reusable *modular* modeling languages with different paradigms instead of using a single monolithic language for the whole system [7]. The term MPM actually finds its origin in the Modeling and Simulation community in 1996, when the EU ESPRIT Basic Research Working Group 8467 Simulation in Europe (SiE) formulated a collection of research directions and policy guidelines [8]. These identified the need for "a multi-paradigm methodology to express model knowledge using a blend of different abstract representations rather than inventing some new super-paradigm". The proposed vision also recommended that all parts and aspects of a system should be modeled explicitly at the most appropriate level(s) of abstraction, using the most appropriate modeling formalisms to deal with engineering heterogeneity.

At that time, there was not yet any precise definition of MPM. However, the work of working group on foundations of MPM4CPS during the COST action IC1404 initiated the recent work [9, 10], which provided a more precise definition of MPM. In this survey, we used this definition that we introduce in the following.

In computer science, paradigms, which emerged as early as 1986 [11], are known as a means for classifying programming languages. For example, Eiffel is *object-oriented* and supports *contract-based design*. Prolog is *declarative* while Lisp is *functional*. A paradigm thus characterizes languages through their syntax and semantics. For example, *object-orientation* imposes viewing the world in terms of communicating objects typed by classes, whereas the declarative paradigm relies on term substitution and rewriting. The idea of combining several paradigms into a single programming language led to more powerful languages such as Java, which is at the same time imperative, object-oriented, concurrent, real-time and functional and Maude, which is declarative, object-oriented, concurrent and real-time, among many others examples.

In [9, 10] the authors consider modeling as a generalization of programming so that programming languages are a subset of modeling languages and therefore, programming paradigms are also a subset of modeling paradigms (the aforementioned programming paradigms are also modeling paradigms).

Like for programming languages, modeling paradigms also characterize both the languages, their formalisms and their semantics. But in addition, modeling paradigms may also characterize the *workflows* (or development processes) that define how models of the modeling languages are used during development. Following this, the example programming paradigms of above (*object-oriented*, *contract-based design*, *declarative* and *functional*) are simple paradigms that only characterizes languages and not the workflows. Conversely, we can say that *agile programming*, which only characterizes workflows is also a simple paradigm.

In [10], two examples of more sophisticated paradigms are given such as Discrete Event Dynamic Systems (DEv) and Synchronous Data Flow (SDF). For the first one, it characterizes formalisms by the fact that the time is continuous, systems dynamics is captured through timed events and that systems state is composed of variables that range over discrete domains. An example of formalism that follow the DEv paradigm is Timed Finite State Automata (TFSA).

For SDF, languages that follow this paradigm are characterized by describing systems computations as a directed graph where signals flow through the arcs representing infinite streams of data, each data piece being called a sample. Nodes represent computation units that execute whenever enough input data become available. They have ports that explicitly define how many samples are used (consumed by inputs, or produced by outputs). Finally, a memory full node should always define an extra port corresponding to initial conditions. An example formalism that follows the SDF paradigm is Causal Block Diagrams (CBDs).

As we can see from these examples, a modeling paradigm can be broadly defined as a set of properties that characterizes languages, formalism and workflows employed in the development of CPSs (see [10] for a more formal definition). This makes that MPM is wide spread. For instance, even if a modeling environment only cover one domain of the system (say embedded system), or consists of only one language, like for programming languages, the language is often multi-paradigm, especially for the case of architecture description languages. For instance, the AADL language follows both the object-oriented and SDF paradigms (but does not follow the DEv paradigm since its notion of time is discrete), and probably more paradigms would appear after a closer look. MPM is even more frequent for CPS development environments that cover different system domains, since often domains make use of paradigms of their own. But at the same time, it can be the case that a single modeling paradigm is used for very different domains. Obviously this is more often the case for paradigms that characterize workflows (e.g., agile).

While somewhat modular and extensible, the current approaches to MPM are limited in their scalability, customizability, and comprehensiveness — as required for practical industrial contexts. In part, the adoption is hampered by lack of broad awareness of the existing MPM methods. Due to their heterogeneity, it is difficult for a practitioner to determine what MPM models/tools exist, what aspect of the system they represent, which domain(s) they apply to, and what engineering training is required to use them.

3. Research Method and Review Process

The methodology used in this Systematic Mapping Study (SMS) follows the process suggested by Kitchenham [12, 13] (see Figure 1). The review process has three main phases: *planning*, *conducting* and *reporting*. This SMS was performed by researchers (PhD students, Post-Docs, Professors and Research Engineers) from various universities, research institutions, and industry, from ten countries: Portugal, Serbia, Sweden, France, Turkey, the Netherlands, the United States, Belgium, Latvia, and the Czech Republic.

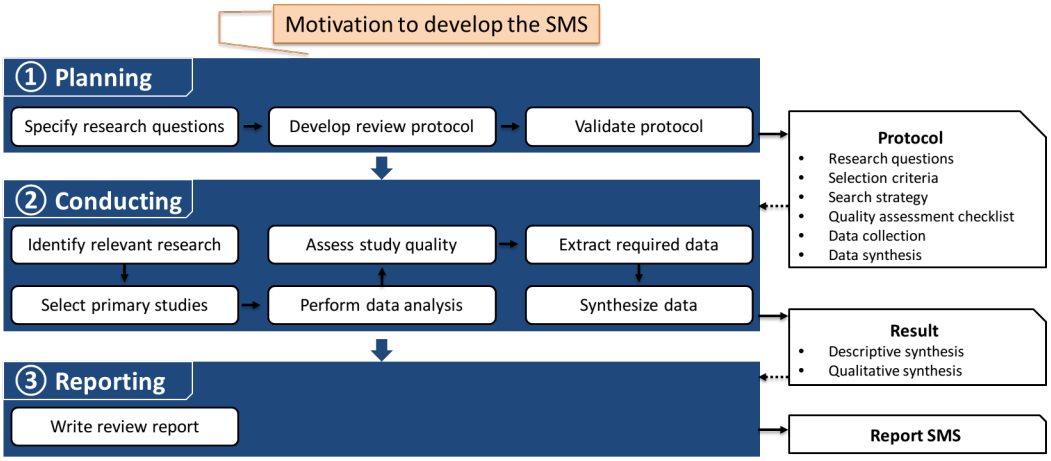

Figure 1: Overview of the review process based on [12]

The motivation to perform SMS originated from the MPM4CPS European COST Action [14]. A working group focused on education and dissemination activities was assigned with the mission to conduct this SMS, whose progress was reported in [15]. In the next subsections, we outline how the first two phases were carried out for this SMS.

3.1. Planning

Kitchenham [12] proposed three stages within planning: (1) *specifying research questions*, (2) *developing a review protocol*, and (3) *validating it*.

Further we highlight the results of each stage, which are detailed in our complimentary data submission[1] [16].

Table 1: PICOC criteria used for SMS

Population	composed of primary studies on multi-paradigm modeling of CPS. It was not limited to any specific industry, system or application domain;
Intervention	clustering the MPM approaches for CPS, models, their application domains, and other characteristics;
Comparison	not applicable to this study;
Outcome	the overview and insights into MPM techniques for CPS;
Context	the set of research papers from scientific libraries, which report MPM4CPS;

We applied the *PICOC criteria* [17] which is systematic approach for the definition of our research question [13] in Table 1. The overall objective of our study is to *provide an overview of the current state-of-the-art of approaches supporting the MPM of CPS*. The goal of this SMS is to *answer four research questions*, further articulated in 13 sub-questions which are presented in Table 2. The research questions were devised over several work sessions of the Education and Dissemination working group and in the plenary meeting of MPM4CPS COST Action [15].

Table 2: Formulated Research Questions

RQ1	**What multi-paradigm modeling approaches exist for CPS?**
Q1	Does the paper report a multi-paradigm modeling approach?
Q1.1	Which development phases are supported by MPM approach?
Q1.2	Which part(s) of the CPS is(are) modeled?
Q1.3	Which formalism(s) is(are) used for modeling the CPS?
Q1.4	What is the integration mechanism for the presented models?
RQ2	**How are multi-paradigm modeling approaches for building CPS presented?**
Q2.1	Does the paper report a model/meta-model?
Q2.2	Does the paper report a tool/language?
Q2.3	Does the paper report the model of the adopted development process?
RQ3	**What CPS application domains have been modeled?**
Q3	Is the approach domain-specific?
Q3.1	Which application domain is addressed?
RQ4	**What is the profile of the person who performs CPS modeling?**
Q4	Does the paper report the actors/stakeholders involved in modeling of the CPS?
Q4.1	Does the paper report the modeler's background knowledge?
Q4.2	What are the research fields of the authors?

The review protocol included the selection of *Research questions* and *Search sources*, *Search keywords and query*, *Inclusion and exclusion criteria* for the primary papers, *Data extraction strategy* and *Quality assessment strategy* with a **data form** to support data collection [18].

We evaluated our research protocol threefold (for more details see [15]): *1) Query Testing.* To clearly define keywords, we analyzed all of our research

[1] https://data.mendeley.com/datasets/jy6ww3hmyw/4

questions separately and divided the keywords into three main groups: (i) cyber-physical systems, (ii) modeling approaches for cyber-physical systems, (iii) combination of modeling approaches for cyber-physical systems. For each group we created a search sub-string, and then integrated the sub-strings into the final **search string**. We defined several versions of query strings and conducted the search process over databases. The members of MPM4CPS COST action voted during the work-group meeting in order to select the best candidate.

2) Validation Survey with MPM4CPS COST Action Members. A survey asked for feedback on the research protocol. An online form was created to collect general information about the participants and their expertise on SMS; and set of validation questions. A total of ten participants participated in this survey, and we updated the protocol according to the results.

3) Pilot session with reviewers. During this session, three participants evaluated the questions in the **data form** where each one of them was assigned two primary studies to read and classify. The reviewers expressed their doubts and eventual problems which they encountered while providing their answers. The completeness of the answers given in text boxes was also evaluated and discussed. As a result, we updated the instructions for each affected question.

3.2. Conducting

Conducting the SMS involved six phases discussed in this section (see Figure 1): (1) *identifying relevant research*, (2) *selecting primary studies*, (3) *performing data analysis*, (4) *assessing quality of studies*, (5) *extracting required data*, and finally (6) *synthesizing the data*.

3.2.1. Relevant research identification

The main goal of an SMS is to identify a large number of primary studies related to the research questions. We obtained data for this SMS twofold:

AUT Using automatic search over digital libraries: *ACM Digital Library (ACM), IEEExplore (IEEE), Science Direct (SD), Springer Link (SL), Scopus,* from *2006* that was considered by experts as adequate for this SMS. The search was executed two times; first time in the beginning of 2018 where we included primary studies including the year 2017; second time in April of 2021 where we included all other primary studies. We obtained 379 studies in total.

MAN Manually extracting the primary studies from another SMS reported in [19] for the period *2011-2017*. This study intersects with ours since it selected studies which were reporting modeling approaches for building

CPS, however, the search string was wider and included the sustainability concern. We obtained 240 primary studies from this source.

We used the following **search string** to obtain primary studies using automatic search:

("multi-paradigm" OR "multi-formalism" OR "heterogeneous formalism" OR "unified modelling formalism" OR "multi-model language") AND ((("cyber physical" OR "cyber-physical" OR cyberphysical OR smart) AND system) OR CPS)AND ("modelling approach" OR "modeling approach" OR "integrate modelling" OR "integrate modeling" OR "model driven" OR "model-driven") AND ("software engineering" OR "software system")*

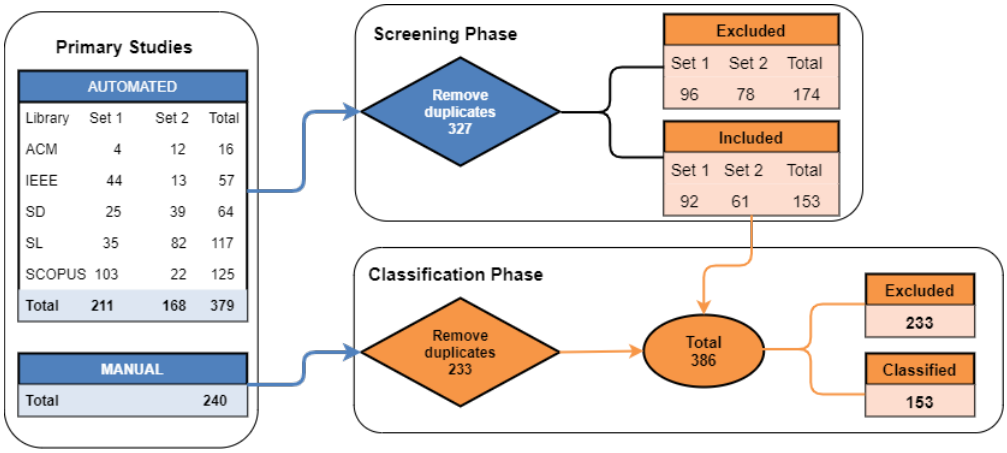

Figure 2: The steps performed during the conducting phase, with study counts

3.2.2. Primary studies selection and data analysis (Screening Phase)

The ReLiS [20] platform was used to support the screening and classification phase since it supports multi-user data extraction [21]. Only primary studies from automatic search passed through the screening phase as ones from manual search were already classified to report modeling approaches for CPS. After removing duplicates, 327 primary studies were imported into the ReLis system (see Figure 2). We defined a set of *inclusion* and *exclusion* criteria (Table 3) to filter out candidate primary studies that would not contribute to answering our research questions. This filter was applied to the *title*, *abstract* and *keywords*. Each study was assigned to two reviewers. In case of a conflict, the reviewers discussed to resolve it. At this stage, our tendency was to be more inclusive than exclusive. From the screening phase, almost half of primary studies were included (153) and 174 were excluded.

