

# A comprehensive survey on modeling of cyber-physical systems

Imen Graja<sup>1</sup> | Slim Kallel<sup>1</sup>  | Nawal Guermouche<sup>2,3</sup> | Saoussen Cheikhrouhou<sup>1</sup> | Ahmed Hadj Kacem<sup>1</sup>

<sup>1</sup>ReDCAD, University of Sfax, Sfax, Tunisia

<sup>2</sup>CNRS, LAAS, Toulouse, France

<sup>3</sup>INSA, Université de Toulouse, Toulouse, France

## Correspondence

Slim Kallel, ReDCAD, University of Sfax, Sfax, Tunisia.

Email: slim.kallel@fsegs.rnu.tn

## Summary

Modeling Cyber-physical systems (CPS) is a challenging step that requires a lot of background from both the cyber and physical fields. However, there is a lack of studies in the existing literature that discuss the state of the art in modeling CPS or explore the research gaps in this area. In this paper, we survey existing approaches to modeling CPS. We focus on studying the considered CPS properties. Based on this study, we classify these properties and discuss their importance in different application domains. Moreover, research directions are presented to address key challenges in the specification of CPS models.

## KEYWORDS

cyber-physical systems, CPS properties, modeling

## 1 | INTRODUCTION

According to the U.S. National Science Foundation, the term Cyber-Physical Systems was coined to describe a broad range of new systems that integrate embedded computing technologies into the physical world.<sup>1</sup> Then, the National Institute of Standards and Technology formed a group of experts to help define these systems. The group defined CPS in more general terms as smart systems that are co-engineered connected networks of physical and computational elements aiming to improve the quality of life in many domains. CPS are also related to other fields such as embedded systems, wireless sensors networks (WSN), and Internet of things (IoT). That said, there are some key differences between the study of these adjacent fields and CPS stemming from the complexity and the broadness of CPS systems. For example, compared to WSN, CPS involve many more physical components. Embedded systems focus primarily on computing elements, while the IoT is focused on physical components (ie, things) that are connected to the Internet and to each other.<sup>2</sup> In cyber-physical systems, physical and computational components are equally important. Such systems have the advantage that smart physical devices, and wireless networks are used to create intelligent services based on information and components from the related physical environment.<sup>3</sup>

CPS may consist of multiple elements such as actuators and sensors, which are connected together and integrated under a smart decision system.<sup>4</sup> The physical devices provide activities that constitute the physical part of the system. Computing and controlling the dynamics of the physical behavior are the main activities of embedded systems.<sup>5</sup> The cyber part of CPS is a set of computing operations, which depend on embedded, cloud, and web-based activities. These cyber activities analyze the collected data and interact with the physical environment in order to make decisions. Therefore, there is a need to support both physical and cyber elements when modeling CPS. Furthermore, the physical elements of CPS are characterized by their continuous nature such as a walking action of a robot or a motion of a car. This continuous behavior can be represented solely by algebraic or differential equations, while the cyber part is mostly described as a discrete behavior.<sup>6</sup> Therefore, both continuous and discrete behaviors have to be considered when modeling CPS. In addition, executing a physical action can impact the state of the related device, and some other involved physical entities. For example, a transport activity can change the location of both the car (ie, device) and the traveler (ie, physical entity).<sup>7</sup> Thus, further information about the cyber and physical resources need to be considered during the modeling process.<sup>8</sup>

To develop CPS, model-based design plays an essential role.<sup>5</sup> Modeling such systems gives us insight into their behavior in the real world. In order to save money and time, the modeled systems may be tested, analyzed, and verified off-line before finally getting to the execution step. Indeed, CPS modeling that considers the different properties of such systems is necessary to identify different problems and develop effective self-adaptation solutions. A complete model of CPS needs to represent a coupling of the physical behavior, computations, and the environment in which the systems reside.<sup>9</sup> The cyber operations control and monitor the physical world, whereas the physical actions have an effect on the computations and

vice-versa.<sup>10</sup> Consequently, in order to accurately model CPS, the designer needs to consider the heterogeneity (ie, heterogeneous elements and behaviors), complexity, physical-aware aspects, cyber-aware aspects, and the close interaction between cyber and physical elements. In addition, since such systems are typically complex, modeling should ensure easy and understandable representation of CPS.

A number of useful surveys cover specific CPS modeling challenges. In the work of Broman et al,<sup>11</sup> the authors identify concerns of stakeholders and parts of the CPS models. A review of different formalisms, languages, and tools suitable for modeling various aspects of the defined concerns was fulfilled. With an emphasis on the modeling and simulation of CPS, in the work of Hehenberger et al,<sup>12</sup> the authors provide a study of the CPS design literature. Focusing on different CPS design requirements, aspects and applications, the work presented by Khaitan and McCalley<sup>13</sup> provides valuable insight into how to deal with those specific CPS modeling challenges and studies how different real-world systems are modeled. While these previous works provide studies that deal with different specific CPS modeling requirements, they still leave a gap in identifying the different CPS non-functional properties and how to select a set of modeling languages that allow the designer to specify an understandable representation of these properties.

CPS are defined using a set of non-functional and functional properties. The functional properties describe a particular behavior of the CPS, whereas non-functional properties characterize abstract behavior of these systems and define the restrictions of their real-time execution (eg, time related properties, energy, cost, etc).<sup>14</sup> Considering these properties during CPS modeling is of paramount importance. For instance, time-related properties can be crucial, particularly in critical domains such as health domain. In addition, dependencies can exist between the different CPS properties. For instance, physical properties can affect timed-related properties of a CPS system (eg, windy weather can affect the arrival time of a flying drone). Specifying and verifying non-functional properties at design-time is extremely important for the development process.

To summarize, modeling CPS requires considering various aspects of CPS such as the cyber and physical aspects, dynamic behavior, involved physical entities, and non-functional properties. To consider these requirements, several researchers have used and proposed modeling languages. The aim of this paper is to present a survey on the specification and modeling of CPS. To achieve that, we identify the considered properties and their importance in the development of CPS. We also studied the different languages and formalisms used in the literature and tried to answer the following questions:

- Which modeling languages are used to specify CPS?
- What are the supported CPS aspects within the modeling approaches?
- Which non-functional properties are specified when modeling these systems?
- What are the studied application domains of CPS?

This paper is organized as follows. Section 2 briefly describes our research methodology and the taxonomy, which is used in its following sections. Section 3 presents a classification of the studied modeling technologies and reviews some existing approaches in this context. Next, Section 4 presents a classification of the non-functional properties into a finite set of domain-independent categories. We then sort the studied works into the derived categories. Section 5 presents another classification of the selected papers according to their application domain in order to highlight their important non-functional properties. In Section 6, we identify future research challenges to address in the field of CPS modeling. Finally, Section 7 concludes the paper.

## 2 | RESEARCH METHOD AND TAXONOMY

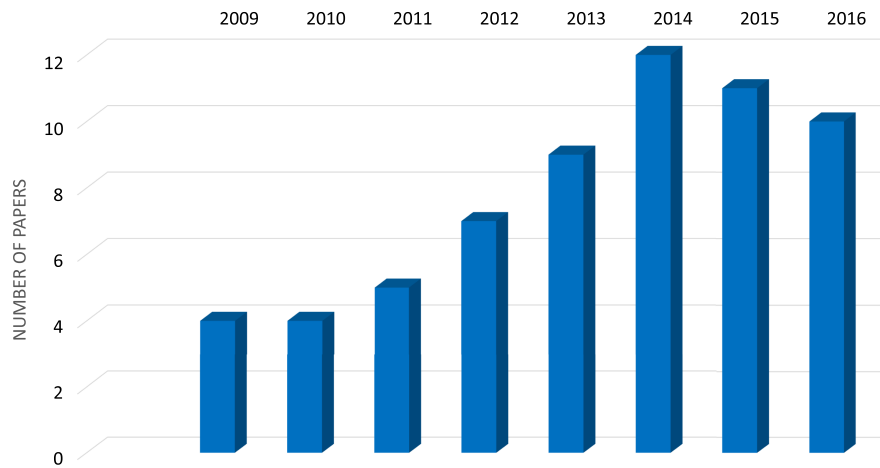
The study of this survey relies on the methodology proposed by Kitchenham and Charters<sup>15</sup> We follow the proposed guidelines to produce a literature survey in which we collect different pertinent modeling approaches, define a classification of these works based on specific questions, identify the supported properties and studied application-domains, and evaluate them to point out pertinent research directions. We concentrated our search on known scientific databases. These electronic databases include IEEE xplora,<sup>16</sup> SpringerLink,<sup>17</sup> ACM Digital Library,<sup>18</sup> and Sciencedirect (Elsevier).<sup>19</sup> We began our research by collecting studies about cyber-physical system modeling approaches. Second, we filtered the list of collected papers. The selection process was performed through applying the following steps:

1. Identify relevant studies about modeling CPS.
2. Include other studies about verifying or analyzing CPS that deal with CPS models.
3. Exclude the studies that are off-topic based on their titles and abstracts.
4. Collect primary studies through a full text review.
5. Obtain the final list of publications that focus on modeling cyber-physical systems.

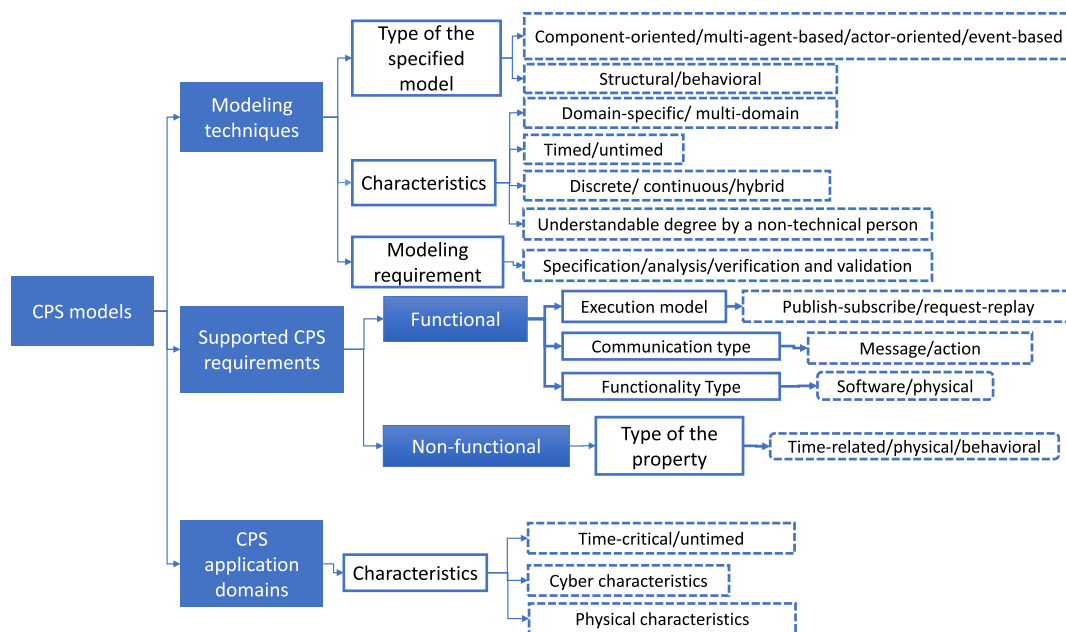
Our search resulted in 62 articles, which are mostly published in well-known journals and conferences. Figure 1 shows the repartition per year of the studied works since 2009.

