Journal of Intelligent Manufacturing (2022) 33:1603–1638
https://doi.org/10.1007/s10845-021-01753-8

A systematic mapping of semi-formal and formal methods in requirements engineering of industrial Cyber-Physical systems

Farzana Zahid[1] · Awais Tanveer[1] · Matthew M. Y. Kuo[1] · Roopak Sinha[1]

Received: 28 April 2020 / Accepted: 24 February 2021 / Published online: 1 April 2021

Abstract
The requirements engineering of Industrial Cyber-Physical Systems is extremely challenging due to large system sizes, component heterogeneity, involvement of multi-discipline stakeholders and machines, and continuous evolution. Formal and semi-formal languages, techniques, tools and frameworks can assist by providing repeatable and rigorous structures for eliciting, specifying, analysing, verifying and maintaining requirements. Various approaches have been proposed, but a contemporary and comprehensive study providing a landscape of the state-of-the-art is currently missing. This article reports a systematic mapping study covering 93 primary studies published between 2009 and October 2020. We categorise surveyed studies by current research directions in the use of semi-formal and formal methods for Requirements Engineering phases for Industrial Cyber-Physical Systems. We also identify gaps in current research and develop a novel conceptual model capturing the relationship between available formalisms and Requirements Engineering activities. We find that extensive work has been carried out on the formal analysis and verification of safety and timings requirements. However, the use of semi-formal notations, works on key phases like requirements elicitation and management, and the adoption of industrial standards are largely missing. Moreover, we find no literature providing methods to handle privacy and trust requirements, which have become critical concerns in this area.

Keywords Formal methods · Industrial Cyber-Physical system · Requirements engineering · Semi-formal methods · Systematic mapping study

Introduction

Cyber-Physical Systems are intelligent embedded systems that have tight coupling between their physical environment and computational components. *Industrial* Cyber-Physical Systems (ICPS) are considered the driving force behind the fourth Industrial Revolution, where they promise a much-needed solution to the problem of developing increasingly complex and larger-scale systems faster. ICPS refer to the integration of large-scale physical processes, machines, computation, and networking components in an industrial environment (Colombo et al. 2014). ICPS span multiple disciplines, including chemical, mechanical, control and software engineering. Owing to the merging of virtual and physical worlds, ICPS feature geographically dispersed stakeholders, devices and computers, which differentiate them from traditional embedded systems. Such integrated systems also undergo continuous evolution and contain emergent behaviours that arise due to the long-term interaction between heterogeneous components and sub-systems. On the one hand, ICPS promise substantial opportunities in sectors like smart manufacturing, smart cities, intelligent transportation, real-time health care, smart grids, cyber defence, aerospace and water treatment, with positive impacts on value-chain contributors such as society, environment, humans, devices and the economy (Öztemel and Gursev 2020). On the other hand, evolutionary and dispersed requirements from a wide range of stakeholders, consider-

✉ Farzana Zahid
farzana.zahid@autuni.ac.nz

Awais Tanveer
awais.tanveer@aut.ac.nz

Matthew M. Y. Kuo
matthew.kuo@aut.ac.nz

Roopak Sinha
roopak.sinha@aut.ac.nz

[1] School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand

ation for emergent behaviour, the scale and heterogeneity of collaborating physical and cyber components, incorporating new business and technological models, new levels of business-human-machine-human and business-machine interactions, amongst many others, have made developing ICPS systems and software intrinsically complex and challenging (Colombo et al. 2014).

The high complexity of ICPS requirements has a profound effect on all system engineering activities of the V-process model (Krueger et al. 2011). Hence, Requirements Engineering (RE) of ICPS, which covers requirements elicitation, analysis, specification, verification and validation (V&V), and management (Loucopoulos and Karakostas 1995), is critical. Systematic RE allows for the ongoing interaction between *problem* domain requirements and *solution* domain requirements (Wiesner et al. 2015). Insufficient RE leads to inadequate or unclear requirements resulting in prohibitive development costs. In contrast to other contexts, RE in ICPS involves several unique challenges:

1. Organisational and social aspects of ICPS become just as important as technological concerns.
2. Requirements changes, originating from a large, dispersed and dynamic group of stakeholders, happen over long periods due to long-term system evolution.
3. Multi-site industrial processes (order, production, service delivery) feature their own methods, tools and models, which must be reused and incorporated into the overall ICPS.
4. Elicitation, V&V and management activities have a higher emphasis in ICPS because of globalisation, heterogeneity of products, domains and services constituting the system, and the collaboration of multi-discipline, multi-culture and multi-site stakeholders.
5. Requirements originating from innovations in the supply-chain cycle, newer industry standards, traceability of standards-based requirements, compliance and resilience must be included.

Current RE methods (languages, techniques, frameworks and tools) do not provide comprehensive and systematic support to deal with the unique challenges in ICPS. In general, RE methods can be informal, semi-formal or formal. *Informal methods* involve natural language requirements, which feature high flexibility and expressiveness but require considerable human effort due to ambiguity. Subsequently, they are difficult to automate (Colombo et al. 2014). *Semi-Formal Methods (SFM)* encompass notations and languages, such as UML, which feature precise syntax and common vocabularies to reduce ambiguity. However, they still require some human effort for interpreting their semantics and therefore, complete automation remains difficult (Ahmed and Robinson 2007). *Formal Methods (FM)* feature well-defined syntax and semantics and are used to enable the correctness by construction principle in safety-critical contexts. The principle aims to *"avoid introducing errors as far as possible, and remove those errors that are introduced, as soon as possible"* (Hall 2005). However, formal methods require specialized expertise and training and do not scale as well as informal or semi-formal methods (Guttag et al. 1993).

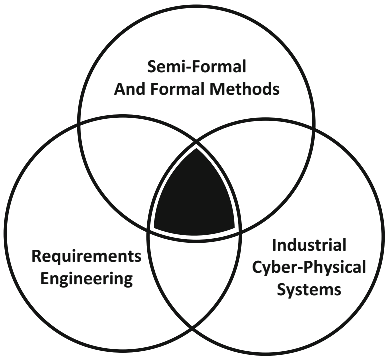
Semi-Formal And Formal Methods
Requirements Engineering
Industrial Cyber-Physical Systems

Fig. 1 Scope for this systematic mapping study, restricted to studies publishes from 2009

Systematic RE in ICPS requires semi-formal and formal methods because informal methods cannot be used reliably for such large-scale and complex systems. SFM and FM are complementary and can be integrated. For instance, SFM may be used to engineer non-critical aspects of an ICPS such as handling customer orders in manufacturing, while FM are better suited to handle safety-critical parts such as for functional safety in manufacturing (Hall 2005). A well-defined approach to RE using SFM and FM in ICPS is essential (Fisher et al. 2014). While various piecemeal solutions have been proposed in the literature, a holistic study of how their relevance to RE in ICPS is currently missing.

This paper presents a Systematic Mapping Study (SMS) for the identification and analysis of existing semi-formal and formal methods that have been used in the requirements engineering of ICPS. Fig. 1 shows the scope of this study, which, to the best of our knowledge (evidenced by an analysis of published secondary works in Sect. 2) has not been covered by any existing secondary study. A total of 3645 papers were identified out of which 93 primary studies were surveyed through the process described in Sect. 3. As presented in Sect. 4, we categorize the selected primary studies by quality (non-functional) requirements, application domains, research types and methods, applicable industrial standards, publication years, publication sources and venue types. A discussion on our findings appears in Sect. 5, which includes identifying critical gaps in current literature and a comparison of the formalisms used in surveyed works. This leads to the development of a conceptual model that can aid academic and industrial practitioners to understand and choose appropriate semi-formal and formal methods for the requirements engineering of their ICPS. Concluding remarks and future directions are discussed in Sect. 6.

Related works

A number of secondary studies that are somewhat related to our work are listed and compared in Table 1. These works were identified via the SMS process described later in Sect. 4.

Secondary studies such as Zheng et al. (2015); Wiesner et al. (2015); Zheng et al. (2015); Penzenstadler and Eckhardt (2012); Rashid et al. (2019); Simon et al. (2019) focus explicitly on cyber-physical systems, and others like Yu et al. (2019); Wiesner et al. (2017); Colombo et al. (2017); Wu et al. (2020) relate directly to ICPS. However, none of these studies carries out a structured SMS, categorises research, discusses the categorisation of quality requirements except in Simon et al. (2019) and considers formalisms except in Zheng et al. (2015). Zheng et al. (2015) perform a broad systematic literature review, a quantitative online survey involving 25 CPS researchers, and qualitative interviews with 9 CPS experts across four continents to uncover the existing approaches and practices for only formal verification and validation in CPS. Similarly, Simon et al. (2019) provide a guideline for performing RE in small and medium-sized enterprises and define the general requirements to be met by a system. Unlike this work, our study focuses on the mapping of both formal and semi-formal methods to all the activities of RE and to identify the quality requirements catered by selected primary studies.