Table 3: Inclusion (I) and Exclusion (E) criteria

ID	Criteria
I1	Publication date from 1/1/2006
I2	Relevance to the research questions
I3	Explicit mention of *multi-paradigm modeling of cyber-physical system*
I4	Papers that report methodologies, metrics, or formalisms for modeling CPS
I5	Analysis of relevant application domains for modeling of CPS
E1	Informal literature (powerpoint slides, conference reviews, informal reports) and secondary/tertiary studies (reviews, editorials, abstracts, keynotes, posters, surveys, books)
E2	Duplicated papers
E3	Papers that did not report a *multi-paradigm modeling method for CPS*
E4	Papers with the same content in different paper versions
E5	Papers written a language other than English
E6	Purely hardware or electrical engineering papers

3.2.3. Data extraction and quality assessment (Classification phase)

We additionally imported 240 studies from manual selection into the ReLis system, which automatically identified 7 duplicates. In total, there were 386 papers to review, 153 from automatic search, and 233 from the manual study after removing duplicates, which are reported in Part 2 in our data repository[2] described more in detail in [16] [3] (see Figure 2). The data extraction form was generated in ReLis using the integrated domain-specific language for specifying forms. The form consisted of the questions (Table 2), descriptive instructions, as well as from the *quality assessment* questions (see Table 12). All questions had at least one text paragraph so reviewer observation could be described freely, and when applicable, included a yes/no field and/or multiple-choice questions with predefined categories. Quality questions were defined to observe the impact of the studies. Therefore, the several criteria about ranking, relevance, clarity of the problem, clarity of the research context, method evaluation, contribution presentation and future research directions served as a complement to the extracted data related to the research questions and were used later in result analysis.

Each study was randomly assigned to one expert. Reviewers were provided with a set of instructions and videos about to use the ReLis system, how to extract data, and when to exclude papers. They were able to monitor the individual and overall classification status. To reflect the confidence of reviewers, we used two self-assessment criteria: reviewers confidence about content of the study and reviewers confidence about quality of the study. When a reviewer was not confident about a paper under review, an additional reviewer was assigned to review the paper and the assessment scores were then discussed until consensus was reached. In total, there were 153

[2] https://data.mendeley.com/datasets/jy6ww3hmyw

[3] https://www.mendeley.com/community/smr-mpm4cps-public/

classified papers and 233 excluded papers.

3.2.4. Data synthesis

During this stage, the previously extracted information was cross-reviewed by all experts with the aim of identifying possible clerical errors by taking into account the research questions. To understand the modeling approaches reported in the primary studies, at least two reviewers were assigned to perform manual qualitative data analysis for each question from Table 2, in order to cluster the approaches in groups by similarity. Different reviewers described approaches with different terms which were in practice equivalent, and these synonymous terms were merged. Reviewers processed the extracted data to identify unambiguous labels for each question under analysis and a tentative cluster name. Finally, reviewers iteratively refined and merged the data until a small number of well-defined clusters emerged. Throughout the iterations, the researchers cross-checked their work by having automatic validation formulas and synchronized their definitions of the clusters.

4. Multi-paradigm Modeling (MPM) Approaches (RQ1)

In this section, we present the findings for **RQ1: What multi-paradigm modeling approaches exist for CPS?** For this, we focus on four subquestions that are the completeness of the approach with respect to the supported development activities, the modeled CPS components, the employed formalisms and the integration mechanism to support combining the different employed modeling languages and their tools. Each of these question is addressed in the following subsections.[4]

4.1. Completeness of reported MPM approaches

As described in [22], the adoption of model-driven development (MDD) depends on the practicality of the tools as much as the scientific merit of the underlying methodology. Modeling tools should allow for capabilities including code generation, model execution (for simulation), and integration with other legacy tools. In [23], a list of challenges for the adoption of MDD is presented emphasizing that toolkits are often missing these capabilities, especially for code generation and reverse engineering. Ideally, such capabilities should be supported within a single toolkit, as a complete methodology implemented by a single modeling tool. However, this is not realistic since

[4]Note that unless otherwise stated, the percentages in this section are relative to the total number of MPM papers.

the complete development of real CPSs involves so many domains each having its own sophisticated tools. Instead, well integrated tool chains should be developed.

Following this, we analyzed the primary studies to evaluate the completeness of the presented MPM approaches according to the number of supported activities (Q1.1 Table 2). These were classified into the categories of Table 4, mainly distinguishing model-related tasks.

Table 4: CPS Development Phases and Descriptions

Phase	**Description**
Editing	Drawing or editing models
Simulation	Running the model or its generated code
Transformation	Converting a model to a different form, whether to a model of different aspects or levels of abstraction or to a different formalism
Analysis	Providing information to the user about the properties of the model
Verification	Performing a more formal analysis to determine if the system fulfills some properties
Integration	Relating heterogeneous models, especially those developed with different tools
Generation	Creating code, test cases, or documentation from a model.

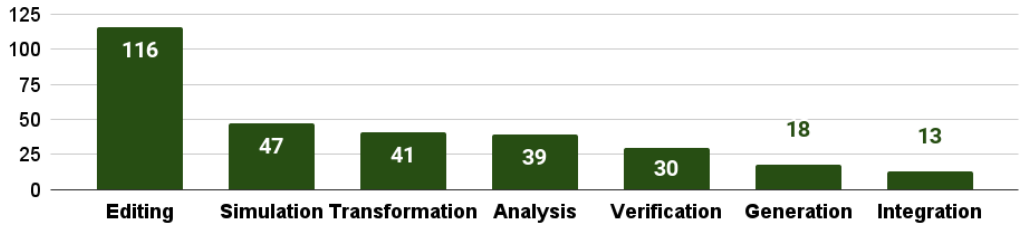

Figure 3: Phases supported by CPS development MPM approaches

All 153 MPM approaches support at least one of the aforementioned phases, 67 (43.8%) support exactly one phase, 61 (39.9%) support exactly 2 phases, 12 (7.8%) support exactly 3 phases, 4 (2.6%) support exactly 4 phases, while 9 studies are complete (i.e., covering all phases) [24, 25, 26, 27, 28, 29, 30, 31, 32].

A large majority of approaches support editing (116 (75.8%)) followed by simulation (47 (30.7%)), transformation (41 (26.8%)) and analysis (39 (25.5%)) (see Fig. 3). Integration capabilities are reported by only 13 (8.5%) studies.

It is worth noting that some of the approaches might support additional capabilities beyond the ones in the reviewed papers.

4.2. Reported modeled components in MPM4CPS

MPM approaches may target an entire CPS or only certain parts of it. The classification intended to determine the scope targeted by the papers, i.e.

whether the approaches considered the *hardware*, *software*, *network*, or *environment* of the system (and possibly combinations of the above) to answer Q1.2 (Table 2). Hardware includes electronic and mechanical components, while software includes the code of the system. A connected system may need its network modeled as well as its environment since a CPS always interacts with the outside world.

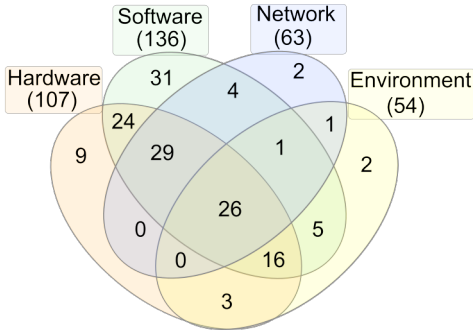

Figure 4: CPS components supported by MPM approaches

Twenty-six (17%) MPM papers reported modeling all four main components of CPS [25, 28, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56]. Most frequently (29 (18%)), the papers modeled the hardware, software, and network of the system (see Fig. 4). In total, most approaches included some modeling of software (136 (89%)) or hardware (107 (70%)), while fewer approaches modeled the other major components, network (63 (41%)) and environment (54 (35%)), or other parts such as stakeholders (7 (%10)). As expected, a majority of papers (95 (62%)) reported modeling some combination of hardware and software.

Not all CPS are connected systems, and while all CPS must function in an environment, most approaches focus on the internal operations and do not often consider to model the system interaction with its environment.

4.3. Reported modeling formalisms

A wide spectrum of formalisms is used for modeling CPS. However, we first note that many studies (77 out of 153) do not state explicitly the employed formalisms, so that these can only be inferred to some extent through the mentioned languages and tools (see section 5.4).

Out of the 79 papers that explicitly mentioned formalisms, Fig. 5 lists each of these formalisms with the corresponding number of papers where they are used in order to address Q1.3 (Table 2). Note that more details on what each formalism category includes is given in table 5.

We note that most of the papers (18 (22.8%)) report using Petri Nets-like formalisms followed by Hybrid Automata (12 (15.2%)), which is a mathematical model for precisely describing systems in which digital computational processes interact with analog physical processes. Discrete Events, which models the operation of a system as a sequence of events in time are also popular (11 (13.9%)). Dataflow, a graphical representation of computation and flow of information among computations, Temporal Logic and Differential Equations are also often reported for MPM for CPS. More details about these formalisms may be found in [57].

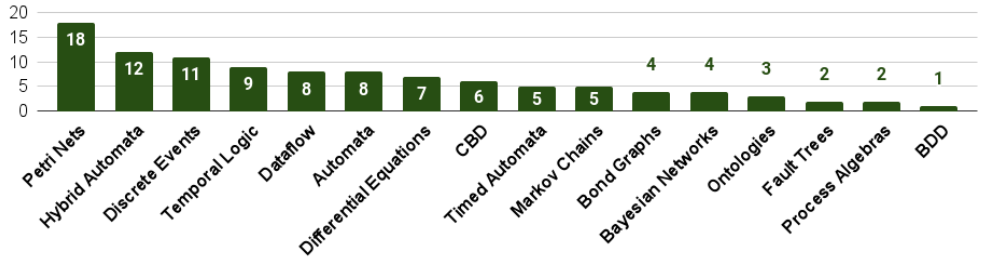

Figure 5: Reported modeling formalisms

Out of 79 papers that report formalism(s), a majority of approaches deal with only one formalism (56 (24.1%)). Nevertheless, these papers were still considered as MPM because of our broad definition of MPM as discussed in section 2. Nineteen (24.1%) papers report using 2 formalisms and only 4 (5.1%) studies are using 3 formalisms. This shows that multi-formalisms approaches are relatively scarce, but that may be due to the lack of formalism being made explicit in many approaches and also to the difficulty of developing multi-formalism approaches. This topic and more details on the encountered multi-formalisms approaches are discussed in section 5.4.

4.4. Mapping between formalisms and modeled components

Given the results of section 4.2, we can map which formalisms are used for which components of the system. For this, we relate the results of section 4.3 with those of section 4.2. For each formalism of figure 5, table 5 shows the number of studies that mention this formalism for each component, prorated by the total number of approaches covering the component, after removing the approaches that do not explicitly state their formalism. In this way, we can compare how often a formalism is used for a component scaled by the number of approaches that were found to cover this component. In this way, formalism usage across components can be properly compared.

Table 5: Employed formalisms per system components

Formalism	Software	Hardware	Network	Environment
Petri Nets (Stochastic Activity Networks / Timed Stochastic Petri Nets)	13.5	19.5	17.3	18.6
Hybrid Automata	11.5	13.0	13.5	7.0
Discrete Events	11.5	11.7	13.5	9.3
Temporal Logic (CTL, LTL, TCTL, PCTL)	8.3	7.8	5.8	11.6
Dataflow	8.3	6.5	5.8	7.0
Automata, Finite State Machines, State Machines	8.3	6.5	3.8	7.0
Differential Equations	7.3	6.5	9.6	7.0
CBD (Causal Block Diagrams)	5.2	6.5	3.8	9.3
Timed Automata	4.2	5.2	5.8	4.7
Markov Chains	5.2	5.2	7.7	4.7
Bond graphs	4.2	3.9	3.8	4.7
Bayesian Networks	4.2	3.9	3.8	7.0
Ontologies	3.1	0.0	0.0	2.3
Fault Trees	2.1	1.3	1.9	0.0
Process Algebras, Statecharts	2.1	1.3	1.9	0.0
BDD (Binary Decision Diagrams)	1.0	1.3	1.9	0.0

The results of table 5 show that the overall popularity of formalism as shown by the order of the bars of figure 5, which is the same as that of the first column of table 5, is relatively independent of the component category. A general trend is that each formalism is roughly used at the same level whatever the CPS component is, with unexpected results for Petri Nets and differential equations, which seem to be more used for other components than software. Similarly, Differential Equations seem to be more used for Network. In this latter case, this is mainly due to some approaches such as [33, 34, 35, 36], which cover all components but do not always explicitly state which formalism is used for which component. Besides, there are also approaches like [58], which use Differential Equations combined with a Markov model to analyze properties such as the resilience of a network. Hence, a deeper analysis would be required to make this mapping analysis more meaningful and useful.

4.5. Reported integration mechanisms

The heterogeneity of CPS and MDE pragmatics require that several models be typically employed to support the various modeling activities pertaining to different domains and levels of abstraction. As a consequence, there is a need to provide integration mechanisms to synthesize the information coming from different modeling activities and perspectives. We identify several MPM approaches which integrate models to answer Q2.4 from Table 2.