Through the studied CPS modeling approaches, we distinguish three dimensions: the modeling techniques, the supported properties, and the application domains. Dependencies may exist between these dimensions. Indeed, the choice of the modeling techniques depends on the the application domain and especially on the required properties.

Figure 2 shows the considered taxonomy and the underlying relationships. Both the dimensions and their sub-dimensions consist of a set of nodes that are connected by directed arrows describing their classification factors. The filled nodes with a solid color represent the dimensions



**FIGURE 1** The number of studied publications by year



**FIGURE 2** The taxonomy

discussed and classified in detail throughout this paper. The transparent nodes (ie, no fill), with a solid outline, are the classification classes. The dashed nodes relate to the definition of different types associated to the proposed classes. We note that the nodes with dashed outline are related to the incorporation of various kinds of each category. Those types indicate mutually different choices, for instance, the characteristic of a modeling technique is concerned with a discrete, continuous, or hybrid system. We also note that the gray nodes are proposed in the taxonomy but not studied or being investigated in this paper.

The modeling techniques are discussed according to the type of the specified model, characteristics, and requirements of using each technique. Researches have used different models to represent CPS such as component-oriented model, actor-oriented model, multi-agent-based model, and event-based models. The modeling technique can be used to specify the structural aspect, or/and the behavioral aspect of the system. Moreover, the modeling technique characteristics relate to the incorporation of timing issues, the easiness of the specified model to be understood by non-technical users, and the nature of the model (ie, continuous, event-discrete, hybrid). The reason of using a modeling technique can differ according to the need of the designer: to specify, analyze, evaluate the CPS under various conditions, or verify the system as a proof for correctness of the model.

The supported CPS requirements are classified as either functional or non-functional properties. Those requirements address the behavior of cyber-physical systems. The functional requirements describe how the system react to specific given inputs and its provided behavior. In CPS, the functional behavior defines two types of execution models, which are the publish-subscribe (event-driven) of request-reply (conversational). The communication between the different components in the system can be through messages or physical actions. These components provide functionalities that can be software functions or physical actions. We note that the functional requirements, which are proposed in the taxonomy, are not studied or going to be investigated in this paper. The non-functional requirements are the constraints that characterize the behavior offered by the

system. Throughout the paper, we will discuss and classify those requirements according to their types. We will also discuss the other dimensions that we proposed, according to their ability to specify or support such properties.

The application domain is the third dimension. This dimension shows that, when modeling CPS, the designers use existing applications to evaluate their approaches. It is convenient to group the applications into different domains that support the same characteristics such as the timing issues, the common cyber aspects, and the common physical aspects. In this paper, we will discuss different studied application domains and their related properties.

### 3 | MODELING LANGUAGES

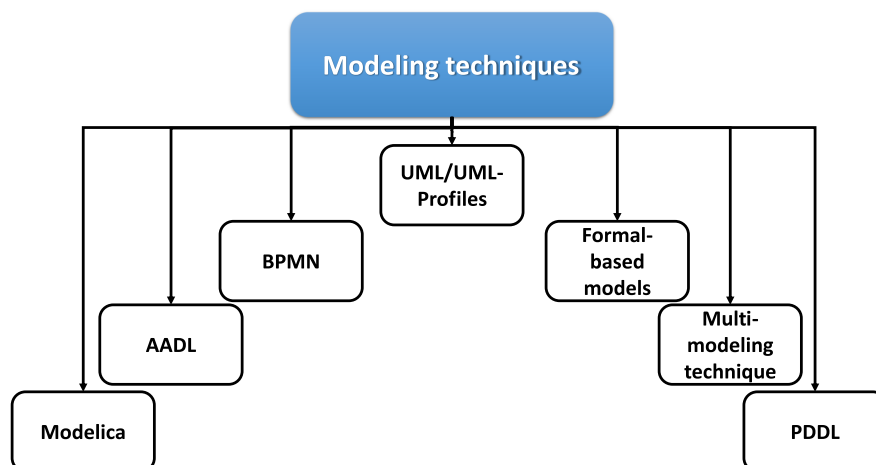
Several research works have been proposed to support CPS modeling. These works have proposed to use or to extend existing formalisms. Figure 3 shows the main languages used in the literature.

#### 3.1 | Modelica

Modelica is a multi-domain language for component-oriented modeling of complex systems, especially those containing physical components (ie, mechanical, electrical, thermal, control, etc). Modelica supports modeling both the discrete behavior and the continuous behavior that can be described using event triggers and differential algebraic equations, respectively. The structural model specifies the components and the connections between them. A connector must contain all information needed to describe the interaction. For example: for electrical components, interactions need to be defined with the levels of voltage and current. Then, to define the behavior, the designer adds a definition of parameters for each component and a set of equations. As proven in other works,<sup>20-22</sup> Modelica has good advantages in modeling and simulating system behavior. However, the particularity of this language is that it is intended mostly to model physical systems. On one hand, none of the studied modelica-based approaches captures the interactions between physical and other software components. On the other hand, the Modelica specification is difficult to read and understand by a non-technical person. Furthermore, the authors of some approaches intend to partially specify time-related and physical properties, which are mostly related to the interactive behavior only such as the communication time between components.

#### 3.2 | Architecture description languages

Architecture description Languages (ADL) are a set languages for model-based specification and analysis. Architecture Analysis & Design Language (AADL)<sup>23</sup> is a standard that has been developed for real-time embedded systems. It allows the designer to specify and analyze the system's structure (software and hardware components and connections). This language also has been extended with a behavioral annex that brings into the architecture the ability to specify the behavior of the components and their interactions.<sup>24</sup> However, AADL does not allow the designer to specify the dynamic physical behavior. For this reason, Banerjee et al<sup>25</sup> extend AADL to support the dynamic behavior of the physical processes (ie, differential equations) through the development of a new annex called BAN-CPS. The authors gave a modeling framework called BAND-AiDe and analyzed the component of the Body Area Networks (BAN). This framework enables the modeling of the system using the extended AADL in order to verify its safety and sustainability. Nevertheless, the development of a BAN-CPS annex is limited to support only physical properties that characterize the physical behavior, environmental factors, and physical entities such as the blood glucose level, blood pressure, and tissue temperature of the human body. In addition, their approach does not support time-related properties.



**FIGURE 3** The CPS modeling languages

In general, to use AADL, the designer does not need to necessarily be an expert in this language, but the designed model should be prepared to be widely understandable by other users at a technical level, which is quite hard with the use of AADL. In addition, AADL lacks the capability to support all the aspects of CPS in order to model both cyber and physical behaviors.

### 3.3 | Business process model and notation

Business Process Model and Notation (BPMN) is a standard for business process modeling that provides a graphical notation for specifying business processes.<sup>26</sup> It has been maintained by the Object Management Group (OMG). In order to model self-adaptive workflows for cyber-physical systems, Seiger et al<sup>27</sup> present a process-based framework in which the abstract system behavior is modeled using BPMN. In fact, this approach shows the strong need for modeling complex collaborative behavior of networked CPS and expressing complex flows of data (event-based) within high-level CPS processes. Furthermore, in order to consider real-world context situations, in the work of Marrella et al,<sup>28</sup> the authors introduce SmartPM which is a framework for automated process adaptation in case of unanticipated exceptions. The designer uses the BPMN to define the process control flows among a set of tasks selected from a repository. However, when modeling the cyber-physical behavior, the proposed framework PROTEUS<sup>29</sup> and SmartPM do not allow the designer to specify a large set of non-functional requirements. The current used version of BPMN lacks the capability required to support some CPS concepts.

As we discussed in our previous work,<sup>8</sup> other works present modeling research initiatives to extend BPMN for other related fields such as IoT and WSN. To model the IoT behavior, in other works,<sup>30–33</sup> the authors propose new BPMN notation concepts. The extended notations are related to only some real-world aspects of IoT such as the devices that provide the physical actions and the entities of interest. Then, to handle non-functional properties, in the work of Martinho and Domingos,<sup>34</sup> the authors propose related specification concepts in business process level for IoT such as the cost of executing a task (eg, energy consumption). In order to effectively model WSN processes, in the work of Sungur et al,<sup>35</sup> the authors propose a set of extensions for BPMN 2.0. The WSN pool represents the set of new structured activities, which are provided by the WSN resources. In the work of Tranquillini et al,<sup>36</sup> the proposed BPMN extension allows the designer to integrate wireless sensor network applications with business processes. Throughout this paper, the authors introduce the actuation and sensing logic that is specified at a process level to collaborate and interact with other existing business processes (eg, enterprise information systems). However, these existing solutions partially address the modeling of cyber-physical systems. In fact, the extended versions of BPMN allows the designer to only model IoT and WSN physical services instead of supporting the cyber elements of a CPS process, physical elements of a CPS process, and the non-functional properties such as the time-related and physical properties.<sup>8</sup>

### 3.4 | UML/UML profiles

Other than BPMN, the Object Management Group (OMG) standardized another number of notations that can be used to fulfill the modeling requirements of cyber-physical systems. In particular, the Unified Modeling Language (UML2)<sup>37</sup> is widely used for behavioral modeling of software systems using UML state machines and activity diagrams. For modeling physical systems, some standard UML profiles such as MARTE<sup>38</sup> and SysML<sup>39</sup> have many features that are useful for such systems.

SysML is an extension of UML to better address the system requirements and parameters. At a high level of abstraction, in the work of Thramboulidis,<sup>40</sup> the designer uses block diagrams to model the physical component. In addition, using SysML, the designer can model the behavior of the system through an activity or action diagram. However, this language does not deal with non-functional properties.

MARTE is a domain-specific modeling language for real-time and embedded systems. MARTE supports additional non-functional requirements and provides parameters for real-time systems. Mallet<sup>41</sup> illustrates possible uses of MARTE to model CPS in which the time-related properties are considered as extra-functional properties. They are related to the start time, finish time, and duration of an action. Although, MARTE enables direct and easy expression of the cyber and physical phenomena, it still lacks for not only an expressive and rich time model but also a specification of the physical properties.

In order to describe time-continuous (ie, physical) and event-based (ie, cyber) dynamic behavior, other UML profiles are proposed such as Modelica Modeling Language (ModelicaML). ModelicaML is a UML profile that extends the UML meta-model with new constructs of Modelica concepts. This profile enables the designer to specify the graphical model and generate an executable Modelica code to evaluate the system using simulations. In this context, the approaches presented in other works<sup>6,42</sup> use ModelicaML to model cyber-physical systems. However, these approaches do not support the specification of non-functional properties that can affect the correctness of the system execution.