Wiesner et al. (2015) elaborate on the challenges of RE, specifically for CPS in distributed environments. Zheng et al. (2015) present a study of developers on the topic of debugging of CPS. Colombo et al. (2017) and Yu et al. (2019) present a generic opinion and experience paper, respectively, to describe the fourth industrial revolution. They discuss the contribution, progress, world-wide emerging trends and challenges posed by ICPS. Likewise, Wiesner et al. (2017), discuss the general challenges for RE process in cyber-physical production systems. Wu et al. (2020) present a survey to examine the academic maturity of cyber-physical production systems. Our work extends these seminal works by providing a comprehensive landscape of how SFM and FM have been used in RE for ICPS with additional categorisations of the selected primary studies based on research type and quality requirements.

Lana et al. (2019) perform a systematic mapping study of formal and semi-formal languages and techniques for requirements modelling of software-intensives systems-of-systems. Penzenstadler and Eckhardt (2012) propose a requirements engineering content model tailored for cyber-physical systems, to be used for requirements elicitation and specification. In contrast, our work focuses on all RE activities.

A mapping study of all the requirements engineering activities in software ecosystems is presented in (Vegendla et al. 2018). This study shows how quality requirements are considered in the software ecosystem, and unlike our work, it neither categorises surveyed works based on formalisms and research type nor presents a conceptual model integrating its overall findings.

Studies like (Hachicha et al. 2019; Gabmeyer et al. 2019) deal with only formal verification in RE. In the case of generic modelling notations, Sepúlveda et al. (2016) discover 54 primary studies for software product lines. Wortmann et al. (2019) identify 408 publications to assess the use of modelling languages in Industry 4.0. For large-scale systems, Takbiri and Amini (2019) analyse different studies discussing large-scale requirements. None of these works focuses on requirements engineering related activities, quality requirements and formalisms.

A survey of formal requirements specification is presented in Sharma and Singh (2013). The paper evaluates specification languages such as object constraint, specification and description and Z languages. Unlike our work, they do not present any conceptual model and do not categorise their works on the quality requirements and research types.

A study by You et al. (2012) surveys formal methods employed for the development of software. Davis et al. (2013) survey the barriers faced by the USA government and private large systems manufacturers while adopting formal methods. They present their work generally instead of focusing on any particular domain. Furthermore, they do not provide categorisation by quality requirements, requirements engineering activities and research types.

Overall, this article provides a broader view for investigating and analysing the use of formal and semi-formal languages, techniques, tools and frameworks in each activity of RE in the context of ICPS. We identify the types of requirements targeted in the selected primary studies. Additionally, we contextualise our work to show the relationship between RE activities of ICPS and formalisms through an integrated conceptual model.

Systematic mapping study (SMS)

SMS is a multi-phase methodology consisting of planning, searching and reporting for identification, analysis and classification of existing literature in a particular domain, along with counting the contributions relating to the categories of that classification (BA and Charters 2007). The major purpose of a mapping study is to identify areas of activity within a relatively larger scope, as compared to systematic reviews. A systematic mapping, therefore, identifies the

Table 1 Comparison of our work with existing secondary studies by domains, requirements engineering activities, identification of quality requirements (Yes/No), formalisms (SFM/FM), categorization of research (Yes/No), research type (SMS/Survey/Opinion paper/Experience paper) and conceptual model (Yes/No)

References	Domains	RE activities	Formalism	Identify quality requirements	Categorize Research	Types of Research	Conceptual Model
(Simon et al. 2019)	Cyber-Physical System/ SMEs	All RE activities	×	Yes	No	Opinion paper	No
(Penzenstadler and Eckhardt 2012)	Cyber-Physical System/ SoS	Requirements Elicitation and Specification	×	No	No	Experience paper	Yes
(Rashid et al. 2019)	Cyber-Physical system	Requirements Verification	×	No	No	Survey	No
(Wiesner et al. 2015)	Cyber-Physical system	Generic (RE process)	×	No	No	Survey	No
(Zheng et al. 2015)	Cyber-Physical system	×	×	No	No	Opinion paper	No
(Zheng et al. 2015)	Cyber-Physical System	Requirements Verification and Validation	FM	No	No	Survey	No
(Colombo et al. 2017)	Industrial Cyber-Physical system	Generic	×	No	No	Opinion paper	Yes
(Wiesner et al. 2017)	Cyber Physical Production System	Generic (RE Process)	×	No	No	Opinion paper	No
(Wu et al. 2020)	Cyber Physical Production system	Generic	×	No	No	Survey	Yes
(Yu et al. 2019)	Industrial Cyber-Physical System	×	×	No	No	Experience paper	Yes
(Lana et al. 2019)	Systems-of-Systems	Requirements Modelling	FM, SFM	No	No	SMS	Yes
(Vegendla et al. 2018)	Software Ecosystems	All RE activities	×	Yes	No	SMS	No
(Hachicha et al. 2019)	Self-adaptive System	Requirements Verification	FM	Yes	No	Survey	No
(Gabmeyer et al. 2019)	Generic	Requirements Verification	FM	No	No	Survey	No
(Wortmann et al. 2019)	Industry 4.0	Generic (Modelling notations)	Generic	No	No	SMS	Yes
(Takbiri and Amini 2019)	Large-Scale Systems	Generic	×	No	No	Survey paper	No
(Sepúlveda et al. 2016)	Software Product Lines	Generic(Modelling Notations)	×	No	No	Survey	No
(Sharma and Singh 2013)	Generic	Requirements Specification	FM	No	No	Survey	No
(Davis et al. 2013)	Generic	×	FM	No	No	Survey	No
(You et al. 2012)	Generic	×	FM	No	No	Survey	No
Our Work	*Industrial Cyber- Physical System*	*All RE Activities*	*FM, SFM*	*Yes*	*Yes*	*SMS*	*Yes*

evidence base within the scope but does not delve into a qualitative evaluation. The SMS process involves the following steps: *planning*, *searching*, and *reporting*.

Mapping study protocol (Planning)

Planning contains the following steps.

Scope definition

As previously shown in Fig. 1, the scope of this research is the intersection of three areas: semi-formal and formal methods, requirements engineering and industrial cyber-physical systems. In addition, we confine the search to works published in the last decade. This time-frame, between 2009 and October 2020, comprehensively covers the coining of the term ICPS in the early 2010s and the intense research activity in the area that followed in subsequent years.

Formulation of research questions

The scope leads to the following research questions:

- RQ1. What are the current research directions within the use of SFM and FM for the RE in ICPS?
  - RQ1.1. Which RE activities have been studied in the literature in the context of ICPS?
  - RQ1.2. Which SFM and FM (languages, techniques, tools and frameworks) have been utilized for performing the RE activities identified in RQ1.1?
  - RQ1.3. Which system's software requirements (functional and/or quality) of ICPS have been targeted by selected studies, discovered in RQ1.1 and RQ1.2?
- RQ2. What approaches are used, in literature, to assess the applicability of primary studies, identified in RQ1?
  - RQ2.1. Which application domains are used to determine the applicability of selected primary studies?
  - RQ2.2. Which research methods and research types have been employed in the selected primary studies?
- RQ3. Referring to RQ1 and RQ2, what are the observed state-of-art contributions of selected primary studies?
  - RQ3.1. Which industrial standards have been adopted in identified primary studies?
  - RQ3.2. How can the identified primary studies be classified according to the publication years?
  - RQ3.3. What are the publication sources and venue types for the identified primary studies?

Keywords identification and search string

Following several methods reported in Petersen et al. (2015), we identified the following keywords: `formal*`, `semi-formal`, `semi formal`, `requirement`, `specif*`, `valid*`, `verif*`, `elicit*`, `analy*`, `document*`, and `manag*`. Similarly, for ICPS, we used different combinations of words like `cyber`, `industrial`, `physical`, and `production`. These keywords were joined with `OR` and `AND` operators to form the search string shown below.

```
(ALL (formal*)  OR  ALL ("semi formal")
OR ALL (semi-formal)  AND  ALL (requirement)
AND TITLE-ABS-KEY (elicit*)  OR
TITLE-ABS-KEY (analy*) OR
TITLE-ABS-KEY (specif*)  OR
TITLE-ABS-KEY (verif*)  OR
TITLE-ABS-KEY (valid*) OR
TITLE-ABS-KEY (document*)  OR
TITLE-ABS-KEY (manag*)  AND
ALL (industrial  AND cyber  AND
physical  AND system) OR
TITLE-ABS-KEY ("industrial cyber-physical
system")  OR  TITLE-ABS-KEY
(cyber-physical AND production AND system)
OR  TITLE-ABS-KEY ("cyber physical
production system")) AND  PUBYEAR  >  2008
```

It is important to note that the terms industrial cyber-physical systems and cyber-physical production systems are used interchangeably in literature.