A first remark is that despite the importance of model integration, a large number of MPM approaches (41 (26.8%)) either do not consider model integration or do not state explicitly what model integration mechanism is

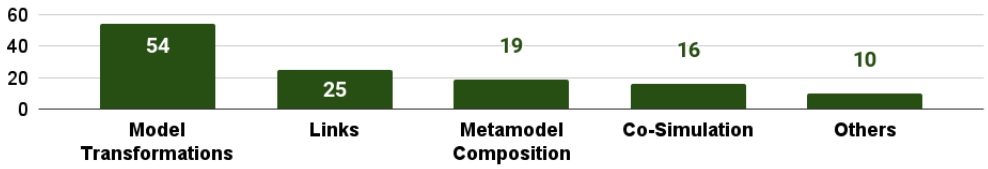

Figure 6: Reported integration mechanisms

used. This result is not surprising since integrating models from different domains is in general an open challenge. We therefore exclude these papers from this particular analysis and only retain the remaining 112 papers that state integration mechanisms.

The dominant model integration approach is *model transformations*, reported in 54 (48.2%) primary studies, where a model is transformed into another model and then used to perform a given activity. Also, model transformations can be used to transform a model into another model providing semantics to the language of the original model (semantic domain). This last category is called *semantic anchoring*. Furthermore, some of these approaches like [55, 27, 59, 60, 46] integrate models by translating all of them into a unifying formalism, which can be Bond Graphs, Discrete Events or even executable code from which analyses of the integrated model can be performed. Among such model transformations, we can distinguish different cases. Most approaches make use of the simplest case where a *batch transformation* creates a whole model from a set of source models. We also find a few approaches such as [61, 62] where *incremental* (also called *live*) *transformations* are used, capable of punctually updating the parts of a target model necessary to maintain consistency given the changes in a source model.

The second most frequent approach makes use of traceability *links* between models to support their integration, as reported by 25 (22.3%) studies. The models are presumably generated by hand and traces are established to represent dependencies between the models and to ease navigation between them. In this way, users can be notified when a change in a model impacts other models so that the impacted models can be inspected to ensure consistency. We note that some of these approaches like [63, 41] make use of explicit model interfaces for specifying how models can be linked.

We find 19 (17.0%) integration approaches that focus on integrating the modeling languages, with dedicated links between their *metamodels*. These approaches often propose specific extensions of existing languages such as UML and SysML to capture aspects of the system that cannot be modeled with the original language.

Sixteen (14.3%) approaches are dedicated to integrating the models at simulation time, namely *co-simulation* approaches. In these approaches, existing simulators for models to be integrated are coupled to constitute a new simulator for the integrated models. Some of these approaches such as [43] make use of the Functional Mock-up Interface (FMI) standard.

Ten (8.9%) approaches are classified as "Others" in figure 6. Among these, 3 approaches mention the use of an *integration language* to integrate models. In these approaches, the integration language is used to specify component interfaces and composition operators between model elements. Four approaches use integration mechanisms based on the specification of constraints over models of different domains that can be evaluated to detect inconsistencies. For instance, this can be supported with ontology languages such as in [25, 64]. Another approach combines the language's models of computation (MoC) so that the language's semantics is also integrated. We finally find more exotic approaches making use of message exchange between the models, knowledge-driven enrichment or programming theory.

Regarding the combination of the identified integration mechanisms, the most frequent combination is Model Transformations and Links [51, 65, 31, 66]. The combinations of Model Transformations and Metamodel Composition [67, 68], Model Transformations and Co-Simulation [34, 65], Links and Metamodel Composition [69] and Links and Co-Simulation [65] are also found. Finally, only one approach jointly uses the 3 mechanisms of Model Transformations, Links and Co-Simulation [65].

5. Presentation of MPM4CPS approaches(RQ2)

In this section, we present the findings for our second research question **RQ2: How are multi-paradigm modeling approaches for building CPS presented?**. All primary studies (in total 153) were classified to report modeling approaches for building a CPS as it was an inclusion criteria. We analyzed which parts of CPS were modeled, and which qualities were considered in modeling approaches. Next, we analyzed which approaches report models and meta models, which tools and modeling languages are used for development of CPS and finally, if the development process for CPS was presented as part of the research.

5.1. Modeled CPS aspects and considered CPS qualities

We analyzed which aspect of the CPS was modeled, and grouped them into six clusters as described in Table 6. One primary study could address several aspects. Most of studies, 75 (49%) present the high-level structure of the CPS, while 48 studies model development artifacts, 34 model software

and digital parts of system, 26 physical parts, and 10 studies the hardware part of CPS (see Figure 7).

Table 6: System aspect clusters and descriptions

Cluster	Description
High-level structure, dynamics, or components	Models tagged with this label include abstract models that do not distinguish between cyber, hardware, or physical dimensions — often due to their high level of abstraction or particular focus. Examples include system architecture components, failure states, and interactions between a system and its attacker.
Development artifacts, processes, or concepts	In this cluster, models are focused on various aspects of engineering a system — rather than on a system itself. The examples include metamodeling, ontologies, and modeling of development workflows.
Cyber, software, or computation	Some models represent the software and digital aspects of the system. The examples include state machines of a program and a software architecture diagram.
Physical dynamics or components	Some models represent physical processes directly, without mixing them with digital aspects. For example, differential equations modeling the system's mechanical movement are in this category.
Computational hardware or infrastructure	When modeling concerns the hardware that is a part of the system and under its control (such as CPU, GPU, networks), we used this category.
Other	This category contains the parts of the system that did not fit into the above clusters, including human models and business processes.

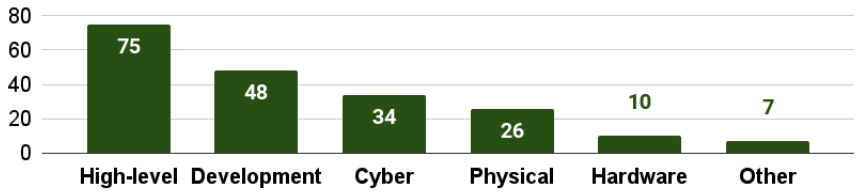

Figure 7: Primary studies clustered regarding modeled aspect of CPS

Furthermore, we wanted to understand which qualitative characteristics of CPS were addressed by modeling approaches. We recognized 14 clusters representing CPS qualities/activities as described in in Table 7. We observed that most of approaches, 42 (27.45%), aim to automate development tasks during construction of CPS (see Figure 8). Other main focus areas are correctness 25 (16.34%), efficiency 22 (14,38%), and resilience 18 (11.76%); additionally, around 10% of studies address safety (16), integration (16), and security (15), while simulation (13), v&v (12), and performance (11) concern about 8% of the studies. Finally, only around 4% of studies target flexibility (7) and complexity (6) of the CPS.

5.2. Instance models and meta-models for CPS

A meta-model is a model describing another model, here called instance model. The studies on modeling of cyber-physical systems have been analyzed according to these two levels of abstraction (see *Q2.1* in Table 2).

Table 7: System Qualities and Descriptions

Cluster	Description
Automating Development	Aims at automating development tasks related to the construction of the system, such as code generation, design space exploration, modeling support, and others. This cluster does not include modeling approaches that support verification, validation, simulation, or complexity reduction (those are separate clusters).
Efficiency	Improves or analyzes an existing mechanism/process in terms of its quality and cost metrics, such as throughput, energy lost, operations performed in a time unit, and etc.
Resilience	Improves or analyzes reliability, robustness, fault tolerance, or dependability.
Correctness	Aims at the functional correctness of the system, often in terms of providing the correct output or demonstrating the existence of a solution (e.g., to a scheduling problem).
Safety	Aims at reducing or analyzing the possibility or probability of critically undesirable outcomes, usually in safety-critical or mission-critical systems.
Security	Seeks to protect the system against an intelligent attacker or analyze its vulnerability to such attackers.
Integration	Enables combination or inter-operation of parts (components, algorithms, subsystems, models, etc.) that could not be combined before. Meta-modeling is a prominent example of an integration approach.
Performance	Improves or analyzes the outcomes of predictive (e.g., classifiers) and other systems that did not fit into correctness or efficiency.
V & V	Supports verification and validation activities in the development cycle.
Flexibility	Makes the system or its development more customizable, changeable, or adaptable.
Complexity	Reduces the complexity either of the system's artifacts or the behaviors of the system.
Simulation	Supports improved simulation, usually in terms of higher fidelity or faster speed.
Other	Did not fit into the other clusters. These include better business outcomes, human usability or trust, privacy, and other miscellaneous system qualities.

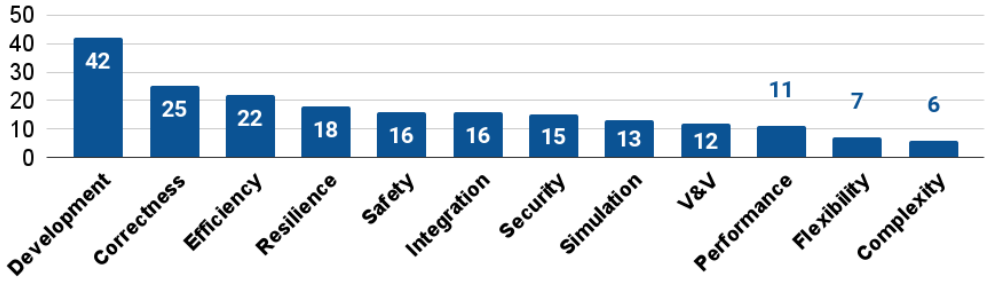

Figure 8: Primary studies clustered regarding considered CPS qualities

About 89% (133) primary studies describe an instance model and/or meta-model, while 10.6% (20) do not (see Figure 9). Most of studies, 113 (73.8%) of them, demonstrate an instance model, while 47 (30.7%) present a meta-model, and only 8 (5%) define both.

A model can represent the approach or illustrate the use case for a case study used to validate an approach. We note that most of studies, 68 (44.4%) of them, contain an instance model of the use case, while 45 (29.4%) present an instance model for the approach. Regarding meta-models, 21 studies (13.7%) use a meta-model for use case and 26 (16.7%) — meta-model for the approach.

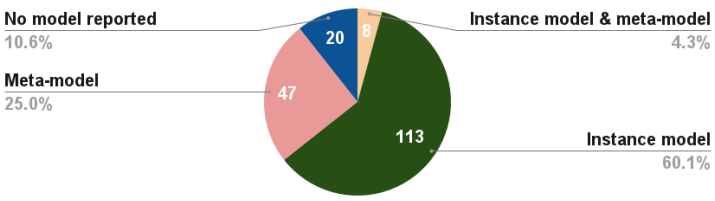

Figure 9: Reported models and/or meta-models

5.3. Tools and modeling languages for building CPS

Next, we analyze which tools or modeling languages are used for CPS development (Q2.2 in Table 2). About 86% (132) of the primary studies report using tools or modeling languages, while about 14% (21) do not. Specifically, 61% (93) of the studies report using existing tools and modeling languages. On another hand, 25.5% (39) of the studies report having developed their own tool or extending an existing one. We highlight the tools and modeling languages which were reported in more then three studies in Figure 10 and Figure 11, respectively.

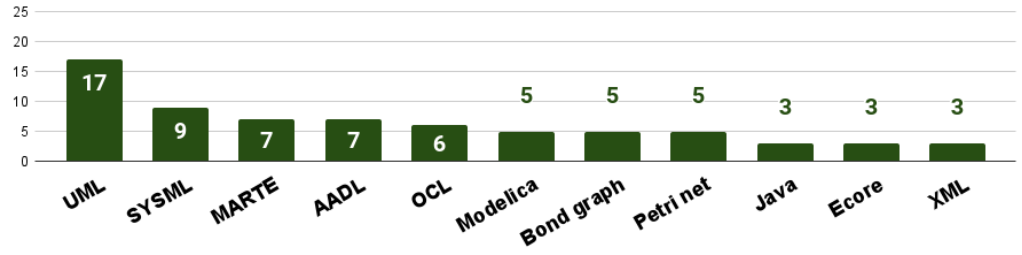

Figure 10: Reported modeling languages for CPS development in primary studies

We note that most commonly used modeling languages are: the Unified Modeling Language (UML), which is a standard method to visualize the design of a system in software engineering; Systems Modeling Language (SYSML), which is an extension of UML targeting systems engineering; Modeling and Analysis of Real Time and Embedded real time and embedded systems (MARTE), which is an extension for the UML targeting real-time systems; and Architecture Analysis & Design Language (AADL) language, which supports modeling the software and hardware architecture of an embedded, real-time system. Finally, the Object Constraint Language (OCL) is a declarative language describing rules applying to UML models. It is interesting that only 5 primary studies report using Modelica, an equation-based language that can be considered as a domain-specific language to conveniently model complex physical systems.

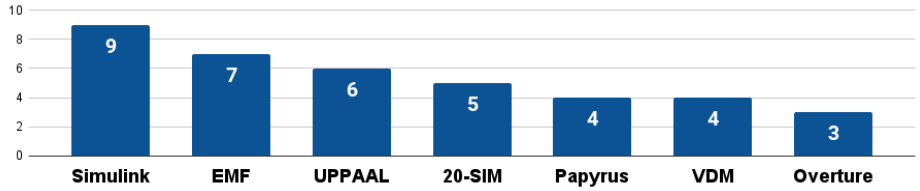

Figure 11: Reported tools for CPS development in primary studies

Regarding the tools, Simulink demonstrated to be a popular environment (reported in 9 studies); it supports modeling, simulating and analyzing multi-domain dynamical systems, and is based on MATLAB, a multi-paradigm numerical computing environment, reported separately in the tools list in less than 3 studies. We also identified several modeling environments being reported, of which mostly used (in 7 primary studies) is the Eclipse modeling framework (EMF), which in the context of this study is primarily used for its language workbench features including code generation for building tools and other applications based on a structured data model. In 6 studies, UPPAAL, an integrated tool environment for modeling, validation and verification of real-time systems, is reported. 5 studies report using 20-sim, a commercial modeling and simulation tool; 4 mention Papyrus, an open-source model-based engineering environment. 4 papers report using VDM, a toolbox for formal modeling of complex systems; related to this, additional 3 papers mention Overture, which is an Eclipse-based open-source IDE for VDM languages.