Moreover, researchers use other UML profiles specific to certain systems such as the mechatronic systems to model CPS. Mechatronic systems combine mechanical, electrical, and software parts that interact tightly with each other to control mechanical operations.<sup>43,44</sup> According to Guerineau et al,<sup>45</sup> CPS and Mechatronics are two different complex systems. On the one hand, with the mechatronic systems, there is a higher focus on the physical elements in terms of mechanics and electronics that interact through the software systems. On the other hand, CPS originate from an information technology domain and strongly focus on both physical and cyber elements that compute, communicate, control, and ensure their integration with the physical world. Accordingly, mechatronics and mechanical engineering play important roles in building CPS. In other words, some proposed techniques used to model mechatronic systems can be extended to model CPS such as MechatronicUML (MUML). MechatronicUML is a UML profile specifically tailored toward the modeling of mechatronic systems. It is based on a well-defined subset of UML diagrams and formal analysis method.<sup>46</sup> Dealing with the coordination behavior of cyber-physical systems, Pohlmann et al<sup>47</sup> presents a design method that enables the

modeling of the systems' components and their message-based communication using MUML. Such interactions require hard real-time constraints. In this example, the supported time-related properties are used to just define some restrictions on the communication behavior (eg, transmission time, waiting time to send or receive a message, etc). Accordingly, MUML partially supports CPS aspects and time-related properties.

Furthermore, other proposed domain specific modeling language (DSML) offer expressive power focusing on a particular CPS problem through some appropriate notations and abstractions, which aim to tightly fit with the designer's needs. In this context, Aziz and Rashid<sup>48</sup> define a DSML in terms of a meta-model that is implemented as a UML profile. This approach supports some cyber-physical elements such as the device, service, and relationships between them. In fact, the proposed approach allows the designer to model the structural and behavioral aspects of CPS. However, there is no clear separation between the cyber and the physical parts of the system, and it does not permit to model non-functional properties, and it is limited to the modeling phase since no verification mechanism is provided.

### 3.5 | PDDL

The Planning Domain Definition Language (PDDL) is used for specifying automated planning problem. Hence, such languages are known as action languages. PDDL separates the model of the planning problem in two major parts, which are the domain description and the related problem description. In the service oriented computing (SOC) research field, some works such as the works of Huang et al<sup>49,50</sup> present the CPS functionalities as services to facilitate the composition task. The authors propose the extension of the web-service ontology model, OWL-S, used to describe the web services syntactically and semantically. The extension addresses the physical services, physical entities, and the state of the world of interest. The authors specify the composition of the CPS services as a CPS workflow pattern, which is a sequence of actions that move the system from the initial state to the desired goal described using PDDL. Since physical properties may impact the operation of the services and affect their availability and functionality, the proposal in other works<sup>51,52</sup> effectively describes the states of the world and support the specification of the physical properties. The above approaches, however, do not consider the detailed time-related properties that can affect the composition of cyber and physical services as well.

### 3.6 | Formal-based models

A large body of early research addresses the modeling of CPS using formal methods. Formal methods are based on the use of mathematically precise specification language to describe the system behavior and their related properties.<sup>53</sup> Examples of modeling formalisms with the purpose of specifying CPS, are pi-calculus, petri-net, timed automata, hybrid automata, etc.<sup>54-59</sup>

In the work of Wang et al,<sup>54</sup> the authors propose an approach to model the CPS services using *Time-Space Pi-Calculus*, which is an extension of pi-calculus with time and space operators. This approach supports the specification of constraints on the position, energy, and transmission time of the exchanged messages between services. Here, the aim of this approach is to verify the collaborative behavior of the specified services (eg, livelock or deadlock). However, physical and time-related properties are partially considered, and the used formalism lacks the capability required to specify the dynamic physical behavior of the system.

The notion of time is captured by temporal logic-based formalisms such as *timed petri-net* and *timed automata*. In fact, a temporal logic is a formal method for representing and reasoning about propositions qualified in terms of time.<sup>55</sup> Some of the studied research use existing temporal logic-based formalisms to define CPS, while others cover several useful extensions in order to capture some CPS aspects. In the work of Pajic et al,<sup>56</sup> a case study demonstrates the application of *timed automata* for the modeling and analysis of cyber-physical systems. In the work of Basile et al,<sup>60</sup> the authors use *colored timed petri-net*, which is an extension of petri-net, to model a detailed representation of the CPS behavior based on a service oriented multi-agent approach. Otherwise, Shao et al<sup>57</sup> extend the existent propositional temporal logic with spatial logic in order to develop a spatio-temporal logic to describe the spatio-temporal properties. Similarly, based on temporal logic, Garro et al<sup>61</sup> define FORM-L (ie, formal requirement modeling language), which is a new formal language for cyber-physical system property modeling such as time-related properties. In general, we emphasize that using temporal logic-based formalisms, the time-related properties can be specified and analyzed at design-time. However, other non-functional properties such as physical properties are either not supported or partially considered within some extended languages. In addition, these approaches often do not completely model CPS since they discard the continuous dynamic behavior of the physical part of the system. Hence, these capabilities make these formalisms unsuitable to specify CPS.

To cover the specification of the physical behavior, *hybrid automata* provides a formal modeling and specification environment.<sup>62</sup> Bak and Chaki<sup>63</sup> show the strength of using hybrid automata to model the physical behavior of the system. Since hybrid automata is a mathematical model that combines discrete control graphs with continuous evolving variables, which are described by a set of ordinary differential equations, it enables the modeling of CPS. On one hand, dealing with periodical on-line verification, Li et al<sup>64</sup> formally express the CPS and their time bounded future behavior using a hybrid automata. To explain, the defined period of time is the predicted time in which the designer verifies the future behavior of a running model. At the same time, designers formally express different physical properties to characterize the physical behavior such as the constraints on the movement of the ventilator cylinder and its maximum height. On the other hand, dealing with off-line modeling and verification of CPS, Ye et al<sup>59</sup> introduce a CPS service composition approach. To accomplish this goal, they use hybrid automata to model the services of CPS, express each service's dynamic physical behavior using equations, and specify their physical properties. Since hybrid automata consider only that the state of the system evolves with time, this formalism is extended to support other dimensional variation of CPS parameters. Banerjee et al<sup>58</sup>



define a spatio-temporal hybrid automata as an extension, which can be used to model the variation of the CPS behavior in both time and space. Hence, the physical behavior is constrained by both time bound and space bound. Despite the effort of modeling, the physical behavior, and their related non-functional properties, these studied approaches lack the consideration of the cyber elements of CPS and do not focus on specifying the time-related properties.

In general, hybrid automata are best suited for modeling physical aspects of CPS behaviors, and its strength is that the verification can be fully automated through the model checker (eg, reachability analysis). However, it suffers from a complexity in specifying the non-functional properties.<sup>53,65</sup> Finally, the visual representation of the designed model, using hybrid automata, is quite hard to understand by the people who will implement these models. For these reasons, researchers usually propose a multi-modeling technique for CPS.

### 3.7 | Muti-modeling technique

Since, the existing modeling languages and their proposed extensions lack the capabilities required to model CPS and cover their characteristics, one of the most promising techniques deals with the combination of multiple modeling formalisms to specify CPS.<sup>66</sup> In this context, we discuss some existing research papers, and we investigate how to best make use of this technique.

To represent a cyber-physical system as a hierarchical multi-model, which is a type of multilevel models, Lee et al<sup>66,67</sup> develop an open-source software framework called Ptolemy II. Ptolemy II is a modeling and simulation environment for heterogeneous systems that can be described using the combination of different formalisms such as discrete events,<sup>68</sup> continuous time,<sup>69</sup> finite state machines,<sup>70</sup> synchronous reactive,<sup>71</sup> process networks,<sup>72</sup> and several varieties of dataflows.<sup>73</sup> This framework is built upon the notion of actors (ie, components) and directors, which control their interactions. The focus is to assemble the components using well-defined models of computation, which describe how the actors are related to each other. We also mention that Ptolemy II can perform simulation of heterogeneous models. It provides mechanisms for heterogeneous models to interact concurrently. These mechanisms can realize hierarchical multi-modeling and co-simulation. The co-simulation consists in combining two different simulators using a standardized interface, which is responsible of the propagation of information between the two constituent models. Then, within a distributed CPS, Lee et al develop PTIDES (Programming Temporally Integrated Distributed Embedded Systems),<sup>74</sup> which is an extension to the Ptolemy II simulation environment. The authors extend Ptolemy II to deal with time-related behavior and clock synchronization problems.<sup>75</sup> In this context, the supported time-related properties are about the worst-case execution time bounds and the desired timing of interactions with the environment (ie, execution time and communication delays). Despite the fact that the Ptolemy II models are a specification of both the physical and cyber behavior, they lack the ability to support physical and rich time-related properties. In addition, Ptolemy II can only be used by expert designers, who must know the different modeling formalisms. Moreover, the designed models are quite complex, complicated, hard to read, and difficult to build.

Zhang<sup>76</sup> integrates AADL with Modelica, UML, and formal models to specify automotive CPS. The cyber part is specified with AADL and UML, while the physical part is modeled using Modelica. Fitzgerald et al<sup>77</sup> combine discrete-event models (ie, the models that represent the cyber elements in CPS) with continuous-time models (ie, the models that represent the physical phenomena in CPS). They use both the vienna development method specification language<sup>78</sup> for the discrete-event modeling and 20-sim method<sup>79</sup> for the continuous-time specification. The 20-sim method is also a multi-formalism technique supporting equations, block diagram, and bond graph description forms. Such co-models are analyzed through co-simulation, where the two models share data during simulation. In fact, this approach considers both the cyber and the physical elements, but it does not support the non-functional properties.

Since there are numerous heterogeneous languages used for modeling CPS, Simko et al<sup>80,81</sup> proposes CyPhyML, which is a component-based integration language for composing various models such as those designed using Modelica or a signal flow language such as ESMoL.<sup>82</sup> The aim of using CyPhyML is to formalize the structure and the behavior of CPS using the same logic-based formalism through establishing a semantic matching between languages. However, this method is still limited and the tool does not provide any guidance for the verification of the model.

Finally, there are several approaches that combine multiple modeling formalisms. This combination tremendously supports and allows the designer to model the different CPS characteristics, but it requires that the designer knows how to use the different modeling formalisms. As a result, the multi-modeling technique is complicated, hard to understand, and difficult to use by non-technical users. In particular, none of the studied approaches focuses on specifying a large set of non-functional properties when modeling CPS.

Table 1 summarizes the studied papers and their modeling techniques.

**TABLE 1** Classification of CPS modeling approaches

Modeling Techniques	References
Modelica	20-22,83
ADL/AADL/Acme	25,84-88
UML/UML Profiles	6,41,47,48,89-91
BPMN	27,28,92,93
Multi-modeling techniques	67,76,77,81,94-97
Formal-based model	54,56-59,60,61,63-65,98-102
PDDL	50-52,103,104

## 4 | NON-FUNCTIONAL CPS PROPERTIES

As stated above, CPS can be specified using several kinds of properties. In this section, we focus on the studied non-functional CPS properties in the literature. We distinguish three classes: (1) time-related properties, (2) physical properties, and (3) behavioral properties.