Database selection

Franceschini et al. (2016); Dyba et al. (2007) advocate choosing four or five robust databases relevant to the field of research. In this study, Scopus, IEEE Xplore, ACM Digital Library and SpringerLink databases were selected. Elsevier's Scopus is one of the largest commercially available database of peer-reviewed articles which also indexes IEEE Xplore, ACM Digital Library and SpringerLink. However, we searched all of these databases individually to find any articles that may not have been indexed by Scopus. The search string, shown above, is in the Scopus format and was tailored as needed for all other databases searched.

Inclusion and exclusion criteria

The following inclusion criteria (IC) and exclusion criteria (EXC) were used to select only the relevant studies from the search results:

- IC1: Studies that investigated formal and/or semi-formal approaches for early- software development, i.e. at requirements engineering process of ICPS.
- IC2: Studies that focused only on research methods such as industrial case studies or industrial experimental work to assess and analyse their proposed solutions.
- IC3: Studies that were from computer science and software engineering or their sub-domains.
- IC4: Research studies that appeared since 2009 till October 2020.
- IC5: Studies published in conferences, journals, workshops, symposiums and technical reports, too.
- EXC1: Studies' whose title, keywords and/or abstract do not lie within the defined scope.
- EXC2: Studies that do not investigate any requirements engineering activity or FM/SFM related to ICPS.
- EXC3: Studies that address system-level or detailed design, unit testing, system integration testing, regression testing and acceptance testing.
- EXC4: Studies that do not provide an evaluation of proposed solutions.
- EXC5: Studies that do not include industrial applications.
- EXC6: Studies that address concepts such as networking/protocols, hardware, middleware, security requirements engineering, internet of things and cloud computing solely.
- EXC7: Studies that discuss challenges and problems in the targeted domain.
- EXC8: Studies that lack full text.
- EXC9: Books, thesis, secondary or tertiary studies, tutorial and opinion papers.
- EXC10: Studies not written in English.
- EXC11: Studies whose new version is available.
- EXC12: Duplicate articles found during search.

Searching

Search execution:

We used both automated and manual searching. *Automated systematic search* was conducted by following the protocol described in Sect. 3.1. The search execution chronology is shown in Fig. 2.

To find the potential primary studies, the initial search returned 3,645 articles using IC1 and IC3-IC5. Out of these, 40 articles were removed according to EXC11-EXC12 resulting in 3,605 studies. Secondly, a significant number of studies (1,478) were excluded by reading titles, keywords or abstracts (IC1 and EXC1) leaving 2,127 studies, and later by applying selection criteria IC2 and EXC6-EXC10, a further 1,000 studies were removed in order to select the related primary studies. The number dropped to 78 after a careful examination of introduction, conclusion and even full texts of each study (IC1 and EXC2-EXC5). In addition to automatic search, another searching technique called *snowballing* was also used to find the most relevant studies either through relevant references of each study or by using the citations related to already-included studies (Jalali et al. 2012). Forward or backward snowballing were used to identify and include another 15 relevant studies resulting in a total of 93 studies.

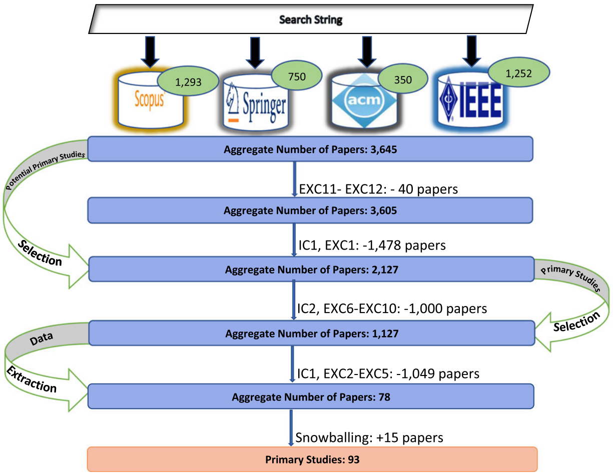
Search String
1,293
Scopus
750
Springer
350
acm
1,252
IEEE
Aggregate Number of Papers: 3,645
EXC11- EXC12: - 40 papers
Aggregate Number of Papers: 3,605
IC1, EXC1: -1,478 papers
Aggregate Number of Papers: 2,127
IC2, EXC6-EXC10: -1,000 papers
Aggregate Number of Papers: 1,127
IC1, EXC2-EXC5: -1,049 papers
Aggregate Number of Papers: 78
Snowballing: +15 papers
Primary Studies: 93
Potential Primary Studies
Selection
Primary Studies
Selection
Data
Extraction

Fig. 2 Search execution chronology

In manual searching, well-known sources such as the IEEE International Conference on Industrial Cyber Physical Systems, International Conference on Industrial Informatics, IEEE International Conferences on Software Testing, Verification and Validation Workshops, International Conference on Model Driven Engineering Languages and Systems, International Conference on Fundamental Approaches to Software Engineering, IEEE Transactions on Software Engineering, Springer's Requirements Engineering Journal, Springer's Formal Methods in System Design, IEEE International Conference of Requirements Engineering, IEEE International Workshop on Empirical Requirements Engineering were searched. Manual search helped reduce the probability of missing any relevant studies and increase confidence in the depth of the review.

Data extraction

Data extraction involves identifying keywords while reading abstracts together with the introduction and conclusion sections if the abstracts are not clear or well-written (Petersen et al. 2015). Open-coding enables the reviewer to allocate a 'code' or 'label' to terms or phrases, not necessarily according to their literal meaning but according to the concept behind the terms (Strauss and Corbin 1998). This underpins the creation of a categorical and conceptual schema of the text or data in focus. Keywords identification followed the

Table 2 Framework rubric for data extraction and classification

T1: Paper Title	T7. Publisher	T13. Languages used based on Formalism
T2. Authors	T8. Year of Publication	T14. Techniques, Frameworks and Tools
T3. Research Group/Organization	T9. Open Code and Keywords	T15. Types of Requirements
T4. Country	T10. Research Type	T16. RE Activities
T5. Pub. Venue	T11. Research Method	T17. Domains
T6. Pub. Type	T12. Formalisms	T18. Standards

three-pass method described in (Keshav 2007). According to this approach, deeper investigations are used to confirm the keywording, if an article's abstract, introduction or conclusion sections are ambiguous.

To address the research questions, we developed the rubric shown in Table 2, which employs both keywording and open-coding for data extraction. The data in Table 2 were analysed using text-mining techniques and maintained in MS-Excel to retrieve quality information, patterns and trends from the data set.

Felizardo et al. (2010) develop text mining tools, particularly for systematic mappings. Our data extraction rubric contains inherent features for implicit classification and categorisation. For tuple T9 of the rubric, the text-mining extension of Rapid Miner tool (Hofmann and Klinkenberg 2013) was applied on open-coded keywords and phrases. The purpose of the tool was to identify patterns, trends and facets within the research area. The outcomes of this analysis are discussed in detail in the following sections.

Conduction reporting (threats to validity)

Missing Important Relevant Studies: SMS is intended to cover the breadth of a research area, unlike systematic literature review where the main objective is to analyse the current work in the field, regarding quality. Moreover, it is not possible to guarantee the coverage of all relevant existing literature. Thus, to overcome this threat, the search string is devised in such a way that it returns the maximum amount of studies from online databases. Sometimes, title and abstracts can also misdirect a reader if they are inarticulate. To mitigate this issue, we also read the introduction, conclusion and in some cases, the internal sections of a study to find any missing information from titles and abstracts. Furthermore, only specific databases were chosen, so there is a possibility of missing relevant studies. To alleviate this matter, a manual search of publication venues was also performed.

Researcher Bias: Researcher bias is another factor that may have affected the validity. To reduce the biases, the study mapping protocol was established carefully with the consent of domain experts and co-authors.

Number of Selected Relevant Studies: The size of the set of primary studies depends on the scope, novelty of research domains or clear separation between concepts. For example, in terms of RE activities, there is a vague line between concerned terms which can be used interchangeably by authors. Consequently, the primary studies were selected carefully to identify only relevant works by going through full text and case-studies of each study. Lastly, whenever there was a doubt, domain experts were consulted.

Researchers' expertise: Researchers' expertise is also a threat when considering the research scope, shown in Fig. 1. Although, thorough knowledge of a particular area is mandatory, a researcher cannot be an expert in all aspects of a wide research area. This validity threat originates from the nature of systematic mapping process, which is wider and does not focus on qualitative aspects of the selected primary studies. To overcome this threat, researchers conferred with domain experts for feedback, advice, and exchange of ideas.