5.4. On the combination of formalisms/languages/tools for building CPS

One of the main features of MPM approaches is expected to be the use of multiple formalisms, languages, and/or tools, to design appropriately the system under study through different points-of-view (see Section 2). Therefore, from research questions Q1.3, Q2.2, and Q2.3 we extracted also the use of multiple formalisms, languages, and tools, respectively. Here, it is worth to remind that formalisms, languages, and tools have to be explicitly mentioned in the papers, hence in general the sum of papers resulting from these analyses is less than the total number of included studies.

As mentioned in section 4.3, a few MPM approaches were found to employ more than one formalism. Among the 19 (24.1%) approaches that use at least 2 formalisms, Petri Nets are the most often combined with Hybrid Automata [70, 63], Markov Chains [63, 28], Discrete Events [71], Bayesian Networks [54], Automata [72] and Timed Automata [44]. Then comes Discrete Events, which is combined with Petri Nets [71], Automata [38, 73],

Hybrid Automata [74] and Bayesian Networks [39]. We also find two different combinations for each of the Dataflow and Causal Block Diagrams formalisms. Among the 4 (5.1%) combinations of 3 different formalisms that were found, we find the combinations of Temporal Logic, Hybrid Automata and Timed Automata [40], Petri Nets, Hybrid Automata and Markov Chains [63], Petri Nets, Discrete Events and Binary Decision Diagrams [71] and finally Temporal Logic, Hybrid Automata and Differential Equations [75]. A more comprehensive analysis of the formalisms combination can be found in the data-set of this study.

Regarding the languages, 50% (45) of the papers mentioning explicitly the language(s) use 2 or more languages, with 31.1% (28) exactly 2, 10% (9) exactly 3, 5.5% (5) exactly 4 and even 3 papers mentioning the use of 5 different languages. Instead, when considering the tools, the majority of the papers mentions exactly 1 tool, specifically 64.7% (44). Two tools are mentioned in 29.4% (20) of the papers, 3 in 7.3% (5), and 4 in 4.4% (3).

After analyzing formalisms, languages, and tools separately, we intersected the works mentioning formalisms (Q1.3) with the formalisms "inherited" by using the mentioned languages (Q2.2b) or tools (Q2.2c). In this respect, we adopted a catalog of formalisms and corresponding languages/tools available as project deliverable of the MPM4CPS COST action IC1404 [5]. The analysis results show that very often a formalism is mentioned together with a corresponding language, while less frequently with a corresponding tool. Since we extracted a relevant number of novel tools (see Q2.2a) and about half of the papers do no mention any formalism explicitly (see Q1.3), it could be possible that formalisms are more frequently left implicit when presenting an approach through a tool.

5.5. Development processes for building CPS

Finally, we analyzed whether the papers propose a process for engineering CPS to answer Q2.3 (see Table 2). More than 40% of the primary studies did not present a development process for CPS. Among studies reporting a development process, the description of the process is given at different levels of formality: 30 (19.6%) of the papers describe the process in an informal, often partial way; 32 (20.9%) describe the process semi-formally, step by step, but still giving only textual descriptions; and only 29 primary studies (18.9%) present their process by means of a formal model.

[5] https://zenodo.org/record/2538711#.YHAaFzw6_0M

6. Application domains for CPS (RQ3)

In this section, we present the findings for **RQ3: What CPS application domains have been modeled?**. The analysis shows that one third of the classified papers (51 papers) present domain-specific approaches, while a majority of the papers (102 papers) propose domain-independent approaches, meaning that they are applicable to CPS in general.

Eight main application domains were suggested for the reviewers based on Gunes et al. classification [76]. The studies which do not fall in any of these categories were assigned to the "Other" category and the reviewer articulated the application domain as description. It is worth to mention that there may be more than one application domain for a study. The reviewers were allowed to specify up to three application domains for each primary study, independent of if the approach is domain-specific or not. The reason behind this is that even for general approaches, the application can be illustrated using one or more domains. We have 34 studies (22.2%) that did not fit into proposed domain categories (see Figure 12). The proposed application domains in the Others category include: Automotive, IoT and WoT, Mechatronic systems, Digital Factory, Self-adaptive software systems, System of systems, and Web services, among the others.

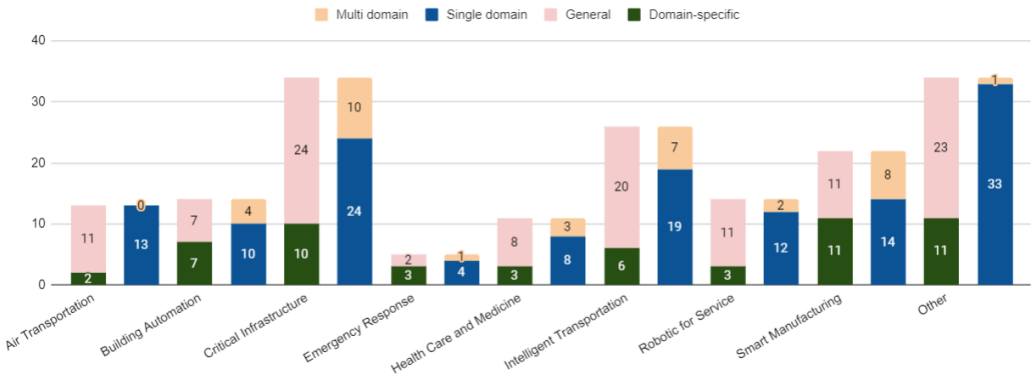

Figure 12: Correlation of application domains and single/multi -domain studies

Figure 12 shows the number of primary studies by each domain indicating if the proposed approach is domain-specific or general and if it is single or multi-domain. While only 16 papers (10.4%) are addressing multi applications domains, the reset 137 papers (89.6%) focuses on a single application domain. We can see that highest number of studies address Critical Infrastructure (34 studies), followed by Intelligent Transportation (26 studies) and Smart Manufacturing (22 studies). Also, Smart Manufacturing has the

highest number of studies presenting domain-specific approaches (11 studies), while Critical Infrastructure has the highest number studies presenting a general approach (with 24 studies).

Of all the primary studies, 20 studies (13%) report more than one main application domain, see Table 8, and the rest (87%) report a single application domain. From the multi domain studies, only 5 studies address 3 different application domains and the other 17 studies address only 2 different application domains. The highest number multi-domain studies are related to Critical Infrastructure and Intelligent Transportation with 5 studies.

Table 8: The studies addressing more than one application domain (multi-domain studies)

Studies	**# Domains**	**Addressed application domains**
[77]	3	Building Automation; Health Care and Medicine; Smart Manufacturing
[32, 31]	3	Critical Infrastructure; Intelligent Transportation; Smart Manufacturing
[40]	3	Health Care and Medicine; Critical Infrastructure; Robotic for Service
[78]	2	Intelligent Transportation; Smart Manufacturing
[79, 80]	2	Critical Infrastructure; Smart Manufacturing
[81]	2	Health Care and Medicine; Critical Infrastructure
[82]	2	Building Automation; Emergency Response
[83, 84, 85]	2	Critical Infrastructure; Intelligent Transportation
[86, 87]	2	Building Automation; Smart Manufacturing
[88]	2	Intelligent Transportation; Robotic for Service
[41]	2	Critical Infrastructure; Other

Finally, by analyzing the correlation between application domains which are addressed and formalisms which are used in the studies, we could find interesting results. From the 8 main application domains, in almost all of them (except "Robotics for Service") the Petrinet formalism has been used. 19 studies have used Petrinet in total for the application domains, which is highest number of studies among other formalisms. The next formalism with the highest number of application domain covered and the highest number of studies is Hybrid Automata with 14 studies covering 6 application domains.

Considering the formalisms used in different application domains, out of total 18 formalisms discussed in Q1.3, the papers with Critical Infrastructure application domain have used 11 different formalisms in 26 papers. The next application domain is Intelligent Transportation with using 9 different formalisms and 15 papers. A more comprehensive analysis can be found in the data-set of this study.

7. Profiles of Stakeholders Involved in CPS Modeling (RQ4)

This section reports the results of the question **RQ4: What is the profile of the person performs CPS modeling?** We observed that only 50

(32.7%) of the reviewed papers explicitly report the involved actors or stakeholders. The reviewed papers vary in the number of reported stakeholders: it ranged between reporting one actor (24 studies, 15.7%), to reporting four actors in only one study [89], as shown in Figure 13.

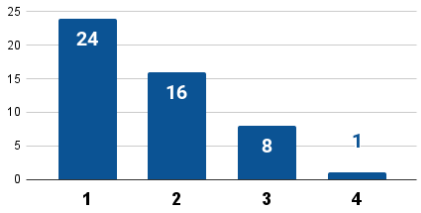

Figure 13: Counts of reported actors per study

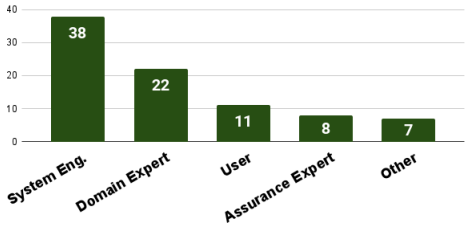

Figure 14: Counts of actors in all papers

Table 9: Descriptions of actor roles

Actor	**Activities**
System Engineer	Development, design, modeling, and analysis of the system.
Domain Expert	Requirements specification, ontology building, consulting, prototyping, and domain-specific analysis/research.
User	Application of the developed engineering tools and/or utilization of the CPS.
Assurance Expert	Evaluation, testing, and other assurance of the system.
Other	Other responsibilities such as management, policy-making, deployment, etc.

The actors reported in the reviewed papers are categorized into the five roles based on the activities for which they are responsible, described in Table 9. Note that actors define the roles, not the specific individuals. Hence, in a study, one person can play multiple actor roles, and one role can be played by multiple people. As shown in Figure 14, the System Engineer is the most reported participant (38 papers, 24.8%) followed by a Domain Expert (22 papers, 14.4%), a User (11 papers, 7.2%), an Assurance Expert (8 papers, 5.2%), and others (7 papers, 4.6%). The most common combination of the roles was a System Engineer and Domain Expert, found in 14 papers (9.2%). The single study with four actors [89] describes how a systems engineer, a design tool user, a domain expert, and an assurance expert can collaborate on electrical models, source code, and spreadsheets using a specialized IDE.

In comparison to RQ 2.3, we noticed that papers with a process description are twice as likely to report an actor (36 out of 86, 41.9%, vs. 14 out of 67, 20.9%), which was expected. Further, we noticed that the papers with a step-by-step process are much more likely to have a any actor mentioned (21 out of 32, 65.6%) than the other types of process descriptions (15 out of 54, 27.8%) or when no process description is given (14 out of 67, 20.9%). Similarly, papers with a step-by-step process are much more likely to report

a system engineer (16 out of 33, 48.5%) than papers with the other types of process (14 out of 69, 20.3%) or no process presented (10 out of 70, 14.3%). This leads us to an interpretation that step-by-step process descriptions are more likely to explicate the actors — and these actors are likely to be system engineers, perhaps because the engineering processes are led by them.

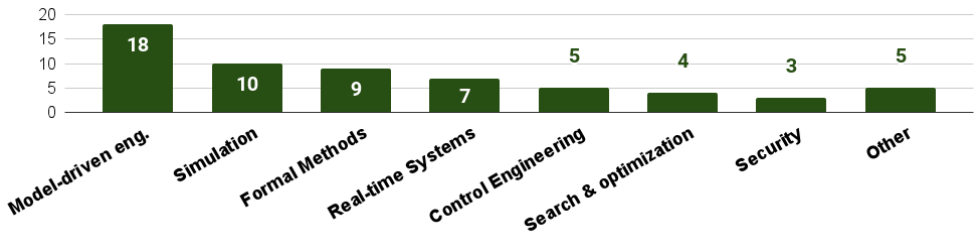

Figure 15: Counts of modelers' background knowledge found in the papers

7.1. Modeler's Background Knowledge

We note that only 33 (21.6%) of all reviewed papers explicitly reported the knowledge requires from the system modeler (note that multiple different roles may be involved in the modeling process). We organized the knowledge requirements into clusters shown in Figure 15. The most common background, reported in 18 papers (11.8%), is *Model-driven Engineering*, which refers to the skills of object-oriented, component-based, and meta-modeling with languages such as UML, AADL, etc.). The second and third most common backgrounds are *Simulation* (10 papers, 6.5%) and *Formal Methods* (9 papers, 5.9%). Among the papers that report any backgrounds, it is frequent to report more than one: 21 (13.7%) studies report at least two backgrounds, and 8 (5.2%) studies report three backgrounds, showing that it is common to require multidisciplinary skills for CPS modeling — at least there is a discussion of the background to begin with.

The backgrounds were reported primarily in the papers with models for use cases (RQ2.1): over half of background reports in papers with reported models (32 out of 60, 53.3%) were reported in papers that only have a model for a use case. The distribution of backgrounds in such papers stayed similar to Figure 15, except that the Model-driven Engineering background was more spread out across other types of models. This observation suggests that the Model-driven Engineering background is currently most suitable for modeling the approaches themselves (not just the use cases) as well as meta-modeling.