### 4.1 | Time-related properties

According to other works,<sup>10,105</sup> the central properties of the physical system are its temporal dynamics where physical actions must be executed at the right and precise time. The time taken to produce a computing task is considered as a quality factor. Time-related properties are a strict requirement for CPS where a missed deadline may render the entire application incorrect.<sup>105</sup> They are the conditions for controlling CPS behavior over time. In our research, we categorize the time-related properties into four classes: (1) the temporal properties related to an activity, (2) the time-dependent occurrence between different activities, (3) the global temporal properties, and (4) the communication time. In fact, the activity is the behavior that the cyber-physical application performs. The time-related properties that restrict the start, finish, execution time, or recurrent behavior of an operation or action are the set of temporal properties related to an activity. The time-dependent occurrence between different activities are the properties that restrict the time elapsed between the end of one operation and the start of a different one. For example, the light-rail train should depart within three minutes of the arrival of the bus. Additionally, the global properties are the set of time-related constraints over the whole behavioral model of the CPS application such as the deadline. Last, the communication time is the time taken to receive a message or to detect an event such as receiving a postal mail.

We noticed that most of the studied modeling approaches in the literature include partial time-related properties. For example, using MARTE, in the work of Mallet,<sup>41</sup> time is represented as a collection of clocks. These clocks are defined as a set of specific events on which the constraints can be applied. The authors show that, for a given action, the designer can define constraints on the start event, finish event, duration, and send/receive time of a message. In addition to these properties, using timed automata, Pajic et al<sup>56</sup> allow the specification of time-related constraints on the coordination and recurrent behavior such as the periodic behavior. For instance, the designer specifies a car-2-car communication scenario,<sup>47</sup> under sensitive time constraints such as the duration, communication time, and deadline. In these papers, the authors specify the coordination behavior by means of state-based models with a continuous control specification, in order to verify safety requirements. However, we noticed that other time-related properties such as the constraints on a cycle (ie, a recurrent sequence of actions/operations) and temporal dependencies between different behaviors of CPS components are not yet properly supported.

### 4.2 | Physical properties

Physical properties are the constraints on the characteristics of a physical entity, its behavior, or its execution context in CPS. These properties can be related to the space (ie, information about the location) in which the physical actions can be provided.<sup>106</sup> Alternatively, since CPS require close interaction between the cyber and physical worlds, in both time and space,<sup>58,107</sup> we can identify the spatio-temporal properties. For example, the state of the vehicle, when moving, changes continuously according to both the space and time. The characteristics of the involved physical entity and the device (eg, the provider of the physical action) can also affect the behavior of a cyber-physical system. In particular, the state of the device or the physical entity can be either static or dynamic. The static states stay unchanged when executing the application (eg, the load capacity of a car), while the dynamic states are those that change with the time or due to other factors (eg, the velocity of a car). Yet, another approach would be to consider that the cost-related properties are the constraints on the cost of using or executing an action/operation. For instance, the cost-related property can be the constraint on the consumed quantity of energy or the fee expected to be paid for executing the required actions/operations. Finally, the state of the environment can affect the execution of the CPS physical actions. Therefore, constraints on the physical environment and context like temperature, humidity, sound, light, weather, etc, are important to consider at an abstract level of the CPS model.<sup>108-110</sup>

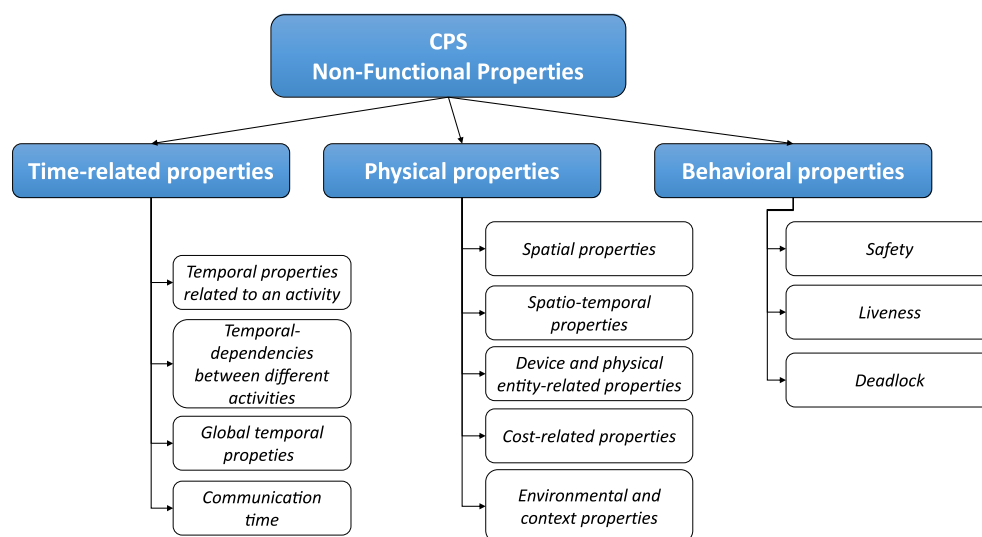
The majority of studied approaches focus on the physical behavior in CPS. They allow the specification of different physical properties such as those of the environment where a concrete CPS stays, device-related characteristics, spatial properties, spatio-temporal properties, and cost-related properties. First, the context situations and physical characteristics of the device can affect the functioning of the physical actions. The authors in other works<sup>51,104,111</sup> specify the characteristics of each involved physical element in the system (eg, device) and its required properties using an ontology model. The aim of these models is to allow a proper device selection and service invocation at run-time. As an illustration, in an evacuation scenario, knowing the different types of cars and their characteristics (eg, load capacities and number of seats) allows the selection of the needed number of vehicles at run-time. Ye et al<sup>59</sup> define the device-related, environmental, and contextual attributes through a set of variables related to the dynamic physical behavior that is formally represented using equations of time. For example, the state, size, position, energy, and velocity of a car are the attributes of a car's acceleration behavior. Second, Abbas et al<sup>65</sup> describe the spatial requirements using discrete as well as continuous state variables. In fact, spatial properties have been extensively studied in CPS. For example, in the work of Petnga and Austin,<sup>106</sup> the authors introduced a space-based modeling and reasoning approach of the CPS behavior. These spatial models support the representation of spatial entities according to which the space can be viewed as an arrangement of points, interval, array, region, or multi-dimensional space. Third, in order to



consider two dimensional variation of parameters, in the work of Banerjee and Gupta,<sup>58</sup> the authors extend the existing formalisms with the aim of modeling the spatio-temporal variation of physical parameters in CPS. According to Shao et al,<sup>57</sup> the CPS dynamics can be spatio-temporal in nature. For example, the train state can continuously vary with both time and space, so to avoid train crash, the designer defines spatio-temporal properties that guarantee the safety of this behavior. Fourth, to address particular issues associated with cost-related properties, cyber-physical energy systems research area is identified by modeling energy systems, controlling energy efficiency (eg, the monitoring and prediction of electrical energy consumption) and managing energy resources as detailed in other works.<sup>112,113</sup> Finally, physical properties are widely studied when modeling CPS; however, they deserve more research attention in order to design a predictable CPS behavior.

### 4.3 | Behavioral properties

In addition, in order to evaluate the distributed behavior of these systems as a whole, the designers need to consider other behavioral properties. Behavioral properties are the constraints on the overall concurrent system behavior.<sup>98</sup> These properties describe some behaviors that should either happen or not happen during the execution of the system. They must guarantee the safety, deadlock-free, and liveness of the system.



**FIGURE 4** CPS non-functional properties

**TABLE 2** Properties of CPS

Properties	References	Modeling Languages
<b>Time-related properties</b>		
Duration	41,47,49,56,64,99,100,111	Formal-based models, UML profiles, PDDL
Start and End time	41,100	UML profiles, Formal-based models
Periodicity	56,99	Formal-based models
Communication time/transmission time	21,47,60	Formal-based models, UML profiles, Modelica
Worst case execution time & Deadline	47,56,64,75,99	Formal-based models, UML profiles, Multi-modeling technique
<b>Physical properties</b>		
Spatial properties	65,76,106	Multi-modeling technique, Formal-based models
Spatio-temporal physical properties	57,58,114,115	Formal-based models
Device-related	49,51,52,59,90,104,111	UML profiles, PDDL, Formal-based models,
& physical entity's state properties	9,25,48,60,61,63,64,65,67,101,116,117	ADL, Multi-modeling technique, BPMN
Cost-related properties (energy and price)	59,65,112,113	Formal-based models
Environmental and context properties	25,49,51,52,59,61,65,104,111,116,117	PDDL, Formal-based models, AADL, BPMN
<b>Behavioral properties</b>		
Safety	56,58,63,76,98,99,118,119	Multi-modeling technique, Formal-based models
Liveness	54,87,98,100	Multi-modeling technique, Formal-based models
Deadlock	54,56	Formal-based models

We realized that in CPS modeling and verification approaches, the designers specify behavioral properties. These properties express safety requirements,<sup>98</sup> guarantee liveness,<sup>87</sup> and avoid deadlock.<sup>56</sup> The non-satisfaction of these properties often causes unnecessary cost and may lead to catastrophic results in critical concurrently executing systems.

Figure 4 summarizes the different CPS properties studied in the literature.

Several researchers present methods to address the issues of representing the CPS properties. Based on the classification presented previously (ie, time-related, physical and behavioral properties), in Table 2, we present an attempt to compare existing research approaches with regard to their supported properties. In addition, we indicate the different modeling techniques used to specify each property. Accordingly, we notice, as a first observation, that formal methods are the most used to specify the non-functional properties in CPS modeling. In particular, Modelica is mostly related to specify the communication time of the interactive behavior. Besides, Modelica is used to model some physical properties while using different multi-modeling techniques since it is intended mostly to specify physical systems. We also notice that some approaches extend some ADLs and BPMN to specify some physical properties such as device-related and environmental properties. UML profiles such as MARTE are mostly defined to specify time-related properties. However, UML profiles are barely defined to support physical properties. PDDL and the multi-modeling technique are used to specify a limited set of time related and physical property. Finally, the behavioral properties that must be guaranteed (eg, deadlock-free) are supported by formal languages.

## 5 | APPLICATIONS OF CYBER-PHYSICAL SYSTEMS

Nowadays, CPS are often concerned by different domains such as transportation, smart cities, health-care, emergency rescue, etc.<sup>120</sup> In this section, we discuss some of the studied applications from different domains and their related properties.