Results and findings

We summarise the key results and findings of the SMS.[1]

Current research directions

Current research directions can be determined by answering the following sub-questions:

- RQ1.1 Which RE activities have been studied in the literature in the context of ICPS?
- RQ1.2 Which SFM and FM (languages, techniques, tools and frameworks) have been utilized for performing the RE activities, identified in RQ1.1?
- RQ1.3 Which system's software requirements (functional and/or quality) of ICPS have been targeted by selected studies, discovered in RQ1.1 and RQ1.2?

[1] All data is available for download via https://github.com/FarzanaZahid/Systematic-Mapping-Study-.git in the form of an Excel worksheet.

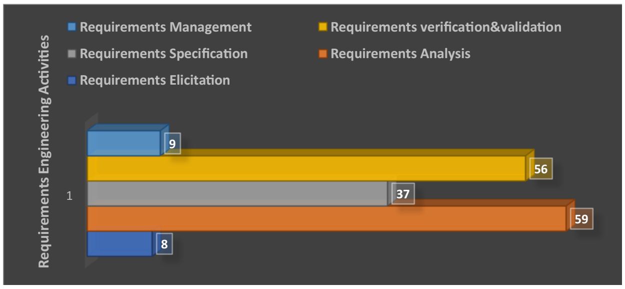

Fig. 3 Classification of primary studies according to requirements engineering activities

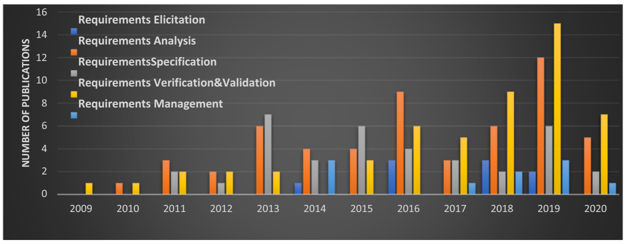

Fig. 4 Requirements engineering activities w.r.t publication years

RE activities in ICPS

The number of studies that stretch over different RE activities in ICPS is depicted in Fig. 3.

The figure shows that a single study can span over multiple activities, for instance (Mashkoor and Hasan 2012; Zhang et al. 2013; Askarpour et al. 2019) illustrate the use of FM on requirements analysis and V&V activities. A few target all RE activities, such as (Mancini et al. 2018). According to the results of mapping study, the most significant proportion of studies (59, 63%) cover requirements analysis followed by requirements V&V, covered by 56 (60%) studies. Requirements elicitation and management featured in only 8 (8.5%) and 9 (10%) studies, respectively.

The distribution of RE activities w.r.t publications and published years is depicted in Fig. 4. Overall, there is increasing attention in all activities of RE. V&V and analysis remain the dominant focus over the given period, except in 2009, where no work on requirements analysis was reported. By the end of the period (October 2020), only 5 research articles were found on requirements analysis while requirements V&V was featured in 7 studies. Studies on requirements elicitation and management started appearing from 2014 but these areas remain relatively unexplored.

Semi-formal and formal methods in ICPS RE

First, we categorised the 93 primary studies based on formalisms such as formal methods, semi-formal methods or integration of both (Both). Next, the existing formal and

Table 3 Languages and techniques for requirements elicitation

Type	Method	Primary studies
Language	Natural Language	(Denno and Blackburn 2014), (Adepu and Mathur 2016b), (Mühlfelder 2018), (Nuzzo et al. 2018), (Wang et al. 2019), (Loucopoulos et al. 2019)
Techniques	Prototype	(Seceleanu et al. 2017)
	Graphical Notation	(Chen et al. 2018)

semi-formal languages, techniques, frameworks or tools were plotted, solely, on identified RE activities (55%) while SFM are utilized in only 11 (11%) studies.

The classification of selected studies based on formalisms indicates that FM have been more the most studied concept. Out of 93 papers, 31 (33%) primary studies combined formal and semi-formal methods (both) in a single study. The numbers of primary studies that used FM are 51.

Requirements elicitation: Requirements elicitation is a challenging task in ICPS. Table 3 shows that only 8 (10%) studies address this activity: Denno and Blackburn (2014); Seceleanu et al. (2017); Adepu and Mathur (2016b); Nuzzo et al. (2018); Mühlfelder (2018); Chen et al. (2018); Wang et al. (2019); Loucopoulos et al. (2019).

In (Denno and Blackburn 2014), the product data sheet is used to capture and communicate the equipment requirements among the designers and suppliers. Other studies use natural language to capture the requirements and used either prototypes or graphical notations for further clarifications. As an illustration, a wind turbine industrial prototype model is used in (Seceleanu et al. 2017) to clarify the timings requirement of the wind turbine system. A semi-formal natural language requirement description model is proposed in (Wang et al. 2019) for elicitation of requirements.

Requirements analysis: Several formal and semi-formal languages, techniques, frameworks and tools have been used for performing requirements analysis activity.

A. Languages: Besides the activity of requirements specification, the reviewed primary studies used various formal and semi-formal specification languages during requirements analysis, too, as shown in Table 4.

These languages are: Architecture Analysis & Design Language (AADL) (Feiler and Gluch 2012), Modelica (Elmqvist et al. 1999), Alloy, Unified Modelling Language (UML)(France et al. 1998), System Modelling Language (SysML) (Mann 2009), Modeling and Analysis of Real-time and Embedded systems (MARTE) (Object Management Group 2011), Parallel Object-Oriented Specification Language (POOSL) (Theelen et al. 2007) and Process Algebra (PA)(Aceto et al. 2007) with its variants.

AADL is an architecture description language developed by SAE International. It is used to model hardware and software architecture of real-time embedded systems. Modelica is an object-oriented, equation-based programming language used to model complex physical system. As a result of the mapping process, we have found that AADL has been used in 10 studies. Among these, 5 selected studies (Zhang 2013e, b, c, d, 2014) integrate a Modelica with AADL, for specifying the cyber and physical part of automotive cyber physical systems, respectively. The reason for integration is that *"the descriptive power of AADL models with the analytic and computational power of Modelica models provides a capability that is significantly greater than provided by AADL or Modelica individually"*(Zhang 2013e). Likewise, Alloy is a specification language used to describe the structure in a software system. Thus, in (Adepu et al. 2016), it is used to model connectors between components and behavioural aspect of the secure water treatment model. Similarly, POOSL is a discrete-time modelling language and is used to developed embedded control software in (Nägele et al. 2019).

Process algebra (PA) deals with formal description or specification of concurrent processes. Algebra of Communicating Processes (ACP) and Security Process Algebra (SPA), belong to this family. ACP, along with its variants, are employed by six primary studies (Zhang et al. 2013; Sanwal and Hasan 2013; Adepu and Mathur 2016c; Goorden et al. 2019; Mancini et al. 2018; Rashid and Hasan 2020) and SPA is employed only in (Akella and McMillin 2009). ACP variants like second-order homogeneous linear differential equations are used in (Zhang et al. 2013) to model damped harmonic oscillation. An algebraic expression has been used in (Mancini et al. 2018) to split the requirements for the product line, and in (Sanwal and Hasan 2013), process in-variants are used to detect attacks in industrial control systems. Similarly, ACP is used to determine the functional requirement of ICPS in (Adepu and Mathur 2016c) and to analyse the smart grids in (Goorden et al. 2019). Moreover, in (Rashid and Hasan 2020), ACP is utilized to formally analyse continuous dynamics of CPS, which are then verified by the theorem prover. In (Akella and McMillin 2009), SPA is used to analyse security requirements of a natural gas transport system.

UML, SysML and MARTE are modelling languages used to model the system. SysML and MARTE are variants of UML. UML is used in eleven primary studies (Mancini et al. 2018; Wang et al. 2019; Loucopoulos et al. 2019; Zhang 2013e, c, 2014; Kulvatunyou et al. 2014; Bernardi et al. 2020; Li et al. 2016; Iglesias et al. 2017; Gomez et al. 2020). SysML is employed in (Zhang 2014; Neghina et al. 2019; Vogel-Heuser et al. 2014; Pagliari et al. 2019; Gomez et al. 2020; Gräßler et al. 2020). In (Zhang 2014), UML and SysML are integrated with Modelica ML (version of Modelica) to model the requirements of cyber-physical systems of systems. Likewise, Modelica, UML and SysML have integrated in Gomez et al. (2020) for the analysis of smart grid's functional and non-functional requirements. On the other side, four primary studies (Seceleanu et al. 2017; Ribeiro et al. 2016; Huang et al. 2019; Du et al. 2018) utilized MARTE along with its extensions.