7.2. Research Fields of the Authors

For each paper, the reviewers analyzed a set of the research fields of the authors, based on their public profiles: academic websites, Google Scholar

pages, and public biographies. The clusters of research fields are described in Table 10. Their occurrences in papers are shown in Figure 16.

Table 10: Descriptions of author research fields

Field	Description
CS/Inf/SE	Computer Science, Informatics, Software Engineering, and other fields of computing.
EE/CE/SE	Electrical Engineering, Control/Computer Engineering, and Systems Engineering.
Other Eng.	Engineering fields *other than* electrical, computer, and mechanical. Examples: civil, aerospace, and chemical.
Mech Eng.	Mechanical Engineering
Math/Stat.	Mathematics and Statistics.
Business	Management, Administration, Business, and Operations.

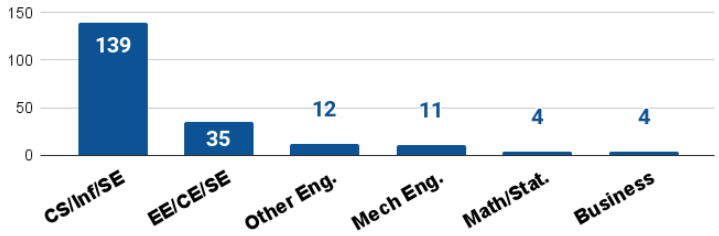

Figure 16: Counts of research fields of the authors in the reviewed papers

As expected, our selection of papers was dominated by Computer Scientists: 139 studies (91.5%) had an author with a computing background, as expected based on our search query. The only other common field was Electrical/Systems Engineering, with 35 (22.7%) papers having one such author. Two fields were found in 41 (26.8%), three fields were found in 4 papers (2.6%), and only one paper [39] had four fields: modeling of monitoring facilities for nuclear non-proliferation was investigated by the authors from Computer Science, Electrical Engineering, Civil Engineering, and Business. The most co-occurring combination of fields was Computer Science and Electrical Engineering, found in 30 papers (19.6%), and most of the remaining cross-field collaborations were combinations of Computer Science and various other engineering fields. This leads us to a sobering observation: while the methods, languages, tools, and skills for CPS modeling come from different technical areas, we did not observe MPM to be a truly *multi-disciplinary* area of research (as least when surveyed from the software engineering perspective).

Checking how the author fields relate to the formalisms (RQ1.3), we noted that Computer Science authors are highly prevalent and contribute to all formalists except one — Binary Decision Diagrams (surprisingly so). Several formalisms are found in papers exclusively from Computer Science: Temporal Logics, Timed Automata, Attribute Grammars, Process Algebras, and

Markov Chains. On the other hand, the most interdisciplinary formalisms, "covering" 5 out of the 6 considered fields, are Discrete Events, Petri Nets, and classical Automata/Finite State Machines.

One might expect that papers with multiple application domains from RQ3 would be more interdisciplinary. However, this expectation was not confirmed in our data: there was no significant association between having multiple domains and having multiple author fields. All domains had a high prevalence of Computer Science authors. Smart Manufacturing was the domain with the most non-Computer Science fields, and the most associated with Electrical Engineering (10 out of 22, 45.5%). Our interpretation is that applying multi-paradigm modeling to multiple domains does not, by itself, lead to interdisciplinary research (at least as reported from the software engineering perspective).

8. Quality assessment and demographics of classified studies

In this section, we report results regarding the quality assessment of classified studies. The results are detailed in Table 12. To characterize the first quality assessment criteria *(QA1)* (ranking of journal or conference), we decided to use the *CORE2018*[90] and *SCImago*[91] ranks lists. Venues which were not registered at the portal were marked as '0-Unranked', where the highest number of papers were published (45.10% comprised of 69 papers), which was expected since we surveyed recent research topics. On the other hand, 28.10% of the papers are ranked as B or C and 27% are highly ranked as A or A+, which indicates that 55.10% of primary studies we reviewed were published in ranked venues. We interpret that one reason for having a high number of not ranked venues is due to the fact that the area we are analysing is emerging and there are a lot of new publishing venues which will be ranked only in the times to come.

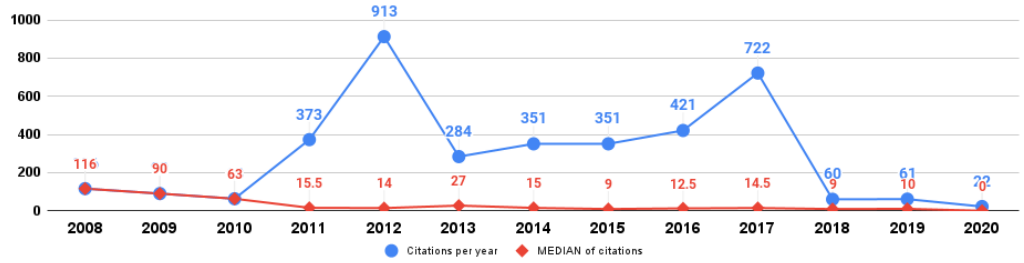

Figure 17: Number of citations per year

Next, we present results related to the relevance of the papers based on number of citations *(QA2)*. We used the number of citations obtained from

Google Scholar[92] (in June 2021). Figure 17 shows the number of citations per year and their median.We note that only 11% of the reviewed papers are never cited, of which most of them were published in last two years. The 3 most cited papers are included in Table 11. The most cited papers happen to be the papers that were published earlier in journals, while as expected for the newest papers there are not yet many citations.

Table 11: Top cited papers

Paper	Year	Title	Venue Type	Citation N
[55]	2012	Adaptive numerical algorithms in space weather modeling	Journal	498
[40]	2012	Ensuring Safety, Security, and Sustainability of Mission-Critical Cyber-Physical Systems	Journal	302
[71]	2011	System-of-systems approach for interdependent critical infrastructures	Journal	286

Furthermore, we analyzed the quality of the content of the studies. More than two-thirds of the studies clearly and precisely describe the problem and 20.26% of them describe the problem vaguely *(QA3)*. This result indicates that reviewers found that almost all primary studies present the motivation for their approach clearly, specific to the problem which the study tries to address. Regarding the clarity of the research context *(QA4)*, reviewers found that a negligible number of studies have vaguely focused on the research context (only 4 papers), while a majority of papers described the research context with references reporting the problems and advantages of the related work. Almost two thirds of the studies proposed solutions which are evaluated by case studies, while 13% do not report any evaluation *(QA5)*. Only 7.19% of the studies are evaluated empirically, while the rest show the applicability of the proposal using illustrative examples. In order to make the modeling approaches better understood and accepted, more empirical studies need to be carried out. More than half of the studies (60.78%) explicitly presented their contribution *(QA6)*, meaning they contribute a concrete solution and explain the scope of their contribution clearly in the conclusions. A negligible number of papers do not present contributions (1.96%), while the remaining 37.25% of the papers describe their contributions generally. Finally, only 14.38% of the papers do not present any direction for future work *(QA7)*. For more than half of the papers (56.86%) directions for future work are presented in a generalized manner. While only (28.76%) presented future work directions with recommendations.

We can conclude that most of the primary studies provide motivation for their problem and provide a research context. About 77.78% approaches are evaluated by a case study or based on empirical foundations. Although in most of cases the contributions are explicit, we note that the future work is

Table 12: Quality assessment results

Question	Scores	Results	
QA1: What is the ranking of the paper according to the conference/journal where it was published?	62.5	1 = Highly ranked	27%
		0.5 = Ranked	28.10%
		0 = Unranked	45.10%
QA3: How clearly is the problem of study described?	137.5	1 = Explicitly	79.74%
		0.5 = Vaguely	20.26%
QA4: How clearly is the research context stated?	119.5	1 = With references	58.82%
		0.5 = Generally	38.56%
		0 = Vaguely	2.61%
QA5: How rigorously is the method evaluated?	86.9	1 = Empirical foundation	7.19%
		0.66 = Case study	70.59%
		0.33 = Lessons Learned	9.15%
		0 = No evaluation	13.07%
QA6: How explicitly are the contributions presented?	121.5	1 = Explicitly	60.78%
		0.5 = Generally	37.25%
		0 = No presentation	1.96%
QA7: How explicitly are the insights and issues for future work stated?	87.5	1 = With recommendations	28.76%
		0.5 = Generally	56.86%
		0 = No statement	14.38%

either very general or lacking and rarely reported with a concrete road-map.

Table 13: Reviewers self-assessment

Question	Scores	Results	
SA1: Reviewer's confidence about content of the study	100.5	1 = Very confident	31.37%
		0.5 = Confident	68.63%
SA2: Reviewer's confidence about quality of the study	100	1 = Very confident	30.72%
		0.5 = Confident	69.28%

Regarding the confidence of reviewers about the content and quality of the study, confident responses were dominant for both as shown in Table 13, likely due to the reviewers' experience in CPS modeling.

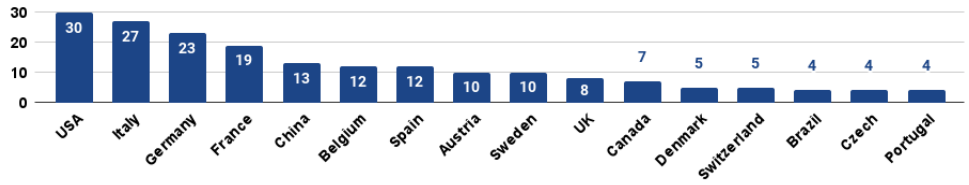

Figure 18: Contributing countries (based on author affiliations)

Further, we analyzed the countries which publish the analyzed set of the primary studies, from a total of different 38 countries Figure 18 highlights countries with at least four publications. We can observe that the United States (16%), Italy (14%), Germany (11.5%), France (10%), and China (7%) are the top five countries. Findings show that 101 (66%) of the papers were written by author(s) affiliated to one country, 41 papers (26.80%) were written jointly by authors affiliated to two countries, 5 papers (3.27%) were

written by authors affiliated to three countries. Finally, 3,2, and 1 primary studies were written by authors affiliated to four, five, and seven countries respectively.

Table 14: Top contributing institutions

Institution	**Country**	**Count**
University of Antwerp	Belgium	11
Flanders Make	Belgium	7
Politecnico di Milano	Italy	7
Universit di Napoli Federico II	Italy	6
Seconda Universita di Napoli	Italy	6
Guangdong University of Technology	China	5
University of California	USA	5
Mlardalen University	Sweden	5
East China Normal University	China	4
Aarhus University	Denmark	4
Newcastle University	UK	4
McGill University	Canada	4
University of Cantabria	Spain	4
INRIA	France	4
Royal Institute of Technology (KTH)	Sweden	4
University of Pennsylvania	USA	4
University of Toulouse	France	4

Table 14 depicts the top contributing institutions with at least four studies out of a total of 206. The top three institutes are the University of Antwerp (with 7% of the studies), Flanders Make and Politecnico di Milano (4.5%) each, Universit di Napoli Federico II and Seconda Universita di Napoli (4%) each.

Table 15: Top publication venues

Venues	**Venue Type**	**Count**
Software & Systems Modeling	Journal	15
IFAC-PapersOnLine	Journal	6
CEUR Workshop Proceedings	Workshop	5
ACM/IEEE International Conference on Model Driven Engineering Languages and Systems	Conference	4
Microprocessors and Microsystems	Journal	4
Advanced Engineering Informatics	Journal	3
Leveraging Applications of Formal Methods, Verification and Validation	Conference	3

Finally, we analyzed to which venues are MPM approaches for CPS getting publish. Journal publications made up 74 (48.3%) of the primary studies, while conferences, book chapters, and workshop papers made up 60 (39.2%), 11 (7%), and 8 (5%) of the studies, respectively. The ranking of the top venues with at least three studies is shown in Table 15, of which convincingly most papers were published in Software & Systems Modeling Journal.

9. Summary of Results

Details about analyzed and filtered data associated with our results can be found in our complementary submission [16] associated with online data repository[6]. We observe that the number of studies reporting multi-paradigm modeling of CPS is increasing constantly until the year of 2017, from which we can note the drop off on the number of the studies. Not to jump to conclusions, this drop off can be impacted due to the two search strategies which we used, of which for the manual source we had only data till the year 2017 (see Figure 19).

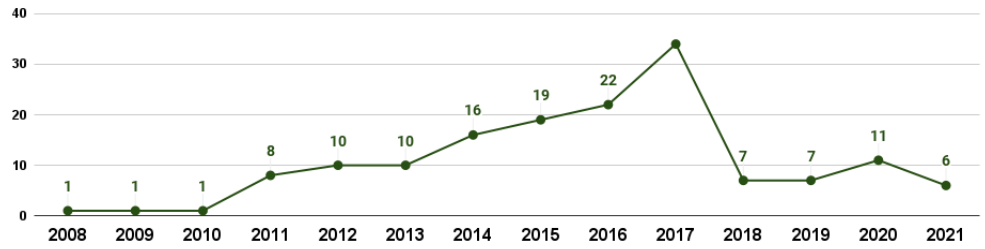

Figure 19: Modeling approaches for CPS over time

Regarding the quality of the analyzed studies we noted that the primary studies are clearly reporting the problem, research context and contributions, while the evaluation of the approaches (especially experimental one) and the road map for future work is often lacking or being very general. Regarding the demographic distribution, we can note that most publications on this research topic actually are ones from EU countries (Italy, Germany, France etc.), followed by USA, China and Brazil, having the top contributing institutions from the EU.