### 5.1 | Transportation systems

Building a modern intelligent transportation system goes way beyond building the mechanical components. The safety and efficiency aspects are very important in these systems, and as such, they are a popular research topic. In a smart environment, a moving device can exchange information through wireless or wired communication with other devices or services. The need of this communication is different from one example to another. It can be autonomous operation such as in a railroad and railway system,<sup>118</sup> to manage the entire systems,<sup>48</sup> to control freeway traffic,<sup>98</sup> to ensure the safety of passengers traveling in an autonomous truck,<sup>119</sup> or to guarantee the autonomy and the connectivity of devices like in vehicular ad-hoc networks.<sup>76</sup> Considering the integration of CPS and the aviation domain, which includes aeronautics and air transport, modeling such systems has been a popular research topic recently. In the work of Bak and Chaki,<sup>63</sup> the authors model a distributed group of communicating quad-copters to verify the collision avoidance. Drones are also increasingly used in our daily lives, to make a delivery, take photos, and access remote locations. For example, Amazon completed the first Amazon Prime Air delivery in December of 2016.<sup>121</sup> The shipment took only 13 minutes from order to delivery. The aim of this project is to dramatically reduce the time taken to deliver their products. Furthermore, other robots in the industrial environment are used for repetitive and pre-defined tasks. In this type of applications, the physical elements are controlled by cyber elements to achieve the overall goal. For example, Kilobots<sup>122</sup> and swarms as given in the work of Fitzgerald et al<sup>77</sup> are used to explore hazardous environments and collaborate through a controller.

In the transportation domain, designers specify the different case studies while considering their mobility property. In addition, studies in this domain focus mostly on the physical, cooperative, and collaborative behavior of their elements. The aim of the specified models is to verify their correctness, safety (eg, road safety), traffic efficiency, and reachability (eg, accomplishing the tasks). Such real-time systems that depend on the time taken to perform activities require temporal reasoning. Specifying time-related properties at design-time allows studying the expected system behavior and verifying the requirements before reaching the execution step. In mobile CPS, executing the different tasks on time is very important for safety and efficiency reasons. In fact, the execution time depends on motion related characteristics such as the location (ie, spatial), the device characteristics like the velocity, and other environmental information such as the weather (sun/rain, windy day, etc). Hence, physical properties are as important as time-related properties when modeling this type of applications.

### 5.2 | Smart home/ smart cities

To make life easier and more comfortable, smart homes connect various devices that we interact with on a daily basis. For example, on your way home, you can simply use your smart-phone to tell your smart thermostat to start warming the house ahead of your arrival. Typically, these systems are developed to collaborate with other connected systems in order to control the physical world, to improve resource management and to improve the quality of life.

Rajhans et al<sup>87</sup> use a temperature control system to illustrate their behavioral modeling approach. Through modeling this system, they aim at maintaining the temperature of the room. Tan et al<sup>114</sup> introduce a smart space case study to help people with disabilities to live a better life. They use this example as an illustration to simulate a specified CPS event model. Seiger et al<sup>27</sup> present a process-based home automation system. The domestic environment is equipped with different actuators (eg, alarm clock, light, heater, etc) and sensors (eg, cameras, microphones, thermostat, etc) for measuring environmental or more complex data. Accordingly, using CPS in the smart home domain can increase users' comfort, reduce energy consumption, and secure the surroundings. We noticed that, for these reasons that while the studied approaches consider some physical properties, they do not focus on the timing behavior of the tasks. However, both types of properties need to be supported since they are dependent. Hence, modeling these systems at design-time and verifying their behavior and properties before reaching the execution step is important due to their complexity.

### 5.3 | Health-care

In the medical domain, CPS connect monitoring devices such as heart rate monitors and delivery devices such as medication infusion pumps in order to build intelligent systems of interconnected medical devices. Coordinating these medical devices can help more precisely diagnose health conditions in an intelligent manner by collecting information, ensuring patient safety, and improving treatment.

Banerjee et al<sup>25</sup> propose an approach to build behavioral models of Body Area Networks (BANs). BAN is a network of medical devices implanted within the human body. Their aim is to perform early evaluations of these models in order to capture the undesirable side effects of the medical device actions on the human body. In the work of Li et al,<sup>64</sup> the authors present an approach that enables the modeling of a laser tracheotomy medical device plug-and-play<sup>123</sup> that interlocks various medical devices. Based on a hybrid automata formalism, the designer specifies the behavioral models of each element in the system and expresses safety hazard conditions. The aim of the behavioral models verification at design-time is to increase safety and prevent future faults. In the same context, with distributed CPS, in the work of Banerjee and Gupta,<sup>58</sup> the authors model multi-channel infusion pumps as a demonstration of the extended spatio-temporal hybrid automata. These distributed nodes (ie, infusion pumps) often perform concurrent infusions that cause accumulated drug effecting certain parts of the body. Thus, ensuring the safety of such systems at earlier stage is required. The aim of this approach is to analyze the behavior of these elements in space (ie, body) and time.

It is obvious that today's smart medical systems are the integration of different computational services and medical devices. The computational services can perform various operations in real-time, while the different medical devices operate on or near the human body. Since these applications are critical and impact human life, there is a strong need to ensure the efficiency, safety, reliability, and correctness. It is also important to maintain the right timing of the behavior of such applications.

### 5.4 | Emergency rescue systems

In the real world, a disaster occurs in various forms such as an earthquake, tornado, fire, hurricane. After a disaster warning, the emergency rescue systems can connect cyber-physical technologies to save lives, reduce cost, rescue people, and attend to their critical needs.

Huang et al<sup>50</sup> use an earthquake scenario to illustrate a modeling approach based on an extended ontology for CPS. This scenario is a collaboration between some sub-tasks that include the transportation of injured victims, scanning the areas and searching for survivors. Marrella et al<sup>28</sup> consider another application scenario that comes from a real train derailment response system. In this system, the robots and humans communicate together. Humans, who are equipped with mobile devices to enable communication with robots, are designated to extinguish fire, evacuate people and take pictures. During the execution of these tasks, robots must remove debris from specific locations and coordinate with humans. According to Zhu et al<sup>51</sup> time in such applications is crucial because the first 24 hours is the most critical time for the mission.

Table 3 presents a classification of the studied works according to their application domain. In addition, we indicate the different modeling techniques used in each application domain. According to Table 3, we notice that formal methods are the most used to specify the different application domains. We also note that some approaches are generic and do not focus on a specific application domain.

**TABLE 3** Application domains of CPS

Applications Domains	References	Modeling Techniques
Automotive Cyber-Physical Systems (ACPS)	4,47,48,77,84,85,94,96,106	ADLs, UML profiles, Multi-modeling technique,
Mobile CPS (MCPS)	57,59,63,76,86,98,119,122,124,125	Formal-based models
Smart home & Smart area	27,28,61,87,90,104,114,126	Formal-based models, ADLs, BPMN, UML profiles, PDDL
Health-Care	4,25,56,58,64,65,99,102,104	ADLs, PDDL, Formal-based models
Emergency rescue	49,51,52,100,104	Formal-based models, PDDL

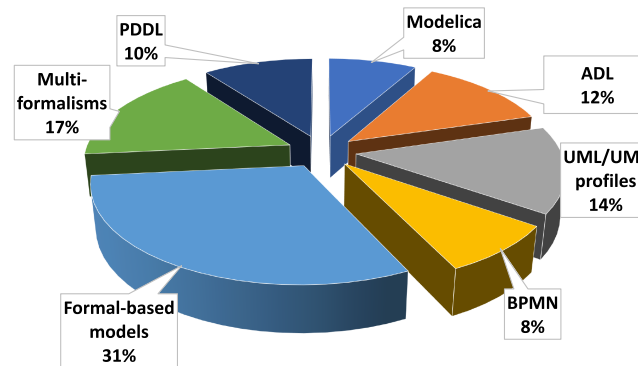
## 6 | DISCUSSION AND RESEARCH CHALLENGES

Throughout this survey, we provided a representative overview of the major efforts in modeling CPS. In particular, we studied the different CPS modeling approaches and emphasized that the system's behavior depends not only on the designed activities to perform but also on other non-functional properties. Particularly, physical properties are usually time-dependent and can affect the timed behavior of physical tasks. Thus, despite the research community's efforts devoted to modeling such systems and their related properties, several prominent challenges remain open.

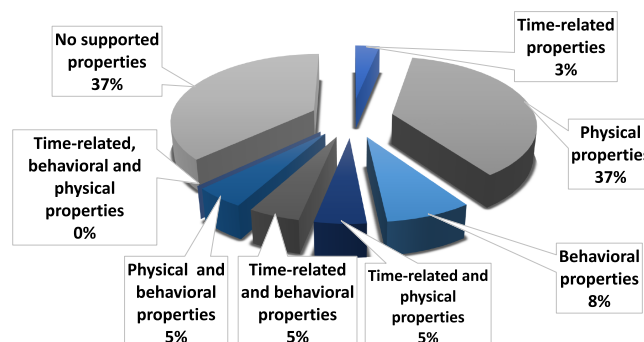
Designers need an appropriate modeling language and tool to specify the behavior of a cyber-physical application and its required non-functional properties at design-time. Such modeling language must allow to represent both the physical and cyber parts of such systems. It also needs to support the expected behavior as actions performed in the physical environment. Additionally, the CPS model requires the consideration of information about the involved entities and the real-world environment. This information is necessary at design-time to verify and validate the designed model, and at runtime for monitoring and reconfiguration.

Figure 5 illustrates the repartition of used languages and formalisms based on 50 studied papers on modeling CPS. We noticed that formal formalisms are the most used to model CPS. Formal models have the advantage to enable formal verifications. However, to design CPS, it is necessary to support specification languages to design. In order to ensure that the specification of a cyber-physical system's process model is easy to understand, intuitive to read, and presents a detailed view of the different activities in the process, BPMN is the best option. Despite this benefit of using BPMN it is the least used to model CPS, based on the examined papers. This due to the fact that BPMN lacks the capability required to represent some CPS aspects. Therefore, using BPMN for CPS is new emerging research field in which BPMN needs to be extended in order to be suitable for CPS process modeling.

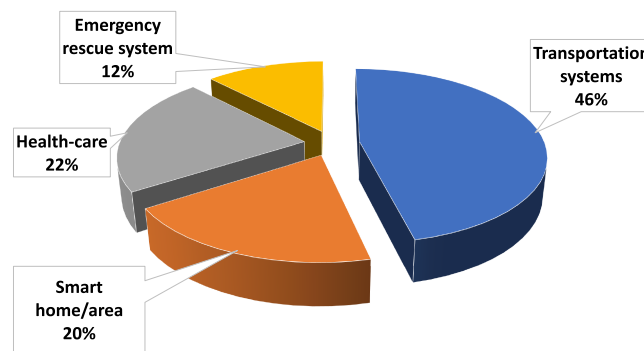
In order to answer the following question: "Which non-functional properties are specified when modeling CPS?," Figure 6 shows a symbolic and visual representation of the percentage of the supported non-functional properties in the 62 papers that we surveyed. We notice that 13% of the approaches have supported some time-related properties. In CPS, satisfying such requirements is related to the correctness of the system especially within critical applications. For example, in healthcare systems, executing an action on time can prevent disastrous consequences. Hence, we consider that 13% of the studied papers supporting some time-related properties is a small number. In addition, not all the time-related properties have been taken into consideration like temporal-dependencies between the occurrences of different activities. Supporting these properties when modeling such systems remains a challenge. 40% of the reviewed papers consider some physical properties. Compared to time-related properties, the researchers focus more on studying physical-related aspects and requirements when modeling CPS. Considering these properties is the major challenge to be addressed when specifying such systems. Moreover, most of the physical properties have an impact on the CPS behavior and can accordingly affect the timing of the actions. In particular, the critical applications are expected to be executed on time, always available when needed,



**FIGURE 5** Modeling techniques for cyber-physical system



**FIGURE 6** Non-functional properties in CPS modeling approaches



**FIGURE 7** Application domains in the CPS modeling approaches

and provided with a correct behavior under some specified conditions. These are other important challenges to consider when thinking about the dependencies between the different specified time-related and physical properties and modeling the behavior of such systems. Despite that fact, based on the 62 selected papers, few research attempts cover the specification of both types of properties (ie, 5%). For example, in the work of Huang et al,<sup>111</sup> the proposed approach does not ensure the modeling of a large set of required time-related and physical properties, and no verification techniques have been developed to deal with such models. In addition, behavioral properties such as safety and deadlock have been specified to verify the behavior of the modeled system. Only 18% of the proposed approaches define and guarantee relevant properties. We noticed that the majority of the studied papers do not yet support non-functional properties when modeling CPS (ie, 37% of the approaches).