B. Techniques: Formal methods based on the technique of Automata theory with their extensions are used by 12 primary studies (Seceleanu et al. 2017; von Birgelen and Niggemann 2018; Zhang 2013e; Kumar et al. 2012; Bu et al. 2011; Ger-

Table 4 Languages used in requirements analysis

Languages	Variants	Primary studies
AADL	-	(Zhang 2013e), (Zhang 2013b),(Zhang 2013c),(Zhang 2013d),(Zhang 2014),(Ruchkin et al. 2015),(Akkaya et al. 2016), (Lin et al. 2018), (Zhan et al. 2019), (Misson et al. 2019)
Modelica	-	(Zhang 2013e), Zhang (2013b),(Zhang 2013c), (Zhang 2013d), (Gomez et al. 2020)
	Modelica ML	(Zhang 2014)
Alloy		(Adepu et al. 2016)
POOSL		(Nägele et al. 2019)
PA	ACP	(Zhang et al. 2013), (Mancini et al. 2018), (Sanwal and Hasan 2013), (Adepu and Mathur 2016c), (Goorden et al. 2019), (Rashid and Hasan 2020)
	SPA	(Akella and McMillin 2009)
UML	-	(Mancini et al. 2018), (Wang et al. 2019), (Loucopoulos et al. 2019), (Zhang 2013e), (Zhang 2013c), (Zhang 2014), (Kulvatunyou et al. 2014), (Li et al. 2016), (Iglesias et al. 2017), (Gomez et al. 2020), (Bernardi et al. 2020)
	SysML	(Zhang 2014), (Neghina et al. 2019), (Vogel-Heuser et al. 2014), (Pagliari et al. 2019), (Gomez et al. 2020), (Gräßler et al. 2020)
	MARTE	(Seceleanu et al. 2017), (Ribeiro et al. 2016), (Huang et al. 2019) , (Du et al. 2018)

aldes et al. 2018; Ye-Jing et al. 2013; Balasubramaniyan et al. 2016; Wang et al. 2011; Lin et al. 2018; Du et al. 2018; Li et al. 2020), as presented in Table 5.

The development of modern-day computer science can be attributed to the work done using a theory of automata in the mid-20th century. A close relationship of automata theory with formal grammars makes it favourable to formally describe a model of a system as a state machine (Hopcroft et al. 2000). In eight primary studies (Seceleanu et al. 2017; von Birgelen and Niggemann 2018; Kumar et al. 2012; Bu et al. 2011; Geraldes et al. 2018; Ye-Jing et al. 2013; Balasubramaniyan et al. 2016; Wang et al. 2011), Hybrid Timed Automata (a version of automata) are used to model the behaviour of ICPS over time. While in Li et al. (2020) and Du et al. (2018), stochastic timed automata are used to represent timings and stochastic behavior in smart city and energy aware building. Cellular Automata is used by Zhang (2013e) for modelling and specification of the spatial-temporal requirements. Similarly, Probabilistic Deterministic Real-Time Automaton (PDRTA) is used with AADL to model the discrete events of the complex water treatment plant in (Lin et al. 2018).

Graph theory is another formal technique used in requirements analysis activity by Askarpour et al. (2019), LeMay et al. (2011), Knüppel et al. (2020) and Adepu and Mathur (2016a) for ICPS modelling. It is closely related to automata and mathematical logic. In fact, the author in (Sakarovitch 2009) defines a graph as a form of automata. For static modelling of topology for the smart city application domain, bigraphical technique is used in (Askarpour et al. 2019). An attack execution graph is used in a framework called ADversary VIew Security Evaluation (ADVISE) to find out the sequences, time, cost, probabilities of and other related information about each attack steps in (LeMay et al. 2011). Likewise, skill graphs in Knüppel et al. (2020) are used to identify the poorly defined safety requirements at an early stage of hybrid system development.

Abstract state machines are used in (Metsälä et al. 2017; Drozdov et al. 2019) to capture states of ICPS for security analysis and to model distributed control systems based on the IEC 61499 production engineering standard, respectively.

Formal Ontology is an implicit or explicit conceptualization of axioms in formal language. Three primary studies (Chen et al. 2018; Kulvatunyou et al. 2014; Sinha et al. 2015) used formal ontology for requirements analysis. The knowledge-base ontology is used in (Chen et al. 2018) to formalise the requirements as rules and relations in order to make it understandable for machines and humans. In addition to this, to describe the software components, functional ontology is identified in (Kulvatunyou et al. 2014) that is applied with use cases (UML diagram). A primary study by Sinha et al. (2015) converts requirements specified in natural language into formalised ontologies for further analysis. The general idea is to extend initial requirements ontology during subsequent phases of V-process model.

Formal contracts are based on invariants, pre and post conditions of software functions. Four studies (Zhang 2013d, 2014; Nuzzo et al. 2018; Westman and Nyberg 2014) concentrate on contracts for the formalisation of the rules. The Analysis contracts framework in (Zhang 2013d) is based on analysis-contracts that are used to analyse the security requirements for the water treatment system. In (Nuzzo et al. 2018), assume-guarantee (A/G) contracts are used to provide formal support to the high-level requirements in Contract-based Hierarchical Analysis and System Exploration (CHASE) framework. Additionally, Zhang (2014)

Table 5 Requirements analysis and formal techniques

Techniques	Variants	Primary studies
Automata	Hybrid Timed Automata	(Seceleanu et al. 2017),(Kumar et al. 2012),(Bu et al. 2011), (Geraldes et al. 2018),(Ye-Jing et al. 2013), (Balasubramaniyan et al. 2016), (Wang et al. 2011), (von Birgelen and Niggemann 2018)
	Cellular Automata	(Zhang 2013e)
	PDRTA	(Lin et al. 2018)
	Stochastic Timed Automata	(Li et al. 2020), (Du et al. 2018)
Graph Theory	Attack Execution Graph	(LeMay et al. 2011)
		(Askarpour et al. 2019),(Adepu and Mathur 2016a)
	Skill Graph	(Knüppel et al. 2020)
Abstract State Machine		(Metsälä et al. 2017), (Drozdov et al. 2019)
Formal Ontology		(Chen et al. 2018), (Kulvatunyou et al. 2014), (Sinha et al. 2015)
Formal Contract		(Zhang 2013d), (Zhang 2014),(Nuzzo et al. 2018), (Westman and Nyberg 2014)
VDM	VDM-RT	(Neghina et al. 2019)

and Westman and Nyberg (2014) defines the contracts for Simulink (Stateflow) diagrams and for structuring safety requirements in Fuel Level Display (FLD)-system respectively.

Vienna Development Method Real-Time (VDM-RT), used in (Neghina et al. 2019), is a formal method technique to model timings constraints of cyber physical production system. VDM-RT is a modified form of Vienna Development Method (VDM) that is one of the longest established formal method techniques to model industrial projects (Wang 2007).

Semi-Formal Techniques are employed by eleven primary studies (Wang et al. 2019; Loucopoulos et al. 2019; Kulvatunyou et al. 2014; Li et al. 2016; Iglesias et al. 2017; Neghina et al. 2019; Vogel-Heuser et al. 2014; Ribeiro et al. 2016; Gomez et al. 2020; Gräßler et al. 2020; Bernardi et al. 2020), as depicted in Table 6.

The studies (Wang et al. 2019; Iglesias et al. 2017) define the domain model showing the monitoring of software family for ICPS and patient control system. Likewise, meta-models are used in (Loucopoulos et al. 2019; Li et al. 2016; Ribeiro et al. 2016; Gräßler et al. 2020). For example, in (Li et al. 2016), meta-model is used to show the relationship between service and functions blocks of industrial assembly line system. A requirement diagram is used to model the functional model unit of USB stick production line in (Neghina et al. 2019) and to support use cases that have been applied to determine the functional and non-functional requirements of the smart grid in Gomez et al. (2020). SysML-AT (SySML extension), a specialized language profile to covers (non-) functional requirements, is adapted in (Vogel-Heuser et al. 2014) for automation, where SysML parametric diagrams models the components involved in the manufacturing process. It is noteworthy that diagrams used in semi-formal methods are considered as techniques in our work.

C. Frameworks: Table 7 shows the eleven primary studies that develop different domain-specific frameworks for requirements analysis. A formal attack model in (Adepu and Mathur 2016b) is used to determine the cyber-attacks on a water treatment system. The study by Loucopoulos et al. (2019) reports on the early Capability Oriented Requirements Engineering (e-CORE) framework for analysis and traceability of requirements for automobile manufacturing industrial-size case study where the association between assets is shown by the meta-model.