The results discussed in Section 4 **(RQ1)** show a widespread use of MPM for designing and developing CPS. Most of the approaches were found to only support a fraction of the development process phases, with model editing, simulation, transformation, and analysis being the most frequent. Conversely, there are fewer studies reporting verification, integration and code generation. Automated code generation from models has been found to produce code with fewer errors [93] and can improve developer efficiency on average by 20-30 percent [23], which should encourage developing more tools to support this activity. In addition, some papers describe purely theoretical approaches and reported no tool support.

[6] https://data.mendeley.com/datasets/jy6ww3hmyw

The studies reported using a wide range of formalisms, of which the most common were Petri Nets, Hybrid Automata and Discrete Events. However, the number of studies explicitly reporting the use of more then one formalism is relatively low. In general, modeling techniques seem to be effective when addressing a single stage of the development process in isolation, and when targeting mostly one or two parts of system (software, hardware, network, environment). In our opinion, these facts are symptoms of the intricacy of CPS development: in fact, it is very difficult to provide a single modeling language and corresponding tool able to address the whole set of development phases, or to effectively model all the parts of a CPS. In this respect, the research community has investigated multiple solutions to integrate different levels of abstraction, different CPS views, and different CPS parts. However, many approaches do not either state or use any integration mechanism, indicating that this topic remain challenging. When stated, automated integration via model transformations is the most frequent, while others leverage several forms of trace links that are exploited to check the consistency between the different integrated parts and/or to derive the synthesized system. We note that only very few studies make use of more than one integration mechanisms, which for realistic industrial development settings would likely not be the case.

As reported in Section 5, most MPM approaches for CPS **(RQ2)** focus on high-level aspects of the system, and a number of them provide models of development artifacts and the cyber parts of the system. Aside from engineering support, it is most common for the studies to analyze or promote automation of development tasks, efficiency, and resilience. There is a relative shortage of research tackling complexity or simulation of CPS.

Most studies focus on instance models, whereas less than a quarter of them use meta-modeling. Approximately one quarter of the studies report developing a custom tool for their approach, whereas over half of the studies use the existing tools and modeling languages. UML is convincingly the most used modeling language, followed by SysML, Marte and AADL, while Simulink, EMF, and UPPAAL are reported to be the most used tools (frameworks). About half of the works mentioning languages include more than one language, and a third of those mentioning tools include more than one tool. These results testify the multi-faceted approach for developing CPS conveyed by MPM. Interestingly, the relationships between formalisms, languages, and tools, are in general made less explicit; one reason for this might be the tight correlation between formalisms and language/tools, such that the users think of them as a single method rather than the combination of different ingredients. Eventually, most of the primary studies do not provide an engineering process, and those that do predominately describe it in an

informal, textual way. This result indicates a lack of research in modeling a process for development of CPS.

As reported in Section 6 **(RQ3)**, about 33.3% of the studies present domain-specific approaches, while the majority propose modeling approaches for any CPS. However, there is a tendency to use only a single domain to validate the approaches, as only 10.4% of studies report more then one application domain being addressed. This can be potential threat to applicability of approaches addressing any CPS in some other domain. The highest number of modeling approaches address the Critical Infrastructure, Intelligent Transportation and Smart Manufacturing application domains, and this trend is the same for MPM approaches. On another hand, there are only a few studies reporting their application for Emergency Response. Also, it worth to note Petrinet formalism is used the most among all application domains (with 19 papers) and Critical Infrastructure uses the highest number of formalism in different studies (11 formalisms).

Finally, the results presented for **RQ4** in Section 7, indicate that a sizeable minority of MPM papers report actors, with the *System Engineer* and *Domain Expert* as the most common ones. A small minority of the papers report the expertise required for modeling, and most of it is for use-case-specific modeling requiring *Model-driven Engineering* skills. A more detailed understanding of how the actor roles and necessary skills relate to MPM approaches could suggest ways to involve more diverse stakeholders in the modeling process, as a result, enhance the efficiency and effectiveness of those approaches [94]. A vast majority of the papers come from authors in computing fields, as expected, and a minority from *Electrical Engineering* and related fields. We found no dependency between the fields of the authors and the domains, suggesting that interdisciplinary MPM research does not necessarily arise from applications to multiple domains.

10. Threats to Validity

In this section, we discuss the main threats to validity in our study and how they were mitigated during the MPM4CPS SMS process.

Construct validity. A main issue in any literature review study is the possibility of missing some relevant primary studies, which may be caused by incomplete or incorrect research strings, inappropriately-worded research questions, inappropriate inclusion/exclusion criteria and incomplete venues or databases. In our study, these issues were mitigated using three techniques (query testing, validation survey with a COST action members and pilot session with reviewers) to validate the proposed protocol The inclusion/exclusion of papers was performed by different reviewers, who may have

different interpretation of what is modeling of CPS. This was mitigated by the use of a standard definition of the all key terms. Also, we adopted a multiple-review approach where at least two reviewers decided on inclusion/exclusion of the initial set of primary studies. In case of disagreement, the third reviewer was included to build consensus.

Internal validity. Bias in data extraction or subjective interpretation about the extracted data from the primary studies may be potential threats to the validity in the conducting phase, potentially leading to incorrect classification of studies. We note that each paper was assigned to a one reviewer, then according to his self-confidence report, the paper may have been revised by a second reviewer. Nevertheless, most papers were reviewed only once, which increases the risk of biases in the resulting classification. However, we had 17 researchers from different countries active in area of modeling of CPS included in classification process, as well as in iterative analysis and filtering of results, therefore we believe that we managed to unify the interpretations.

For instance, the reviewers' subjective interpretation of how many models/modeling methods are used in a study, and what they are, may also have influenced the results for questions *RQ1* and *RQ2*. To minimize the influence of the reviewer preconceptions on extracting the modeling data, we performed ground-up analysis where the cluster tags were assigned as specifically as possible, and then merged into larger categories (such as "state machine-like models"). During the analysis phase, the unsatisfactory data synthesis may lower the quality of the data analysis. In our case, papers with poor-quality or incomplete extracted data (missing answers, unclear descriptions, etc.) were re-discussed between reviewers until a consensus emerged.

External validity. Finally, given the repetitive retrieval of our data sets, we noticed that the database search results may differ along the time, and that the search strategies change, for instance the number of the keywords gets reduced, asking for slicing the search string and retrieving a several different sets. This may result in obtaining a slightly different data sets.

Also, we used two methods to perform our search, the first one which was automatic based on search string, and the second one which was based on data sets manually retrieved from another mapping study having identified modeling approaches for CPS. In a first run of our study in 2017, we obtained primary studies from both sources, however, in the second run in 2021, we obtained just primary studies from the automatic data set. This can impact the generalizability of our results, like assumed drop of this type of studies from 2017.

However, the research on CPS, and even the term itself, is relatively recent. Note that the rise in CPS studies began approximately in 2011,

thus leading to an appropriate coverage period of our SMS. Furthermore, our focus on the software engineering studies may limit the generalizability of our conclusions to other broad communities (e.g., control or mechanical engineering).

11. Related Work

Multiple literature reviews have been produced on different topics in software engineering, but so far none has been conducted that investigates MPM of CPS, nor do they study the background of authors or developers. An original contribution of our effort is that for the first time in this research field, we have followed an SMS method to be as objective as possible in our selection of primary studies as we mentioned before. In this section, we review the literature surveys that were excluded from our classification as secondary studies, and mention few other related studies, and compare them to our work.

In [95], the authors describe the state and directions of research on systematic CPS engineering. This paper defines CPS and gives their typical examples: electronic control units in a car and a sensor network for earthquake early warning. The authors considered model-based design as an appropriate paradigm for CPS engineering. They reviewed existing approaches used in different development phases: systems and requirements analysis, modeling, model transformation, and code generation. They underline the particular challenges which CPS pose to software development, such as its interaction with the physical environment and how it should be modeled and analyzed. This paper, however, focuses on describing CPS development phases instead of surveying the available literature.

Another category of papers focuses on specific CPS issues, such as [96], which reviews the use of time, clocks, and clock synchronization protocols in CPS. This survey details solutions used in designing and implementing CPS (e.g., oscillator technology, in GNSS, in network technology, and in time and frequency distribution protocols). Our paper, however, studies more general characteristics of CPS design.

Some papers focus on surveying the concepts and the applications of CPS such as [76]. In order to shed some light on the origins, the terminology, relatively similar concepts, and challenges in CPS, the authors presented a survey on related literature discussing practical applications and major research domains. Since CPS is an extensive research area, CPSs span diverse applications in different scales. Therefore, each application necessitates strong reasoning capabilities with respect to unique system-level requirements/challenges, the integration of cutting-edge technologies into the

related application, and practical impact on the real world. The authors conclude that existing legacy systems have limited awareness of the CPS requirements, and that revolutionary design approaches are necessary to achieve the overall system objectives. Their paper examines some of the same research questions as this paper in types of domains used and qualities of CPS, but not others such as formalisms, tools, or actors in CPS design.

In [97], the focus is on the challenges of modeling CPSs that arise from the heterogeneity, concurrency, and sensitivity to timing of such systems. A model of a CPS comprises models of physical processes as well as models of the software, computation platforms, and networks. One of the main challenges is to keep model components consistent and to check for correctness of connections between components. As the model grows, the possibility of error also grows. In this paper, the authors identified three types of errors: (1) unit errors, (2) semantic errors, and (3) transposition errors. They also analyzed the state of the art in existing tools and methods. This paper also focuses on challenges and their solutions, and examines a few approaches instead of performing a literature review like this paper.

As for [98], it discusses the importance of design, modeling, simulation and integration of CPS and focus on methods and applications. The authors emphasize that designing CPS requires a multi-disciplinary development process during which designers should focus on integration and interaction of physical and computational components. Furthermore, they presented case studies and current best practice from industry. This paper also focuses on CPS design aspects while our work

The above papers also lack the quantitative analysis about the distribution of papers covering each domain and CPS quality that we perform in this paper.

Another CPS security study [99] identifies, classifies, and analyzes existing research on CPS security in order to better understand how security is actually addressed in CPS. The authors empirically define a comparison framework for classifying methods or techniques for CPS security. From the collected data authors observe that even if solutions for CPS security have emerged only recently, in the last few years, they have been gaining a sharply increasing scientific interest across heterogeneous publication venues. The systematic map of research on CPS security provided here is based on, for instance, application fields, various system components, related algorithms and models, attacks characteristics, and defense strategies. While they use a similar literature search/review methodology, study the institutions of the authors of studies, and perform quantitative analysis on surveyed papers, our work studies different research questions instead of focusing only on security.

A paper of CPS application domains [100] performs a systematic review

over EU projects in period 2007–2016 to find the most targeted domains and technical challenges by industry. The focus was on self-adaptive CPS with collective behavior, which are presented in the demonstrators of the projects. Similarly to our study, the preliminary results of the study present the domains of the demonstrators. However, the paper does not take into account the intersections between the domains in the applications nor the domain specificity of the used modeling approaches. It only maps the system properties that addresses the targeted challenges to the domains.

There is also a recent systematic study related to applying modeling languages in Industry 4.0 [101]. The authors aim to assess the use of modeling techniques by research and contribution type facets and identifying countries and popular venues. They conclude that there is a lack of experimentation and relatively little research on validation and verification, similar as in our study. Also, they find that the UML is used most often, as well as other domain-specific languages, which in our study are listed in detail. Also they identify integrations as a major challenge, which seems to be shifting toward information management and process modeling. Similar to our insights, they conclude that this area demands further research. Compared with the papers in this section, our survey includes a greater breadth of research questions including on the author backgrounds that others lack.

12. Conclusion and Future Work

This paper presents a systematic mapping study on multi-paradigm modeling for cyber-physical systems. A total of 560 primary studies have been reviewed by 17 researchers from different academic institutions and companies. The presented results were obtained from the analysis of 386 relevant primary studies, and can help researchers and practitioners to orient themselves to address open challenges, and as guide toward specific solutions for their problems.

We observed that most of the approaches report the use of models or meta-models and modeling tools and languages to manage the complexity of CPSs across many domains. Moreover, the fact that half of the researchers use existing tools/languages as well as continue to develop their own ones shows that the range of modeling languages and tools supporting the modeling of CPSs is not yet fully saturated. We also observe that the majority of modeling approaches offered for CPSs only cover a few parts of the system being developed and only consider a few system properties — and mostly come from the computing fields of research. In realistic industrial settings, where all very heterogeneous parts of large complex systems must be covered (for example the air frame, electrical cables, mechanical systems, embedded

systems, etc. of an airplane), it becomes much easier to reuse well-proven formalisms languages and tools already mastered by engineers than to develop a new single formalism / language / tool capable of covering all these aspects of CPSs. In addition, the diversity of industry and even project-specific needs and large number of approaches makes it impractical to develop a new formalism for each approach. We are then left with the issue of integrating heterogeneous models and tools.

Among the approaches that use several existing languages and tools, a large number of approaches do not state or consider model integration. Among the studies that consider this aspect, a majority rely on batch model transformations for which the generated model has to be completely regenerated every time the source model(s) is (are) changed. When the generated model has to be edited manually, a major challenge is to keep the source model(s) consistent with those changes. This issue is even more critical for the most general case of concurrent engineering, for which the models are modified in parallel by the teams of many different engineering disciplines. Our survey revealed that this challenge is currently not addressed well, despite a few studies presenting model synchronization techniques working at the syntactic level and others at the semantics level by composing models of computation.

Another future work direction when modeling a CPS is to model the development process with appropriate language(s), so that this information is also made explicit and can be processed by tools to better support the orchestration of the development phases. While less than half of the studies report on the process, those that do it only described processes using natural language and therefore lack the precision and formality that can be obtained by modeling. Similar to the models describing the system being developed, which are always expressed with adequate formal language, CPS development processes would also benefit from such formalization.