CPS operate at a very large scale of real-world scenarios such as smart transportation, smart buildings, smart environment, healthcare and emergency rescue. Figure 7 answers the following question: “*what are the studied application domains of CPS?*” We notice that transportation systems are the most studied applications based on 41 selected papers that discuss specific application domains. This is understandable since these scenarios play a major role in our everyday lives so the integration of cyber and physical elements makes life easier, more secure, and more efficient on the road. Even though the chart shows that the healthcare and emergency rescue applications are not as widely studied to be modeled at design-time, these systems are equally as important as transportation systems. We conclude that in order to model the CPS, the designer needs a multi-domain modeling technique.

Modeling cyber-physical systems has been a topic of intense research recently. Through this critical analysis presented within this survey paper, we noticed that this emerging research field still faces a multitude of challenges; We identified the main CPS processes modeling challenges and requirements as follows:

- Future CPS modeling work needs suitable abstractions for intuitive modeling of the CPS behavior (eg, processes) and their real-world aspects (ie, the devices and the physical entities affected by the execution of a physical actions). These models need to be intuitive to read, easy to understand, present a detailed view of the CPS behavior, represent concurrent behavior, and target technical and non-technical stakeholders. The main challenge is to enable the designer to specify the CPS processes using a multi-domain modeling approach.
- CPS are heterogeneous, in the sense that they combine various aspects relying on both physical and cyber world abstractions. Due to the increasing integration of cyber capabilities in the physical world, there is a need to develop new design methodologies that provide rich extensions of CPS-specific functionality (ie, different types of cyber and physical activities) and enable seamless models of integrated activities (ie, the integration between cyber and physical behaviors).
- The behavior of CPS elements can be seen as collection of functionalities in the cyber, physical or control part of the system. In the context of CPS modeling there is a need to support both physical and cyber parts. To ensure the communication between cyber and physical parts and the collaboration of their different functionalities, the control part allows the reception of the monitored data, makes control decisions to find the needed services and sends the instructions to the physical devices.<sup>4,127</sup>
- Cyber and physical functionalities are the key concepts of cyber-physical systems. In addition, entity-based/resource-based concepts associated to CPS entities providing the functionalities are as important. The physical device can be a mobile or a static device. This characteristic is also important because we take into account different types of information with a moving behavior. Consequently, information about the affected and used entities, resources or devices need to be supported when modeling CPS. However, existing process modeling capabilities provided through the modeling notation are considered insufficient.
- Working with abstract models that omit details has major advantages in research but such methodology is not effective when studying cyber-physical systems. At an abstract level, the designer needs also to consider assigning non-functional properties as restrictions on the CPS behavior, especially with critical applications where malfunctions can threaten human lives. In different CPS, ensuring the correctness of their timing behavior is important. Fulfilling this need will require an efficient modeling method and a tool that enable the specification of time-related properties. These properties are the constraints on the timing of a single activity (ie, temporal property related to an activity such as the duration, start time, end time and recurrent behavior), time-dependent occurrence between activities (ie, the temporal-dependency and cycle), and timing of the collaborative behavior (ie, the global properties and communication time). In addition, the physical objects are not static,

in several scenarios. Their characteristics and states can change from one available instance to another. They can also change during their execution and can affect the timing behavior of an action. Thus, specifying physical properties (eg, device-related, spatial, context properties, etc.) is required at design-time. So another challenge, when modeling CPS, is developing common terminologies and semantics for describing physical and behavioral properties that can be time-dependent.

- A specified CPS processes model at-design time needs to be verified. Then, the challenge is to propose a future verification purpose of the consistency of the entire model. The consistency of the model is the ability to successfully execute the processes while respecting all the non-functional properties defined in this model. Hence there is a need to transform the behavioral model to a mathematical model with the goal of specifying the structural, the non-functional properties and the dependencies between them as differential or algebraic equations.<sup>128</sup>

## 7 | CONCLUSION

In this paper, we took on the problem of CPS requirements at design time by conducting a comprehensive survey of 62 research papers focused on this topic. Based on these requirements, we reviewed several research works in the modeling of CPS. First, we discussed the available modeling languages and techniques for specifying CPS. Then, for each of the surveyed works, we extracted the non-functional properties from the proposed modeling approach. Then, we discussed different application domains and some specific scenarios studied in these papers to highlight the importance of the CPS requirements. Finally, we highlighted potential research challenges that need to be addressed at design-time. We concluded that we need generic modeling methods that allow the specification and verification of the CPS behavior and their non-functional properties. These tools need to be targeted to different participants in the development process, in order to bridge the gap between process intention and implementation. We believe that finding solutions to the outlined challenges will dramatically improve the development process of cyber-physical systems and greatly improve their correctness and effectiveness at run-time. Besides modeling of CPS, several challenging problems such as self-healing and self-adaptation remain open. In this context, we plan to extend this study to analyze the existing works on this topic. This will enable to understand the existing problems and propose effective adaptation solutions.

## ORCID

Slim Kallel  <http://orcid.org/0000-0002-2824-167X>

## REFERENCES

1. Yu X, Xue Y. Smart grids: a cyber-physical systems perspective. *Proc IEEE*. 2016;104(5):1058-1070.
2. Wan J, Chen M, Xia F, Li D, Zhou K. From machine-to-machine communications towards cyber-physical systems. *Comput Sci Inf Syst*. 2013;10(3):1105-1128.
3. Mišić VB, Mišić J. *Machine-to-Machine Communications: Architectures, Technology, Standards, and Applications*. Boca Raton, FL: CRC Press; 2014.
4. Hu L, Xie N, Kuang Z, Zhao K. Review of cyber-physical system architecture. In: *Proceedings of the IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW)*; 2012; Shenzhen, China.
5. Derler P, Lee EA, Sangiovanni-Vincentelli AL. Addressing Modeling Challenges in Cyber-Physical Systems. Technical Report No. UCB/EECS-2011-17. Berkeley, CA: University of California, Berkeley; 2011.
6. Fritzson P. Modelica — A cyber-physical modeling language and the OpenModelica environment. In: *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC)*; 2011; Istanbul, Turkey.
7. Yu C, Jing S, Li X. An architecture of cyber physical system based on service. In: *Proceedings of the International Conference on Computer Science Service System (CSSS)*; 2012; Nanjing, China.
8. Graja I, Kallel S, Guermouche N, Kacem AH. BPMN4CPS: A BPMN extension for modeling cyber-physical systems. In: *Proceedings of the IEEE International Conference on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE)*; 2016; Paris, France.
9. Jensen JC, Chang DH, Lee EA. A model-based design methodology for cyber-physical Systems. Paper presented at: 2011 7th International Wireless Communications and Mobile Computing Conference; 2011; Istanbul, Turkey.
10. Lee EA. CPS foundations. In: *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*; 2010; Anaheim, CA.
11. Broman D, Lee EA, Tripakis S, Törngren M. Viewpoints, formalisms, languages, and tools for cyber-physical systems. In: *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling*; 2012; Innsbruck, Austria.
12. Hehenberger P, Vogel-Heuser B, Bradley D, Eynard B, Tomiyama T, Achiche S. Design, modelling, simulation and integration of cyber physical systems. *J Comput Ind*. 2016;82(C):273-289.
13. Khaitan SK, McCalley JD. Design techniques and applications of cyberphysical systems: a survey. *IEEE Syst J*. 2015;9(2):350-365.
14. Lohmann D, Schröder-Preikschat W, Spinczyk O. Functional and non-functional properties in a family of embedded operating systems. In: *Proceedings of the International Workshop on Object-Oriented Real-Time Dependable Systems*; 2005; Sedona, AZ.
15. Kitchenham B, Charters S. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001. Durham and Keele, UK: Keele University and Durham University, UK; 2007.
16. IEEE. IEEE Xplore Digital Library. <http://ieeexplore.ieee.org/Xplore/home.jsp>
17. Springer. Springer Link. <https://link.springer.com/>
18. ACM. ACM Digital Library. <http://dl.acm.org/>