A Secure Modelling Framework (SeMF) was developed to analyse the security flaws in smart grid in (Fuchs et al. 2010). Likewise, a Unified Graphical Framework in (Zhan et al. 2019) is a graphical framework that is consists of AADL and Simulink (Stateflow) to model, simulate and validate ICPS. Real-Time Maude Framework in (Bae et al. 2015) is used to design a multirate distributed hybrid systems consisting of an airplane maneuvered by a pilot. In (Xu and Zhang 2013), the clock theory concept is used to analyse the tim-

Table 6 Requirements analysis and semi-formal techniques

Technique	Primary studies
Use-cases	(Kulvatunyou et al. 2014), (Gomez et al. 2020), (Bernardi et al. 2020)
Domain Model	(Wang et al. 2019), (Iglesias et al. 2017)
Meta-Models	(Loucopoulos et al. 2019), (Li et al. 2016), (Ribeiro et al. 2016), (Gräßler et al. 2020)
Requirements Diagram	(Neghina et al. 2019), (Gomez et al. 2020)
Parametric Diagram	(Vogel-Heuser et al. 2014)

Table 7 Frameworks for requirements analysis

Frameworks	Primary studies
Formal Attack Model	(Adepu and Mathur 2016b)
Contract-based Hierarchical Analysis and System Exploration (CHASE)	(Nuzzo et al. 2018)
Early Capability Oriented RE (e-CORE)	(Loucopoulos et al. 2019)
Analysis Contract Framework	(Ruchkin et al. 2015)
Unified Graphical Framework	(Zhan et al. 2019)
ADversary VIew Security Evaluation (ADVISE)	(LeMay et al. 2011)
Secure Modelling Framework(SeMF)	(Fuchs et al. 2010)
Real-Time Maude Framework	(Bae et al. 2015)
Clock	(Xu and Zhang 2013)
SKEDITOR	(Knüppel et al. 2020)
Surreal	(Bernardi et al. 2020)

ing requirements of different applications like steam boiler control system, press and railway cross system. Similarly in (Knüppel et al. 2020), a framework called SKEDITOR is proposed that combined formalised skill graphs and theorem prover, KeYmaera X, to analyse and verify safety requirements in a domain of transportation. A framework known as Surreal is developed in (Bernardi et al. 2020) where security and safety requirements of smart cars are analysed by misuse cases and verified by a model checker (nuSMV).

D. Tools: Simulation and co-simulation are used for requirements analysis in 12 studies, as shown in Table 8.

For simulation, Ptolemy II (Akkaya et al. 2016; Pagliari et al. 2019) models the performance of a delivery robotic system. Simulink is used in (Sanwal and Hasan 2013; Lin et al. 2018; Kang et al. 2018; Singh et al. 2019; Clarke and Zuliani 2011) for modelling transportation and manufacturing systems. Another simulation tool, AADLSim is used to model an Isollete system in (Zhan et al. 2019). TrueTime simulates performance requirementd of industrial mine pump in (Balasubramaniyan et al. 2016). Co-simulation can be performed either by tools or Functional mock-up Interface standard (FMI). FMI has been used in (Gomez et al. 2020) for multi-domain simulation whereas tools like 20-sim in (Nägele et al. 2019) and Overture in (Neghina et al. 2019) are used for co-simulation.

Requirements specification: Table 9 shows the various formal and semi-formal specification languages have been used in the selected primary studies. Common Algebraic Specification Language (CASL) is a specification language that is based on first-order logic with mathematical proofs. Researchers have also used other variants of first-order logic. CASL-First Order Logic is employed in four studies. Eleven studies use the temporal extension of CASL called CASL-Temporal Logic (CASL-TL).

Four studies use Linear Temporal Logic (LTL). LTL can express sequence or paths (Huth and Ryan 2004) of states of reactive systems over time. In (Clarke and Zuliani 2011), LTL's variant Bounded Linear Temporal Logic (BLTL) is used to express reliability properties of aircraft.

Signal Temporal Logic (STL), another variant of LTL, is reported in (Nejati et al. 2019) to specify real-time temporal operators and real-valued constraints for smart manufacturing. Similarly, another LTL extension called Metric Temporal-Spatial Logic (MTSL) is used in (Sun et al. 2015) to represent safety requirements of a train control system. Computation Tree Logic (CTL) is a branching-time logic which is employed by five studies. The timed extension of CTL called Timed Computation Tree Logic (TCTL) uses a clock variable to reason about system behaviours over time. TCTL is used to specify the timing and functional requirements of unmanned aerial vehicles in Misson et al. (2019), which were verified by the UPAAL model checker.

Table 8 Simulation and co-simulation tools used in requirements analysis

Tools	Types	Primary studies
Simulation	AADLSim	(Zhan et al. 2019)
	Simulink	(Sanwal and Hasan 2013), (Lin et al. 2018), (Kang et al. 2018), (Singh et al. 2019), (Clarke and Zuliani 2011)
	TrueTime	(Balasubramaniyan et al. 2016)
	PtolemyII	(Akkaya et al. 2016), (Pagliari et al. 2019)
Co-Simulation	20-sim	(Nägele et al. 2019)
	Overture Tool	(Neghina et al. 2019)
	FMI	(Gomez et al. 2020)

B Language and Z notation are two formal specification model-based languages based on set theory. These can be compared in terms of object orientation, concurrency, tool support and their industrial applications in (Kaur et al. 2012). Mashkoor and Hasan (2012); Zhang (2011) apply Z notation as a formal specification language. In (Mashkoor and Hasan 2012) Object-Z, which extends Z notation with object-oriented features, is used. Event-B, a derivation of B language, has been used in the study (Singh et al. 2019). AADL is used as a specification language in (Hissam et al. 2015; Ahmad et al. 2015). In (Zhang 2013a), an extension of AADL called AO4AADL is defined for the specification of aspect-oriented systems. Similarly, Multirate Synchronous AADL language is used in (Bae et al. 2015) for multirate synchronous systems. Another version of AADL is EAST-AADL, which is used for automotive embedded systems. It has been used in (Kang et al. 2018) whereas its probabilistic extension is utilized in (Kang et al. 2018). SMV is a state-based formal languages and is used in (Drozdov et al. 2017) to represent IEC 61499 model of a distributed automated cyber-physical system.

ICPS typically feature high concurrency. Communicating Sequential Processes (CSP) is a formal language based on calculus that allows modelling concurrency (Roscoe 1998). CSP has found its application in two primary studies (Zhang 2013c, a) in the form of Timed CSP (TCSP) to specify timing properties for ICPS in transportation.

Apart from traditional formal specification languages, some Domain-Specific Languages (DSLs) have been introduced over the years to accommodate domain-specific requirements formally. Although DSLs have limitations in applicability the solutions they create for a particular domain can be re-used in that specific domain. We find that four primary studies use DSL. A value specification language, variant of MARTE, is used to analyse the requirements engineering process of ICPS by using stereotypes and annotations in (Ribeiro et al. 2016) to model industrial packing ICPS. A language called ASLan++ Language, an extension of Aslan Formal Specification Language, is used in (Rocchetto and Tippenhauer 2017) which consists of assertions to identify different classes of attacks on water treatment system. Restricted Test Case Modeling (RTCM) language can be used to generate test cases based on formalised rules (Yue et al. 2015). In (Bouskela et al. 2017), a new language FORM-L is introduced to describe temporal constraints that closely relates to LTL.

Requirements V&V:

A. Languages: Table 10 shows the languages used in V&V. These include Hybrid CSP (HCSP), a variant of CSP, which is used in (Zhan et al. 2019) for verifying an Isolette system. Simulink models are translate into HCSP to carry of verification of a train control system in (Ahmad et al. 2015). Real-Time Maude (Ölveczky and Meseguer 2007) is a lan-

Table 9 Formal and semi-formal requirements specification languages

Languages	Variants	Primary studies
CASL-First Order Logic	Signal First Order	(Sanwal and Hasan 2013),(Menghi et al. 2019), (Nejati et al. 2019),(Cengic and Akesson 2010)
CASL-Temporal Logic (TL)	LTL	(Loucopoulos et al. 2019),(Kang et al. 2018),(Gawanmeh et al. 2017), (Grobelna 2020)
	LTL-BLTL	(Clarke and Zuliani 2011)
	LTL-STL	(Nejati et al. 2019)
	LTL-MTSL	(Sun et al. 2015)
	CTL	(Balasubramaniyan et al. 2016), (Meseguer and Ölveczky 2012), (Gawanmeh et al. 2017),(Wisniewski et al. 2020)
	CTL-TCTL	(Misson et al. 2019)
B Language	Event-B	(Singh et al. 2019)
Z Language		(Mashkoor and Hasan 2012), (Zhang 2011)
CSP	Timed CSP	(Zhang 2013c),(Zhang 2013a)
DSLs	Value Specification Language	(Ribeiro et al. 2016)
	ASL++ Language	(Rocchetto and Tippenhauer 2017)
	RTCM	(Yue et al. 2015)
	FORM-L	(Bouskela et al. 2017)
State-Based Languages	SMV	(Drozdov et al. 2017)
AADL		(Hissam et al. 2015), (Ahmad et al. 2015)
	AO4AADL	(Zhang 2013a)
	EAST-AADL	(Kang et al. 2018), (Kang et al. 2018)
	Multirate Synchronous AADL	(Bae et al. 2015)