Our final finding is that there is a lack of reporting on the roles and expertise of CPS stakeholders and modelers. Reporting such information explicitly would be helpful to create educational programs tailored for CPS engineers. One way to address this issue could be to extend our study to industrial case studies and interview the CPS stakeholders. This future study would also allow us to cover industrial development setups and check the validity of our results.

Acknowledgment

This work was supported by MPM4CPS IC1404 COST action and partially supported by NOVA LINCS with the support of FCT Portuguese

national funds (UID/CEC/04516/2019) and the Knowledge Foundation in Sweden through the MINEStrA (nr. 20170133) and SACSys (nr. 20190021) projects.

We would like to thank Prof. Paulo Carreira, Prof. Geylani Kardas, Prof. Ivan Lukovic, and Prof. Mauro Iacono for their feedback during the construction of the protocol. We thank Prof. Vasco Amaral, Prof. Miguel Goulao, Prof. Bedir Tekinerdogan, Dr. Claudio Gomes, and Ferhat Erata for their contribution to the classification of the primary studies. Finally, we thank Prof. Eugene Syriani and Brice Bigendako for their continual support with the ReLis platform.

References

[1] Y. Z. Lun], A. DInnocenzo, F. Smarra, I. Malavolta, M. D. D. Benedetto], State of the art of cyber-physical systems security: An automatic control perspective, Journal of Systems and Software 149 (2019) 174 – 216. doi:https://doi.org/10.1016/j.jss.2018.12.006.

[2] P. Derler, E. A. Lee, A. Sangiovanni Vincentelli, Modeling cyber-physical systems, Proceedings of the IEEE 100 (1) (2012) 13–28. doi:10.1109/JPROC.2011.2160929.

[3] E. A. Lee, The past, present and future of cyber-physical systems: a focus on models., Sensors (Basel, Switzerland) 15 (3) (2015) 4837–69. doi:10.3390/s150304837.

[4] T. Kuhne, Matters of (Meta-) Modeling, Software & Systems Modeling 5 (4) (2006) 369–385. doi:10.1007/s10270-006-0017-9.

[5] D. Broman, E. A. Lee, S. Tripakis, M. Törngren, Viewpoints, Formalisms, Languages, and Tools for Cyber-physical Systems, in: Proceedings of the 6th International Workshop on Multi-Paradigm Modeling, MPM '12, ACM, New York, NY, USA, 2012, pp. 49–54. doi:10.1145/2508443.2508452.

[6] A. G. Kleppe, J. Warmer, J. B. Warmer, W. Bast, MDA explained: the model driven architecture: practice and promise, Addison-Wesley Professional, 2003.

[7] H. Vangheluwe, J. De Lara, P. Mosterman, An introduction to multi-paradigm modelling and simulation., in: F. Giambiasi, N. Barros (Eds.), 2002 Conference on AI, Simulation and Planning in High Autonomy Systems (AIS'2002), Lisboa, Portugal, 2002, pp. 9–20.

[8] H. L. Vangheluwe, G. C. Vansteenkiste, E. J. H. Kerckhoffs, Simulation for the Future: Progress of the Esprit Basic Research Working Group 8467, Tech. rep.

[9] M. Amrani, D. Blouin, R. Heinrich, A. Rensink, H. Vangheluwe, A. Wortmann, Towards a formal specification of multi-paradigm modelling, in: Proceedings - 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion, MODELS-C 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 419–424. doi:10.1109/MODELS-C.2019.00067.

[10] M. Amrani, D. Blouin, R. Heinrich, A. Rensink, H. Vangheluwe, A. Wortmann, Multi-paradigm modelling for cyberphysical systems: a descriptive framework, Software and Systems Modeling 20 (3) (2021) 611–639. doi:10.1007/s10270-021-00876-z.

[11] B. Hailpern, Multiparadigm Languages, IEEE Software 3 (1 1986). doi:10.1109/MS.1986.232426.

[12] B. Kitchenham, Guidelines for performing Systematic Literature Reviews in Software Engineering, Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report (7 2007). doi:10.1016/j.scico.2013.03.017.

[13] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering–a systematic literature review, Information and software technology 51 (1) (2009) 7–15. doi:10.1016/j.infsof.2008.09.009.

[14] ICT COST Action IC1404, Multi-Paradigm Modeling for Cyber-Physical Systems (MPM4CPS), http://mpm4cps.eu/, accessed: June 2020.

[15] A. Barišić, D. Savić, R. Al Ali, I. Ruchkin, D. Blouin, A. Cicchetti, R. Eslampanah, O. Nikiforova, M. Abshir, M. Challenger, C. Gomes, B. Tekinerdogan, F. Erata, V. Amaral, M. Gouão, Systematic Literature Review on Multi-Paradigm Modelling for Cyber-Physical Systems, Tech. Rep. WG4.4, COST Action IC1404 MPM4CPS (2019). doi:10.5281/zenodo.2528953.

[16] A. Barišić, A. Cicchetti, I. Ruchkin, D. Blouin, Literature classification data for a systematic mapping study on multi-paradigm modeling for cyber-physical systems, Data In Brief Journal.

[17] M. Petticrew, H. Roberts, Systematic reviews in the social sciences: A practical guide, John Wiley & Sons, 2008.

[18] A. Barišić, STSM Report: Systematic literature review on multi-paradigm modeling for CPS Systems (5 2018). doi:10.5281/zenodo.1454531.

[19] A. Barišić, J. Cunha, Sustainability in Modelling of Cyber-Physical Systems: A Systematic Literature Review - Intermediate Technical Report (4 2017). doi:10.5281/ZENODO.1481946.

[20] B. Bigendako., E. Syriani, Revue Litteraire Systématique (ReLiS), http://relis.iro.umontreal.ca/, accessed: June 2020.

[21] B. M. Bigendako, E. Syriani, Modeling a Tool for Conducting Systematic Reviews Iteratively., in: MODELSWARD, 2018, pp. 552–559.

[22] B. Selic, The pragmatics of model-driven development, IEEE Software 20 (5) (2003) 19–25. doi:10.1109/MS.2003.1231146.

[23] J. Whittle, J. Hutchinson, M. Rouncefield, The State of Practice in Model-Driven Engineering, IEEE Software 31 (3) (2014) 79–85. doi:10.1109/MS.2013.65.

[24] R. E. Bloomfield, P. Popov, K. Salako, V. Stankovic, D. Wright, Preliminary interdependency analysis: An approach to support critical-infrastructure risk-assessment, Reliability Engineering & System Safety 167 (2017) 198–217. doi:https://doi.org/10.1016/j.ress.2017.05.030.

[25] I. Dávid, A multi-paradigm modeling foundation for collaborative multi-view model/system development, in: CEUR Workshop Proceedings, Vol. 1775, SRC@MoDELS, 2016.

[26] F. Cicirellia, G. Fortino, A. Guerrieri, G. Spezzano, A. Vinci, Metamodeling of Smart Environments: from design to implementation, Advanced Engineering Informatics 33 (2017) 274–284. doi:10.1016/J.AEI.2016.11.005.

[27] M. Chadli, J. H. Kim, K. G. Larsen, A. Legay, S. Naujokat, B. Steffen, L. M. Traonouez, High-level frameworks for the specification and verification of scheduling problems, International Journal on Software Tools for Technology Transfer 20 (4) (2018) 397–422. doi:10.1007/s10009-017-0466-1.

[28] M. Bozzano, H. Bruintjes, A. Cimatti, J.-P. Katoen, T. Noll, S. Tonetta, Formal Methods for Aerospace Systems, in: Cyber-Physical System Design from an Architecture Analysis Viewpoint, Springer Singapore, Singapore, 2017, pp. 133–159. doi:10.1007/978-981-10-4436-6{_}6.

[29] J. A. E. Isasa, P. G. Larsen, F. O. Hansen, Energy-aware model-driven development of a wearable healthcare device, in: SEHC 2014, FHIES 2014: Software Engineering in Health Care, Vol. 9062 LNCS, Springer, Cham, 2017, pp. 44–63. doi:10.1007/978-3-319-63194-3{_}4.

[30] Technische Universitat Wien, Institute of Electrical and Electronics Engineers, IEEE Industrial Electronics Society, Towards Model-Based Performability Evaluation of Production Systems.

[31] P. Araújo-de Oliveira, F. Durán, E. Pimentel, A procedural and flexible approach for specification, modeling, definition, and analysis for self-adaptive systems, Software - Practice and Experience (2021). doi:10.1002/spe.2962.

[32] M. Challenger, B. T. Tezel, V. Amaral, M. Goulão, G. Kardas, Agent-based cyber-physical system development with SEA_ML++, in: Multi-Paradigm Modelling Approaches for Cyber-Physical Systems, Elsevier, 2021, pp. 195–219. doi:10.1016/b978-0-12-819105-7.00013-1.

[33] L. Zhang, View oriented approach to specify and model aerospace cyber-physical systems, in: Proceedings - 2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing, DASC 2013, IEEE, Chengdu, China, 2013, pp. 296–303. doi:10.1109/DASC.2013.78.

[34] J. Denil, P. De Meulenaere, S. Demeyer, H. Vangheluwe, DEVS for AUTOSAR-based system deployment modeling and simulation, Simulation 93 (6) (2017) 489–513. doi:10.1177/0037549716684552.

[35] H. Heinze, K. Kallow, H. Lackner, S. Sadeghipour, H. Schlingloff, S. Tahirbegovic, H.-W. Wiesbrock, Application and Evaluation in the Healthcare Domain, in: Model-Based Engineering of Embedded Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 215–230. doi:10.1007/978-3-642-34614-9{_}15.

[36] J. Sztipanovits, T. Bapty, Z. Lattmann, S. Neema, Composition and Compositionality in CPS, in: Handbook of System Safety and Se-

curity, Syngress, Boston, 2017, Ch. Chapter 2, pp. 15–38. doi:10.1016/B978-0-12-803773-7.00002-4.

[37] L. Apvrille, Y. Roudier, Designing safe and secure embedded and cyber-physical systems with SysML-Sec, in: MODELSWARD 2015: Model-Driven Engineering and Software Development, Vol. 580, Springer, Cham, 2015, pp. 293–308. doi:10.1007/978-3-319-27869-8{_}17.

[38] P. i Casas, A. i Casas, Using Specification and Description Language for Life Cycle Assesment in buildings, Sustainability (Switzerland) 9 (6) (2017). doi:10.3390/su9061004.

[39] H. E. Garcia, T. L. Burr, G. A. Coles, T. A. Edmunds, A. J. Garrett, M. B. Gorensek, L. L. Hamm, J. F. Krebs, R. L. Kress, V. E. Lamberti, D. A. Schoenwald, C. P. Tzanos, R. C. Ward, Integration of facility modeling capabilities for nuclear nonproliferation analysis, Progress in Nuclear Energy 54 (1) (2012) 96–111. doi:10.1016/j.pnucene.2011.07.007.

[40] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, S. K. S. Gupta, Ensuring Safety, Security, and Sustainability of Mission-Critical CyberPhysical Systems, Proceedings of the IEEE 100 (1) (2012) 283–299. doi:10.1109/JPROC.2011.2165689.

[41] T. Courtney, S. Gaonkar, K. Keefe, E. W. D. Rozier, W. H. Sanders, Mobius 2.3: An extensible tool for dependability, security, and performance evaluation of large and complex system models, in: 2009 IEEE/IFIP International Conference on Dependable Systems Networks, 2009, pp. 353–358. doi:10.1109/DSN.2009.5270318.

[42] C. Derksen, T. Linnenberg, R. Unland, A. Fay, Structure and classification of unified energy agents as a base for the systematic development of future energy grids, Engineering Applications of Artificial Intelligence 41 (2015) 310–324. doi:10.1016/J.ENGAPPAI.2014.10.005.

[43] P. G. Larsen, J. Fitzgerald, J. Woodcock, R. Nilsson, C. Gamble, S. Foster, Towards semantically integrated models and tools for cyber-physical systems design, in: ISoLA 2016: Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications, Vol. 9953 LNCS, Springer, Cham, 2016, pp. 171–186. doi:10.1007/978-3-319-47169-3{_}13.

[44] F. Moscato, V. Vittorini, F. Amato, A. Mazzeo, N. Mazzocca, Solution Workflows for Model-Based Analysis of Complex Systems, IEEE

Transactions on Automation Science and Engineering 9 (1) (2012) 83–95. doi:10.1109/TASE.2011.2161981.

[45] K. Vanherpen, J. Denil, H. Vangheluwe, P. De Meulenaere, Model transformations for round-trip engineering in control deployment co-design, Simulation Series 47 (8) (2015) 55–62.

[46] B. P. Zeigler, J. J. Nutaro, C. Seo, Combining DEVS and model-checking: Concepts and tools for integrating simulation and analysis, International Journal of Simulation and Process Modelling 12 (1) (2017) 2–15. doi:10.1504/IJSPM.2017.082781.

[47] E. Francalanza, J. Borg, C. Constantinescu, A knowledge-based tool for designing cyber physical production systems, Computers in Industry 84 (2017) 39–58. doi:10.1016/J.COMPIND.2016.08.001.

[48] A. Tolk, A. Harper, N. Mustafee, Hybrid models as transdisciplinary research enablers, European Journal of Operational Research 291 (3) (2021) 1075–1090. doi:10.1016/j.ejor.2020.10.010.