19. ScienceDirect. ScienceDirect. <http://www.sciencedirect.com/>
20. Liping C, Xiaoping W, Xiong G, et al. Modeling and simulating CAN-based cyber-physical systems in Modelica. In: Proceedings of the International Conference on Software Security and Reliability Companion; 2012; Gaithersburg, MD.
21. Henriksson D, Elmqvist H. Cyber-physical systems modeling and simulation with Modelica. In: Proceedings of the 8th Modelica Conference; 2011; Dresden, Germany.
22. Nazari S, Sonntag C, Engell S. A Modelica-based modeling and simulation framework for large-scale cyber-physical systems of systems. *Int Fed Autom Control*. 2015;48(1):920-921.
23. Franca RB, Bodeveix J-P, Chemouil D, Filali M, Rolland J-F, Thomas D. The AADL behaviour annex – experiments and roadmap. In: Proceedings of the 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS); 2007; Auckland, New Zealand.
24. Feiler PH, Gluch DP, Hudak JJ. The Architecture Analysis & Design Language (AADL): An Introduction. Technical Report CMU/SEI-2006-TN-011. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University; 2006.
25. Banerjee A, Kandula S, Mukherjee T, Gupta SKS. BAND-AiDe: a tool for cyber-physical oriented analysis and design of body area networks and devices. *J ACM Trans Embed Comput Syst*. 2012;11(S2):1-30.
26. Object Management Group. Business process model and notation. 2011. <http://www.omg.org/spec/BPMN/2.0/>
27. Seiger R, Huber S, Schlegel T. Toward an execution system for self-healing workflows in cyber-physical systems. *Softw Syst Model*. 2016;17(2):551-572.
28. Marrella A, Mecella M, Halapuu P, Sardina S. Automated process adaptation in cyber-physical domains with the SmartPM system. In: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA); 2015; Rome, Italy.
29. Seiger R, Huber S, Schlegel T. PROTEUS: an integrated system for process execution in cyber-physical systems. *Enterprise, Business-Process and Information Systems Modeling: 16th International Conference, BPMDS 2015, 20th International Conference, EMMSAD 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Proceedings*. Basel, Switzerland: Springer International Publishing AG; 2015:265-280.
30. Meyer S, Sperner K, Magerkurth C, Debortoli S, Thoma M. Internet of Things Architecture IoT-A Project Deliverable D2.2 - Concepts for Modelling IoT-Aware Processes. 2012. [http://www.meet-iot.eu/deliverables-IOTA/D2\\_2.pdf](http://www.meet-iot.eu/deliverables-IOTA/D2_2.pdf)
31. Meyer S, Ruppen A, Magerkurth C. Internet of Things-aware process modeling: integrating IoT devices as business process resources. In: Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE); 2013; Valencia, Spain.
32. Meyer S, Ruppen A, Hilty LM. The things of the internet of things in BPMN. Paper presented at: International Conference on Advanced Information Systems Engineering; 2015; Stockholm, Sweden.
33. Sperner K, Meyer S, Magerkurth C. Introducing entity-based concepts to business process modeling. Paper presented at: International Workshop on Business Process Modeling Notation; 2011; Lucerne, Switzerland.
34. Martinho R, Domingos D. Quality of information and access cost of IoT resources in BPMN processes. *Procedia Technol*. 2014;16:737-744.
35. Sungur CT, Spiess P, Oertel N, Kopp O. Extending BPMN for wireless sensor networks. In: Proceedings of the International Conference on Business Informatics (CBI); 2013; Vienna, Austria.
36. Tranquillini S, Spieß P, Daniel F, et al. Process-based design and integration of wireless sensor network applications. In: Proceedings of the International Conference on Business Process Management (BPM); 2012; Tallinn, Estonia.
37. Bock CE. SysML and UML 2 support for activity modeling. *J Syst Eng*. 2006;9(2):160-186.
38. Object Management Group. A UML profile for MARTE: Modeling and Analysis of Real-Time Embedded systems. 2008. <http://www.omg.org/omgmarte/Documents/Specifications/08-06-09.pdf>
39. Object Management Group. OMG Systems Modeling Language (OMG SysML). 2012. <http://sysml.org/docs/specs/OMGSysML-v1.3-12-06-02.pdf>
40. Thramboulidis K. A cyber-physical system-based approach for industrial automation systems. *J Comput Ind*. 2015;72:92-102.
41. Mallet F. MARTE/CCSL for modeling cyber-physical systems. *Formal Modeling and Verification of Cyber-Physical Systems: 1st International Summer School on Methods and Tools for the Design of Digital Systems, Bremen, Germany, September 2015*. Wiesbaden, Germany: Springer Fachmedien Wiesbaden; 2015:26-49.
42. Schamai W, Fritzson P, Paredis CJJ. Translation of UML state machines to Modelica: handling semantic issues. *SIMULATION*. 2013;89(4):498-512.
43. Chemnitz University of Technology. MechatronicUML. 2015. <https://www.tu-chemnitz.de/informatik/ST/research/projects/mechatronicuml>
44. de Silva CW, Behbahani S. A design paradigm for mechatronic systems. *Mechatronics*. 2013;23(8):960-966.
45. Guérineau B, Bricogne M, Durupt A, Rivest L. Mechatronics vs. cyber physical systems: towards a conceptual framework for a suitable design methodology. Paper presented at: 2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS)/17th International Conference on Research and Education in Mechatronics (REM); 2016; Compiègne, France.
46. Greenyer J, Rieke J, Schäfer W, Sudmann O. The Mechatronic UML Development Process. In: *Engineering of Software: The Continuing Contributions of Leon J. Osterweil*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2011:311-322.
47. Pohlmann U, Holtmann J, Meyer M, Gerking C. Generating Modelica models from software specifications for the simulation of cyber-physical systems. Paper presented at: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications; 2014; Verona, Italy.
48. Aziz MW, Rashid M. Domain specific modeling language for cyber physical systems. Paper presented at: 2016 International Conference on Information Systems Engineering (ICISE); 2016; Los Angeles, CA.
49. Huang J, Bastani F, Yen I-L, Jeng J-J. Toward a smart cyber-physical space: a context-sensitive resource-explicit service model; 2009; Seattle, WA.
50. Huang J, Bastani FB, Yen I-L, Zhang W. A framework for efficient service composition in cyber-physical systems. In: Proceedings of the International Symposium on Service Oriented System Engineering (SOSE); 2010; Nanjing, China.
51. Zhu W, Zhou G, Yen I-L, Bastani F. A PT-SOA model for CPS/IoT services. In: Proceedings of the International Conference on Web Services (ICWS); 2015; New York, NY.
52. Zhao Y, Dong J, Huang J, Zhang Y, Yen I-L, Bastani F. Abstract service for cyber physical service composition. *Service Life Cycle Tools and Technologies: Methods, Trends, and Advances*. Hershey, PA: Information Science Reference; 2012:303-322.
53. Krishna SN, Trivedi A. Hybrid automata for formal modeling and verification of cyber-physical systems. *J Indian Inst Sci Spec Issue Cyber Phys Syst*. 2015;93(3).

54. Wang P, Xiang Y, Zhang SH. Cyber-physical system components composition analysis and formal verification based on service-oriented architecture. Paper presented at: 2012 IEEE Ninth International Conference on e-Business Engineering; 2012; Hangzhou, China.
55. Venema Y. Temporal logic. *The Blackwell Guide to Philosophical Logic*. Oxford, UK: Blackwell Publishers; 2016. <https://staffscience.uva.nl/y.venema/papers/TempLog.pdf>
56. Pajic M, Mangharam R, Sokolsky O, Arney D, Goldman J, Lee I. Model-driven safety analysis of closed-loop medical systems. *IEEE Trans Ind Inform*. 2014;10(1):3-16.
57. Shao Z, Liu J, Ding Z, Chen M, Jiang N. Spatio-temporal properties analysis for cyber-physical systems. In: Proceedings of the 18th International Conference on Engineering of Complex Computer Systems (ICECCS); 2013; Singapore.
58. Banerjee A, Gupta SKS. Spatio-temporal hybrid automata for safe cyber-physical systems: a medical case study. Paper presented at: 2013 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS); 2013; Philadelphia, PA.
59. Ye L, Qian K, Zhang L. S-PDH: A CPS service contract framework for composition. *Service-Oriented Computing - ICSOC 2015 Workshops: WESOA, RMSOC, ISC, DISCO, WESE, BSCI, FOR-MOVES, Goa, India, November 16-19, 2015, Revised Selected Papers*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2016:79-90.
60. Basile F, Chiacchio P, Coppola J. A colored timed Petri Net model for a cyber-physical view of automated warehouse systems. In: Proceedings of the 21st International Conference on Emerging Technologies and Factory Automation (ETFA); 2016; Berlin, Germany.
61. Garro A, Tundis A, Bouskela D, et al. On formal cyber physical system properties modeling: a new temporal logic language and a Modelica-based solution. Paper presented at: 2016 IEEE International Symposium on Systems Engineering (ISSE); 2016; Edinburgh, UK.
62. Li G, Song D, Liao L, Sun F, Du J. Learning automata-based adaptive web services composition. Paper presented at: 2014 IEEE 5th International Conference on Software Engineering and Service Science; 2014; Beijing, China.
63. Bak S, Chaki S. Verifying cyber-physical systems by combining software model checking with hybrid systems reachability. In: Proceedings of the International Conference on Embedded Software (EMSOFT); 2016; Pittsburgh, PA.
64. Li T, Tan F, Wang Q, Bu L, Cao J-N, Liu X. From offline toward real time: a hybrid systems model checking and CPS codesign approach for medical device plug-and-play collaborations. *IEEE Trans Parallel Distrib Syst*. 2014;25(3):642-652.
65. Abbas H, Fainekos G, Sankaranarayanan S, Ivančić F, Gupta A. Probabilistic temporal logic falsification of cyber-physical systems. *J ACM Trans Embed Comput Syst (TECS)*. 2013;12(2s):1-30.
66. Eker J, Janneck JW, Lee EA, et al. Taming heterogeneity - the Ptolemy approach. *Proc IEEE*. 2003;91(1):127-144.
67. Lee EA. Constructive Models of Discrete and Continuous Physical Phenomena. Technical Report No. UCB/EECS-2014-15. Berkeley, CA: EECS Department, University of California, Berkeley; 2014.
68. Lee EA. Modeling concurrent real-time processes using discrete events. *Ann Softw Eng*. 1999;7(1-4):25-45.
69. Lee EA. The problem with threads. *Computer*. 2006;39(5):33-42.
70. Lee EA. Finite State Machines and Modal Models in Ptolemy II. Technical Report No. UCB/EECS-2009-151. Berkeley, CA: EECS Department, University of California, Berkeley; 2009.
71. Edwards SA, Lee EA. The semantics and execution of a synchronous block-diagram language. *Sci Comput Program*. 2003;48(1):21-42.
72. Lee EA, Parks TM. Dataflow process networks. *Readings in Hardware/Software Co-Design*. Norwell, MA: Kluwer Academic Publishers; 2002.
73. Lee EA, Matsikoudis E. The semantics of dataflow with firing. In: *From Semantics to Computer Science: Essays in Memory of Gilles Kahn*. Cambridge, UK: Cambridge University Press; 2007.
74. Zhao Y, Liu J, Lee EA. A programming model for time-synchronized distributed real-time systems. Paper presented at: 13th IEEE Real Time and Embedded Technology and Applications Symposium; 2007; Bellevue, WA.
75. Eidson JC, Lee EA, Matic S, Seshia SA, Zou J. Time-Centric Models for Designing Embedded Cyber-Physical Systems. Technical Report No. UCB/EECS-2009-135. Berkeley, CA: EECS Department, University of California, Berkeley; 2009.
76. Zhang L. Applying system of systems engineering approach to build complex cyber physical systems. *Progress in Systems Engineering: Proceedings of the Twenty-Third International Conference on Systems Engineering*. Basel, Switzerland: Springer International Publishing Switzerland; 2015:621-628.
77. Fitzgerald J, Pierce K, Larsen PG. Co-modelling and co-simulation in the engineering of systems of cyber-physical systems. In: Proceedings of the 9th International Conference on System of Systems Engineering (SOSE); 2014; Adelaide, Australia.
78. International Organization for Standardization. Information Technology - Programming Languages, Their Environments and System Software Interfaces - Vienna Development Method - Specification Language - Part 1: Base language. Technical Report ISO/IEC JTC 1/SC 22. Geneva, Switzerland: International Organization for Standardization; 1996.
79. Broenink JF. Modelling, simulation and analysis with 20-sim. *J A*. 1997;38(3):22-25.
80. Simko G, Lindecker D, Levendovszky T, Neema S, Sztipanovits J. Specification of cyber-physical components with formal semantics- integration and composition. *Model-Driven Engineering Languages and Systems: 16th International Conference, MODELS 2013, Miami, FL, USA, September 29 - October 4, 2013. Proceedings*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2013.
81. Simko G, Levendovszky T, Maroti M, Sztipanovits J. Towards a theory for cyber-physical systems modeling. In: Proceedings of the 4th ACM SIGBED International Workshop on Design, Modeling, and Evaluation of Cyber-Physical Systems; 2014; Berlin, Germany.
82. Porter J, Hemingway G, Nine H, et al. The ESMoL Language and Tools for High-Confidence Distributed Control Systems Design. Part 1: Design Language, Modeling Framework, and Analysis. Technical Report ISIS-10-109. Nashville, TN: Institute for Software Integrated Systems - Vanderbilt University; 2010.
83. Junjie T, Jianjun Z, Jianwan D, et al. Cyber-physical systems modeling method based on Modelica. In: Proceedings of the International Conference on Software Security and Reliability Companion (SERE-C); 2012; Gaithersburg, MD.
84. Darragi N, El-Koursi M, Collart-Dutilleul S. Architecture description language for cyber physical systems analysis: a railway control system case study. Paper presented at: 14th International Conference on Railway Engineering Design and Optimization (COMPRAIL); 2014; Rome, Italy.
85. Sun Z, Zhou X. Extending and recompiling AADL for CPS modeling. Paper presented at: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing; 2013; Beijing, China.
86. Bhawe A, Krogh BH, Garlan D, Schmerl B. View consistency in architectures for cyber-physical systems. Paper presented at: 2011 IEEE/ACM Second International Conference on Cyber-Physical Systems; 2011; Chicago, IL.