Table 10 Languages, techniques and frameworks in requirements verification and validation

Methods	Types	Primary studies
Languages	HCSP	(Zhan et al. 2019), (Ahmad et al. 2015)
	Real-Time Maude	(Bae et al. 2015), (Meseguer and Ölveczky 2012)
	LTL-STL	(Nejati et al. 2019), (Nuzzo et al. 2019), (Ezio et al. 2019)
	LTL-MTL	(Li et al. 2020)
	DSL-ETL	(Bouskela and Jardin 2018)
Techniques	Formal Contract	(Nuzzo et al. 2019)
	CIPNs	(Wisniewski et al. 2020)
	State invariants	(Adepu and Mathur 2016c)
	Cause-effect graphing	(Kim et al. 2019)
	Knowledge based (quantitative fitness functions)	(Nejati et al. 2019)
	CPSDebug	(Ezio et al. 2019)
Frameworks	SOCRaTes	(Menghi et al. 2019)
	Unified Graphical Framework	(Zhan et al. 2019)
	Formal Framework	(Dang et al. 2016)
	Assume-Guarantee Contract Framework	(Nuzzo et al. 2019)

guage and tool for the formal specification and verification of real-time systems and is a dialect of Maude (Clavel et al. 2007). It is used in (Bae et al. 2015; Meseguer and Ölveczky 2012) to verify safety and timing requirements of avionics systems. Metric Temporal Language (MTL), a variant of LTL, is used in Li et al. (2020) to formally validate timing and performance requirements of emergency response missions in the smart cities. Likewise, a domain-specific language called Extended Temporal Language (ETL) is used in (Bouskela and Jardin 2018) to formally verify the timing requirements of traffic lights and used Modelica, as well, to simulate these requirements.

B. Techniques: Techniques like formal contracts in (Nuzzo et al. 2019), Control Interpreted Petri Nets (CIPNs) in (Wisniewski et al. 2020), CPSDebug in (Ezio et al. 2019), state invariants in (Adepu and Mathur 2016c) and a graph-based technique called cause-effect graphing is used in (Kim et al. 2019) for verification purposes, as shown in Table 10. A Petri Net is a form of bipartite graph and a formal mathematical modelling tool. Its graphical representation enables the visualization of state changes in a system during the runtime because of and they are used to model event-driven distributed computer systems (Wang 2007). In Wisniewski et al. (2020), CIPNs (variants of Petri nets) are used to determine the safety aspects of a beverage production and distribution system while its functional requirements are specified in CTL. A model checker then verifies these specifications. Model checking and theorem proving (Kallel et al. 2011) are formal verification methods which are implemented in model checkers and theorem provers. Furthermore, model checking is a technique in which a state-machine model of a system can be automatically analysed by an algorithm to verify whether the model satisfies requirements stated as temporal logic properties. In theorem proving, the correctness of system is proved by building mathematical proofs, often in a semi-automatic manner.

Requirements validation of ICPS requirements is performed by testing, as shown in Table 10. Test models in (Nejati et al. 2019) use Signal Temporal Logic (STL) and convert it into knowledge-based quantitative functions to determine failures in models based on a large number of sampled test inputs. In the same way, the CPSDebug technique presented in (Ezio et al. 2019) is used for testing the functional and fault tolerance requirements of ICPS specified in STL via CPSDebug testing tool (Bartocci et al. 2020).

C. Frameworks: Frameworks like Simulink Oracles for CPS Requirements with uncertainty (SOCRaTeS) (Menghi et al. 2019), Unified Graphical Framework (Zhan et al. 2019), Formal framework (Dang et al. 2016) and an Assume-Guarantee Contract Framework (Nuzzo et al. 2019) have been used for requirements V&V. In (Menghi et al. 2019), Signal First Order logic (SFOL) is used to specify requirements and then test oracles, specified in Simulink, are used to test the ICPS. Formal framework addresses requirements testing along with simulation in (Dang et al. 2016). In this framework, simulation traces are described in the form of a tree to map input signals to the output signals in an industrial HVAC system. The Assume-Guarantee Contract Framework presented in (Nuzzo et al. 2019) uses Stochastic Signal Temporal Logic (StSTL), a variant of STL language, to formalise the specifications and verify the functional and probabilistic requirements of an aircraft electric power distribution system by using the SCANS simulation tool.

Table 11 Models checkers and theorem provers employed in primary studies

Tools	Name	Primary studies
Model Checkers	SMV/nuSMV	(Drozdov et al. 2019),(Gawanmeh et al. 2017), (Bernardi et al. 2020)
	Zot	(Askarpour et al. 2019)
	QVTrace Tool	(Nejati et al. 2019)
	UPAAL	(Kumar et al. 2012),(Balasubramaniyan et al. 2016), (Seceleanu et al. 2017), (Kang et al. 2018), (Kang et al. 2018),(Kim et al. 2019), (Misson et al. 2019), (Huang et al. 2019)
	UPAAL-Statistical Model Checker	(Mancini et al. 2018), (Clarke and Zuliani 2011), (Kang et al. 2018), (Du et al. 2018), (Li et al. 2020)
	nuXmv	(Drozdov et al. 2017), (Grobelna 2020), (Wisniewski et al. 2020)
	Alloy Analyzer	(Adepu et al. 2016)
	CoPS	(Akella and McMillin 2009)
	MR-SynchAADL	(Bae et al. 2015)
	Spin	(Drozdov et al. 2019)
Theorem Prover	HOL4	(Mashkoor and Hasan 2012), (Sanwal and Hasan 2013)
	HOL Light	(Rashid and Hasan 2020)
	TVEC Theroem Prover	(Denno and Blackburn 2014)
	Hybrid Hoare Logic (HHL) Prover	(Nägele et al. 2019)
	KeYmaera X	(Garcia et al. 2019),(Knüppel et al. 2020)

Table 12 Requirements validation and domain-specific tools used in primary studies

V&V Tools	Tools Name	Primary studies
Requirements Testing Tools	MaTeLo	(Seceleanu et al. 2017)
	Test Oracle	(Menghi et al. 2019)
	Toucan4Test	(Yue et al. 2015)
	CPSDebug	(Bartocci et al. 2020)
Domain-Specific Tools	CHASE	(Nuzzo et al. 2018)
	CL-Atse	(Rocchetto and Tippenhauer 2017)
	HyPLC	(Garcia et al. 2019)
	xSAP	(Ferrante et al. 2017)

D. Tools: Several tools have been utilized for performing V&V, as shown in Table 11. Twenty-three primary studies used model checking and seven were found to employ theorem provers.

In addition to traditional formal verification methods, various requirements validation and domain-specific tools have been used in primary studies, as shown in Table 12.

For requirements validation purpose, tools such as MaTeLo in (Seceleanu et al. 2017), test oracle in (Menghi et al. 2019), CPSDebug in (Bartocci et al. 2020) and Toucan4Test in (Yue et al. 2015) are used. In four primary studies, (Nuzzo et al. 2018; Rocchetto and Tippenhauer 2017; Garcia et al. 2019; Ferrante et al. 2017), researchers implement domain-specific tools such as CHASE, CL-AtSe, HyPLC and xSAP for verification. Furthermore, different simulation and co-simulation tools, depicted in Table 13, are used by 12 primary studies.

Requirements management: Requirements are managed not only during RE but also throughout the system development life cycle. Key requirements management concerns are maintaining traceability between requirements and other requirements or system components and handling changes in requirements during system evolution. Table 14 shows that only nine primary studies focus on this key activity.

A. *Techniques*: Requirements traceability is carried out in (Denno and Blackburn 2014; Westman and Nyberg 2014; Wang 2007; Sinha et al. 2018) by different methods such as traceability matrix, contract graphs, process Petri Nets and traceability graphs. A traceability matrix is used in (Denno and Blackburn 2014) to generate the test vectors for requirements-to-test traceability automatically. The safety requirements of a fuel level display system are traced through contract structure, a graph-based approach, in (Westman and Nyberg 2014). A generalized form of Petri Net, called process Petri Net, is used to provide traceability solution for manufacturing activities of bee products (Huang et al. 2016). Non-conflicting checks are employed in (Goorden et al. 2019) to detect the impact of requirements splitting on modules.

The process of change management (categorised as a technique in this study) is adopted by Kulvatunyou et al. (2014) and Lima and Faria (2018). The change in high-level functional requirements of a reference functional ontology is reconfigured automatically in (Kulvatunyou et al. 2014). In (Lima and Faria 2018), a triage-based automatic personnel allocation system in hospital waiting rooms is implemented.