[49] B. J. Oakes, C. Gomes, F. R. Holzinger, M. Benedikt, J. Denil, H. Vangheluwe, Hint-Based Configuration of Co-simulations with Algebraic Loops, Tech. rep.

[50] C. Wang, L. Wan, T. Xiong, Y. Xie, S. Wang, A Relational Abstraction of Structure and Behavior for Cyber-Physical System Design, IEEE Access 9 (2021) 40388–40401. doi:10.1109/ACCESS.2021.3064249.

[51] R. Abdallah, A. Motii, N. Yakymets, A. Lanusse, Using model driven engineering to support multi-paradigms security analysis, Communications in Computer and Information Science 580 (2015) 278–292. doi:10.1007/978-3-319-27869-8{_}16.

[52] S. Balasubramaniyan, S. Srinivasan, F. Buonopane, B. Subathra, J. Vain, S. Ramaswamy, Design and verification of Cyber-Physical Systems using TrueTime, evolutionary optimization and UPPAAL, Microprocessors and Microsystems 42 (2016) 37–48. doi:10.1016/J.MICPRO.2015.12.006.

[53] J. Fitzgerald, K. Pierce, P. G. Larsen, Collaborative Development of Dependable Cyber-Physical Systems by Co-Modeling and Co-Simulation, in: Research on Embedded Systems Design, IGI Global, 2014, pp. 1–28. doi:10.4018/978-1-4666-6194-3.ch001.

[54] S. Marrone, R. J. Rodríguez, R. Nardone, F. Flammini, V. Vittorini, On synergies of cyber and physical security modelling in vulnerability assessment of railway systems, Computers & Electrical Engineering 47 (2015) 275–285. doi:https://doi.org/10.1016/j.compeleceng.2015.07.011.

[55] G. Tóth, B. van der Holst, I. V. Sokolov, D. L. De Zeeuw, T. I. Gombosi, F. Fang, W. B. Manchester, X. Meng, D. Najib, K. G. Powell, Q. F. Stout, A. Glocer, Y.-J. Ma, M. Opher, Adaptive numerical algorithms in space weather modeling, Journal of Computational Physics 231 (3) (2012) 870–903. doi:10.1016/J.JCP.2011.02.006.

[56] C. Tsigkanos, T. Kehrer, C. Ghezzi, Architecting dynamic cyber-physical spaces, Computing 98 (10) (2016) 1011–1040. doi:10.1007/s00607-016-0509-6.

[57] R. Al-Ali, M. Amrani, A. Barišić, F. Barros, D. Blouin, H. Giese, M. Goulão, M. Iacono, E. Navarro, K. Vangheluwe, Hans; Vanherpen, M. Wimmer, State-of-the-art on Current Formalisms used in Cyber-Physical Systems Development 1201234567891-0 1, Tech. rep., ICT COST Action IC1404 Multi-ParadigmModelling for Cyber-Physical Systems (MPM4CPS) (2017).

[58] M. Hecht, A Model Based Systems Engineering Approach to Resiliency Analysis of a Cyberphysical System, in: 2016 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), IEEE, 2016, pp. 25–28. doi:10.1109/ISSREW.2016.51.

[59] V. R. Sampath Kumar, M. Shanmugavel, V. Ganapathy, B. Shirinzadeh, Unified meta-modeling framework using bond graph grammars for conceptual modeling, Robotics and Autonomous Systems 72 (2015) 114–130. doi:10.1016/j.robot.2015.05.003.

[60] E. A. Lee, Disciplined Heterogeneous Modeling, Model Driven Engineering Languages and Systems (2010) 273–287doi:10.1007/978-3-642-16129-2{_}20.

[61] D. Varró, G. Bergmann, . Hegedüs, . Horváth, I. Ráth, Z. Ujhelyi, Road to a reactive and incremental model transformation platform: three generations of the VIATRA framework, Software and Systems Modeling 15 (3) (2016) 609–629. doi:10.1007/s10270-016-0530-4.

[62] I. Dávid, I. Ráth, D. Varró, Foundations for Streaming Model Transformations by Complex Event Processing, Software & Systems Modeling 17 (1) (2018) 135–162. doi:10.1007/s10270-016-0533-1.

[63] E. Barbierato, G.-L. D. Rossi, M. Gribaudo, M. Iacono, A. Marin, Exploiting product forms solution techniques in multiformalism modeling, Electronic Notes in Theoretical Computer Science 296 (2013) 61–77. doi:https://doi.org/10.1016/j.entcs.2013.07.005.

[64] I. Dávid, E. Syriani, C. Verbrugge, D. Buchs, D. Blouin, A. Cicchetti, K. Vanherpen, Towards inconsistency tolerance by quantification of semantic inconsistencies, in: CEUR Workshop Proceedings, Vol. 1717, COMMitMDE@MoDELS, 2016, pp. 35–44.

[65] E. Barbierato, M. Gribaudo, M. Iacono, A. Jakóbik, Exploiting CloudSim in a multiformalism modeling approach for cloud based systems, Simulation Modelling Practice and Theory 93 (2019) 133–147.

[66] S. J. I. Herzig, S. Mandutianu, H. Kim, S. Hernandez, T. Imken, Model-transformation-based computational design synthesis for mission architecture optimization, in: 2017 IEEE Aerospace Conference, 2017, pp. 1–15. doi:10.1109/AERO.2017.7943953.

[67] I. Chun, J. Kim, H. Lee, W. Kim, S. Park, E. Lee, Faults and adaptation policy modeling method for self-adaptive robots, in: Communications in Computer and Information Science, Vol. 150, Springer, Berlin, Heidelberg, 2011, pp. 156–164. doi:10.1007/978-3-642-20975-8{_}17.

[68] A. Bucaioni, S. Mubeen, F. Ciccozzi, A. Cicchetti, M. Sjödin, Modelling multi-criticality vehicular software systems: evolution of an industrial component model, Software and Systems Modeling 19 (5) (2020) 1283–1302.

[69] K. Falkner, C. Szabo, V. Chiprianov, G. Puddy, M. Rieckmann, D. Fraser, C. Aston, Model-driven performance prediction of systems of systems, Software and Systems Modeling 17 (2) (2018) 415–441. doi:10.1007/s10270-016-0547-8.

[70] E. Barbierato, M. Gribaudo, M. Iacono, Defining Formalisms for Performance Evaluation With SIMTHESys, Electronic Notes in Theoretical Computer Science 275 (2011) 37–51. doi:https://doi.org/10.1016/j.entcs.2011.09.004.

[71] I. Eusgeld, C. Nan, S. Dietz, System-of-systems approach for interdependent critical infrastructures, Reliability Engineering & System Safety 96 (6) (2011) 679–686. doi:10.1016/J.RESS.2010.12.010.

[72] S. Bliudze, S. Furic, J. Sifakis, A. Viel, Rigorous design of cyber-physical systems: Linking physicality and computation, Software and Systems Modeling (2017) 1–24doi:10.1007/s10270-017-0642-5.

[73] S. Jafer, B. Zeigler, D. D. H. Kim, A framework for rapid configuration of collaborative aviation system-of-systems simulations, in: International Workshop on Modelling and Simulation for Autonomous Systems, Springer, 2017, pp. 92–105.

[74] M. E. Vara Larsen, J. Deantoni, B. Combemale, F. Mallet, A Behavioral Coordination Operator Language (BCOoL), in: 2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems, MODELS 2015 - Proceedings, IEEE, 2015, pp. 186–195. doi:10.1109/MODELS.2015.7338249.

[75] A. Müller, S. Mitsch, W. Retschitzegger, W. Schwinger, A conceptual reference model of modeling and verification concepts for hybrid systems, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8793 (2014) 368–379.

[76] V. Gunes, S. Peter, T. Givargis, F. Vahid, A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems, TIIS 8 (2014) 4242–4268. doi:10.3837/tiis.2014.12.001.

[77] M. Zhang, S. Ali, T. Yue, R. Norgren, O. Okariz, Uncertainty-Wise Cyber-Physical System test modeling, Software and Systems Modeling (2017) 1–40doi:10.1007/s10270-017-0609-6.

[78] J. Huang, S. Barner, A. Raabe, C. Buckl, A. Knoll, A framework for reliability-aware embedded system design on multiprocessor platforms, Microprocessors and Microsystems 38 (6) (2014) 539–551. doi:10.1016/J.MICPRO.2014.02.007.

[79] D. Bumblauskas, D. Gemmill, A. Igou, J. Anzengruber, Smart Maintenance Decision Support Systems (SMDSS) based on corporate big data analytics, Expert Systems with Applications 90 (2017) 303–317. doi:https://doi.org/10.1016/j.eswa.2017.08.025.

[80] C. B. Vellaithurai, S. S. Biswas, A. K. Srivastava, Development and Application of a Real-Time Test Bed for Cyber-Physical System, IEEE Systems Journal 11 (4) (2015) 2192 – 2203. doi:10.1109/JSYST.2015.2476367.

[81] K. Grüttner, P. A. Hartmann, K. Hylla, S. Rosinger, W. Nebel, F. Herrera, E. Villar, C. Brandolese, W. Fornaciari, G. Palermo, C. Ykman-Couvreur, D. Quaglia, F. Ferrero, R. Valencia, The COMPLEX reference framework for HW/SW co-design and power management supporting platform-based design-space exploration, Microprocessors and Microsystems (2013). doi:10.1016/j.micpro.2013.09.001.

[82] R. Seiger, S. Huber, T. Schlegel, Toward an execution system for self-healing workflows in cyber-physical systems, Software & Systems Modeling (2016) 1–22doi:10.1007/s10270-016-0551-z.

[83] A. Drago, S. Marrone, N. Mazzocca, R. Nardone, A. Tedesco, V. Vittorini, A model-driven approach for vulnerability evaluation of modern physical protection systems (12 2016). doi:10.1007/s10270-016-0572-7.

[84] T. Hartmann, A. Moawad, F. Fouquet, Y. Le Traon, The Next Evolution of MDE: A Seamless Integration of Machine Learning into Domain Modeling, Proceedings - ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems, MODELS 2017 (2017) 180doi:10.1109/MODELS.2017.32.

[85] I. Dragomir, I. Ober, C. Percebois, Contract-based modeling and verification of timed safety requirements within SysML, Software and Systems Modeling 16 (2) (2017) 587–624. doi:10.1007/s10270-015-0481-1.

[86] R. Lerm, D. Doering, C. E. Pereira, Design-Space Exploration Tool for the HIPAO Methodology, IFAC-PapersOnLine 48 (10) (2015) 93–98. doi:10.1016/J.IFACOL.2015.08.114.

[87] A. Murguzur, H.-L. Truong, S. Dustdar, Multi-perspective Process Variability: A Case for Smart Green Buildings (Short Paper), in: 2013 IEEE 6th International Conference on Service-Oriented Computing and Applications, IEEE, 2013, pp. 25–29. doi:10.1109/SOCA.2013.40.

[88] A. Koubeissi, A Multi-Level Data-Centric Model of System of Systems Engineering, in: ACIT 2018 - 19th International Arab Conference on

Information Technology, Institute of Electrical and Electronics Engineers Inc., 2019. doi:10.1109/ACIT.2018.8672676.

[89] M. Tröls, A. Mashkoor, A. Demuth, A. Egyed, Ensuring safe and consistent coengineering of cyber-physical production systems: A case study, Journal of Software: Evolution and Process (2020). doi:10.1002/smr.2308.

[90] CORE 2018 ranking, http://portal.core.edu.au/conf-ranks/, http://portal.core.edu.au/jnl-ranks/, accessed: June 2020.

[91] Scimago ranking, https://www.scimagojr.com/, accessed: June 2020.

[92] Google Scholar, https://scholar.google.com/, accessed: June 2020.

[93] P. Mohagheghi, V. Dehlen, Where is the proof?-a review of experiences from applying MDE in industry, in: European Conference on Model Driven Architecture-Foundations and Applications, 2008, pp. 432–443. doi:/10.1007/978-3-540-69100-6{_}31.

[94] A. Barišić, V. Amaral, M. Goulão, Usability Driven DSL development with USE-ME, Computer Languages, Systems and Structures (ComLan) 51 (2018) 118–157. doi:10.1016/j.cl.2017.06.005.

[95] B.-H. Schlingloff, Cyber-Physical Systems Engineering, in: Engineering Trustworthy Software Systems, Springer, 2016, pp. 256–289. doi:10.1007/978-3-319-29628-9{_}5.

[96] D. Broman, P. Derler, J. C. Eidson, Temporal issues in cyber-physical systems, Journal of the Indian Institute of Science 93 (3) (2013) 389–402.

[97] P. Derler, E. A. Lee, A. Sangiovanni Vincentelli, Modeling cyber-physical systems, Proceedings of the IEEE 100 (1) (2012) 13–28. doi:10.1109/JPROC.2011.2160929.

[98] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, S. Achiche, Design, modelling, simulation and integration of cyber physical systems: Methods and applications, Computers in Industry 82 (2016) 273–289. doi:10.1016/J.COMPIND.2016.05.006.

[99] Y. Z. Lun, A. D'Innocenzo, I. Malavolta, M. D. Di Benedetto, State of the art of cyber-physical systems security: An automatic control perspective, The Journal of Systems and Software 149 (2019) 174–216.

[100] R. Al-Ali, Industrial Use Cases of Cyber Physical Systems in EU Projects: Preliminary Study., in: Preprints of position papers of the Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 2017, pp. 187–193.

[101] A. Wortmann, O. Barais, B. Combemale, M. Wimmer, Modeling languages in industry 4.0: an extended systematic mapping study, Software and Systems Modeling 19 (1) (2020) 67–94.