87. Rajhans A, Cheng S-W, Schmerl B, et al. An architectural approach to the design and analysis of cyber-physical systems. *Electron Commun EASST*. 2009;21:1-10.
88. Feng S, Zhang L. Model transformation for cyber physical systems. *Advances in Computer Science and Its Applications*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2014.
89. Magureanu G, Gavrilescu M, Pescaru D. Validation of static properties in unified modeling language models for cyber physical systems. *J Zhejiang Univ SCI C*. 2013;14(5):332-346.
90. Calvo I, Portillo E, Albéniz OG, et al. Towards an infrastructure model for composing and reconfiguring cyber-physical systems. *Ubiquitous Computing and Ambient Intelligence: 6th International Conference, UCAmI 2012, Vitoria-Gasteiz, Spain, December 3-5, 2012. Proceedings*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2012.
91. Becker S, Dziwok S, Gerking C, et al. The MechatronicUML method: model-driven software engineering of self-adaptive mechatronic systems. In: *Proceedings of the 36th International Conference on Software Engineering*; 2014; Hyderabad, India.
92. Bocciarelli P, D'Ambrogio A, Giglio A, Paglia E. A BPMN extension for modeling cyber-physical-production-systems in the context of industry 4.0. Paper presented at: 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC); 2017; Calabria, Italy.
93. Panfilenko D, Poller P, Sonntag D, Zillner S, Schneider M. BPMN for knowledge acquisition and anomaly handling in CPS for smart factories. Paper presented at: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA); 2016; Berlin, Germany.
94. Canedo A, Al Faruque MA, Richter JH. Multi-disciplinary integrated design automation tool for automotive cyber-physical systems. Paper presented at: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE); 2014; Dresden, Germany.
95. Ruchkin I. Architectural and analytic integration of cyber-physical system models. In: *Proceedings of the ACM Student Research Competition at the ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*; 2015; Ottawa, Canada.
96. Larsen PG, Fitzgerald J, Woodcock J, Nilsson R, Gamble C, Foster S. Towards semantically integrated models and tools for cyber-physical systems design. *Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications: 7th International Symposium, ISO/LA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part II*. Basel, Switzerland: Springer International Publishing AG; 2016.
97. Pereira F, Gomes L. Combining data-flows and Petri nets for cyber-physical systems specification. *Technological Innovation for Cyber-Physical Systems: 7th IFIPWG 5.5/SOCOLNET Advanced Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2016, Costa de Caparica, Portugal, April 11-13, 2016, Proceedings*. Basel, Switzerland: Springer International Publishing AG; 2016:65-76.
98. Coogan S, Arcak M. Freeway traffic control from linear temporal logic specifications. In: *Proceedings of the International Conference on Cyber-Physical Systems (ICCPs)*; 2014; Berlin, Germany.
99. Arney D, Pajic M, Goldman JM, Lee I, Mangharam R, Sokolsky O. Toward patient safety in closed-loop medical device systems. In: *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs)*; 2010; Stockholm, Sweden.
100. Qian Z, Yu H. A TAOPN approach to modeling and scheduling cyber-physical systems. In: *Proceedings of the International Conference on Information Science and Applications (ICISA)*; 2013; Suwon, South Korea.
101. Saeedloei N, Gupta G. A methodology for modeling and verification of cyber-physical systems based on logic programming. *ACM SIGBED Rev - Spec Issue First Workshop Declar Program Real-Time Cyber-Phys Syst*. 2016;13(2):34-42.
102. Sanwal MU, Hasan O. Formally analyzing continuous aspects of cyber-physical systems modeled by homogeneous linear differential equations. In: *Cyber Physical Systems. Design, Modeling, and Evaluation: 5th International Workshop, CyPhy 2015, Amsterdam, The Netherlands, October 8, 2015, Proceedings*. Basel, Switzerland: Springer International Publishing AG; 2015:132-146.
103. Feljan AV, Mohalik SK, Jayaraman MB, Badrinath R. SOA-PE: a service-oriented architecture for planning and execution in cyber-physical systems. In: *Proceedings of the International Conference on Smart Sensors and Systems (IC-SSS)*; 2015; Bangalore, India.
104. Yen I-L, Zhu W, Bastani F, Huang Y, Zhou G. Rapid service composition reasoning for agile cyber physical systems. In: *Proceedings of the IEEE Symposium on Service-Oriented System Engineering (SOSE)*; 2016; Oxford, UK.
105. Petnga L, Austin M. Ontologies of time and time-based reasoning for MBSE of cyber-physical systems. *Procedia Comput Sci*. 2013;16:403-412.
106. Petnga L, Austin M. Spatial ontologies and models for safety-critical cyber-physical systems. In: *Proceedings of the International Conference on Complex Systems Engineering (ICCSE)*; 2015; Storrs, CT.
107. Shao Z, Liu J. Spatio-temporal hybrid automata for cyber-physical systems. *Theoretical Aspects of Computing - ICTAC 2013: 10th International Colloquium, Shanghai, China, September 4-6, 2013. Proceedings*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2013.
108. Rasch K, Li F, Sehic S, Ayani R, Dustdar S. Context-driven personalized service discovery in pervasive environments. *World Wide Web*. 2011;14(4):295-319.
109. Khabou N, Rodriguez IB. Threshold-based context analysis approach for ubiquitous systems. *Concurr Comput Pract Exp*. 2015;27(6):1378-1390.
110. Rodriguez IB, Van Wambeke N, Drira K, Chassot C, Jmaiel M. Multi-layer coordinated adaptation based on graph refinement for cooperative activities. *Commun SWIN*. 2008;4:163-167.
111. Huang J, Zhang Y, Yen I-L, et al. Real-time service-oriented distributed governance. In: *Proceedings of the 6th World Congress on Services*; 2010; Miami, FL.
112. Yang H, Ha S. Modeling and power optimization of cyber-physical systems with energy-workload tradeoff. In: *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED)*; 2015; Rome, Italy.
113. Al Faruque MA, Ahourai F. A model-based design of cyber-physical energy systems. In: *Proceedings of the 19th Asia and South Pacific Design Automation Conference (ASPDAC)*; 2014; Singapore.
114. Tan Y, Vuran MC, Goddard S, Yu Y, Song M, Ren S. A concept lattice-based event model for cyber-physical systems. In: *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs)*; 2010; Stockholm, Sweden.
115. Tan Y, Vuran MC, Goddard S. Spatio-temporal event model for cyber-physical systems. In: *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*; 2009; Montreal, Canada.
116. Wang T, Cheng L, Zhang K. An efficient context-sensitive service composition framework for precise controlling in cyber-physical systems. Paper presented at: 2011 International Conference of Information Technology, Computer Engineering and Management Sciences; 2011; Nanjing, China.
117. Seiger R, Huber S, Heisig P, Assmann U. Enabling self-adaptive workflows for cyber-physical systems. *Enterprise, Business-Process and Information Systems Modeling: 17th International Conference, BPMDS 2016, 21st International Conference, EMMSAD 2016, Held at CAiSE 2016, Ljubljana, Slovenia, June 13-14, 2016, Proceedings*. Basel, Switzerland: Springer International Publishing AG; 2016:3-17.

118. Zhang L. Aspect-oriented modeling of railway cyber physical systems based on the extension of AADL. Paper presented at: 2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing; 2013; Zhangjiajie, China.
119. Kandula S, Mukherjee T, Gupta SKS. Toward autonomous vehicle safety verification from mobile cyber-physical systems perspective. *SIGBED Rev*. 2011;8(2):19-22.
120. National Institute of Standards and Technology. Cyber-physical systems. NIST. <https://www.nist.gov/el/cyber-physical-systems>. 2016.
121. Amazon. Amazon completes its first drone-powered delivery. Engadget. 2016. <https://www.engadget.com/2016/12/14/amazon-completes-its-first-drone-powered-delivery/>
122. Rubenstein M, Ahler C, Hoff N, Cabrera A, Nagpal R. Kilobot: a low cost robot with scalable operations designed for collective behaviors. *Robot Auton Syst*. 2014;62(7):966-975.
123. Kim C, Sun M, Mohan S, Yun H, Sha L, Abdelzaher TF. A framework for the safe interoperability of medical devices in the presence of network failures. In: Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs); 2010; Stockholm, Sweden.
124. Passarini RF, Farines J-M, Fernandes JM, Becker LB. Cyber-physical systems design: transition from functional to architectural models. *Des Autom Embed Syst*. 2015;19(4):345-366.
125. Zhang Z, Eyisi E, Koutsoukos X, Porter J, Karsai G, Sztipanovits J. A co-simulation framework for design of time-triggered automotive cyber physical systems. *Simul Model Pract Theory*. 2014;43:16-33.
126. Lin J, Sedigh S, Miller A. Modeling cyber-physical systems with semantic agents. Paper presented at: 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops; 2010; Seoul, South Korea.
127. La HJ, Kim SD. A service-based approach to designing cyber physical systems. In: Proceedings of the IEEE/ACIS 9th International Conference on Computer and Information Science (ICIS); 2010; Yamagata, Japan.
128. Derler P, Lee EA, Sangiovanni-Vincentelli A. Modeling cyber-physical systems. *Proc IEEE*. 2012;100:13-28.

**How to cite this article:** Graja I, Kallel S, Guermouche N, Cheikhrouhou S, Hadj Kacem A. A comprehensive survey on modeling of cyber-physical systems. *Concurrency Computat Pract Exper*. 2020;32:e4850. <https://doi.org/10.1002/cpe.4850>