B. Frameworks: In Sinha et al. (2018), a framework called Traceability of Requirements Using Splices (TORUS) is presented for the development of large-scale safety-critical CPS, which is based on a traceability graph (a graph-based structure) to create and manage the trace links between requirements and components of smart grids. Similarly, JavaScript Object Notation (JSON) is used to establish a formal requirement model in (Jue et al. 2019) to modify and trace the requirements. JSON is a lightweight, text-based, language-independent data interchange format (Bray 2017) that is integrated with semi-formal natural language input format. Similarly, in (Michael et al. 2020), a Model/Analyzer Framework is developed for traceability, consistency checking and impact analysis of safety requirements during the development of ICPS.

Classification of ICPS software requirements

Software requirements are derived from system requirements and can be categorised as *functional* and *quality* (non-functional) requirements, according to The Guide to Software Engineering Body of Knowledge (SWEBOK) (Bourque and Fairley 2014). Functional requirements describe features (the “what”) while quality requirements deal with the qualitative aspects of how functional requirements are achieved (the “how well”). We report the results of categorising requirement types in the surveyed works by using the requirements classification described in SWEBOK, as shown in Fig. 5.

Fig. 5 shows that 82 (88%) of the studies emphasise quality requirements while around 49 (52%) papers focus on functional requirements. Some studies, like (Vogel-Heuser et al. 2014), (Kang et al. 2018) and (Singh et al. 2019), address both functional and quality requirements in a single study.

Table 13 Simulation and co-simulation tools employed during requirements verification and validation

Tool	Name	Primary studies
Simulation	Simulink	(Zhang et al. 2013), (Nägele et al. 2019), (Singh et al. 2019), (Kim et al. 2019), (Clarke and Zuliani 2011), (Menghi et al. 2019)
	Simulink Design Verifier	(Lin et al. 2018), (Kang et al. 2018), (Kang et al. 2018)
	SCANS	(Nuzzo et al. 2019)
	Modelica	(Bouskela and Jardin 2018)
	3D Siemens's Solid Edge ST9	(Metsälä et al. 2017)
Co-Simulation	Modelica	(Clarke and Zuliani 2011)
	CIROS	(Metsälä et al. 2017)

Among those targeting quality requirements, safety was considered in 42 (45%) papers followed by timing requirements (31, 33%). This shows that safety and timing are both critical and also interdependent in ICPS. Comparatively, least number (3, 3%) of the studies discuss robustness and throughput requirements.

Findings on the applicability Of primary studies

This section focuses on the evidence of the credibility of the surveyed works and answers the following research questions:

- RQ2.1 Which application domains are used to determine the applicability of selected primary studies?
- RQ2.2 Which research methods and research types have been employed in the selected primary studies?

Application domains

Application domains of the 93 primary studies were analysed separately. As shown in Fig. 6, wind-turbine, oil and gas, health care, water treatment, smart grid (Saleh et al. 2015), smart city (Gracia et al. 2018), smart manufacturing (Monostori et al. 2016), avionics & aerospace and transportation are the nine main application domains. 27 (29%) studies (Xu and Zhang 2013; Michael et al. 2020; Huang et al. 2019; Grobelna 2020; Wisniewski et al. 2020; Denno and Blackburn 2014; Gräßler et al. 2020; Dang et al. 2016; Goorden et al. 2019; Drozdov et al. 2019; Askarpour et al. 2019; Neghina et al. 2019; Chen et al. 2018; Balasubramaniyan et al. 2016; Gawanmeh et al. 2017; Drozdov et al. 2017; Rashid and Hasan 2020; Metsälä et al. 2017; Iglesias et al. 2017; Ribeiro et al. 2016; Akkaya et al. 2016; Li et al. 2016; Wang et al. 2011; Huang et al. 2016; Zhan et al. 2019; von Birgelen and Niggemann 2018; Vogel-Heuser et al. 2014) target the domain of smart manufacturing which was the most prominent ICPS domain. It was followed by transportation systems covered in (25, 26%) studies (Jue et al. 2019; Mühlfelder 2018; Loucopoulos et al. 2019; Kang et al. 2018; Dang et al. 2016; Vogel-Heuser et al. 2014; Zhang et al. 2013; Ye-Jing et al. 2013; Huang et al. 2019; Mashkoor and Hasan 2012; Kumar et al. 2012; Clarke and Zuliani 2011; Bu et al. 2011; Zhang 2011; Akella and McMillin 2009; Westman and Nyberg 2014; Zhang 2014, 2013b, c, a, d; Xu and Zhang 2013; Goorden et al. 2019; Knüppel et al. 2020; Bernardi et al. 2020). The lowest consideration was given to the oil and gas sector (Yue et al. 2015) and to industrial wind-turbines with just 1 (1.2%) study (Seceleanu et al. 2017) targeting each domain.

Table 14 Techniques and framework in requirements management activity

Type	Methods	Type	Primary studies
Techniques	Requirements Traceability	Traceability Matrix	(Denno and Blackburn 2014)
		Contracts graph	(Westman and Nyberg 2014)
		Petri Net Process	(Huang et al. 2016)
		Traceability Graph	(Sinha et al. 2018)
	Non-Conflicting Checks		(Goorden et al. 2019)
	Change Management		(Kulvatunyou et al. 2014), (Lima and Faria 2018)
Frameworks		TORUS	(Sinha et al. 2018)
		Formal Requirement Model	(Jue et al. 2019)
		Model Analyzer Framework	(Michael et al. 2020)

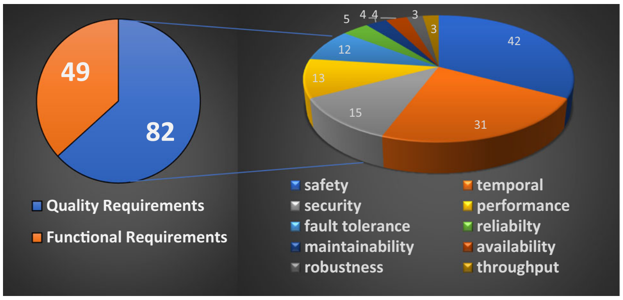

Fig. 5 Functional and quality requirements targeted in primary studies

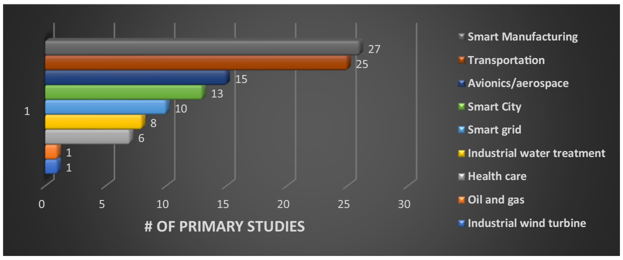

Fig. 6 Categorisation of primary studies based on application domains

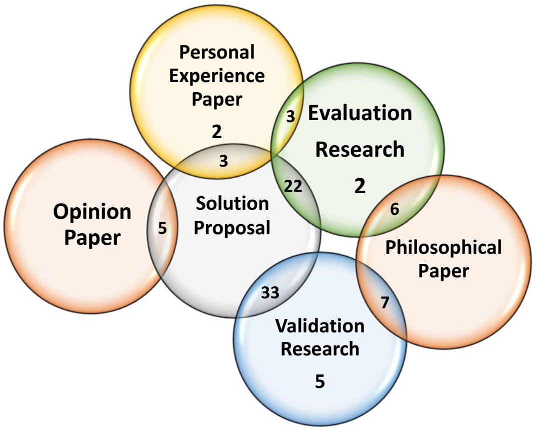

Fig. 7 Categorisation of primary studies based on research types

Research methods and research types

Fig. 7 classifies the primary studies by research types (Wieringa et al. 2006).

We divide the studies into six categories: solution proposal, evaluation research, validation research, philosophical papers, opinion papers and personal experience paper.

According to the figure, some papers span more than one category. For example, primary studies (Sanwal and Hasan 2013) and (Kulvatunyou et al. 2014) appear in the solution proposal, validation research and opinion paper categories. The most prominent distinction worth mentioning is between validation and evaluation research (Wieringa et al. 2006). Papers classified as evaluation research include case studies, experiments with practitioners, action research, and variations of these methods. In contrast, validation research proposes a novel solution such as mathematical analysis, proof of concept, simulation and prototyping. Solution proposal papers contain novel ideas or techniques that lack full validation or provide evidence on the noteworthy enhancement of existing techniques. Papers with new frameworks are classified as philosophical papers and opinion papers typically report information from leading experts. In personal experience papers, authors report on their own experiences. 63 out of the 93 primary studies were characterized as solution proposals. 33 of these present proof of validity and therefore fall in both solution and validation research categories. 13 papers lie in the group of philosophical papers. Out of these 7 studies contain validation and the other 6 feature empirical validation. Evaluation research converges with the solution proposal and personal experience classes in 22 and 3 primary studies, respectively. The (5) opinion papers also present solutions and hence overlap with the respective category.

Besides the research type, Fig. 8 shows that the use of different *research methods* and experimental setups (Wieringa

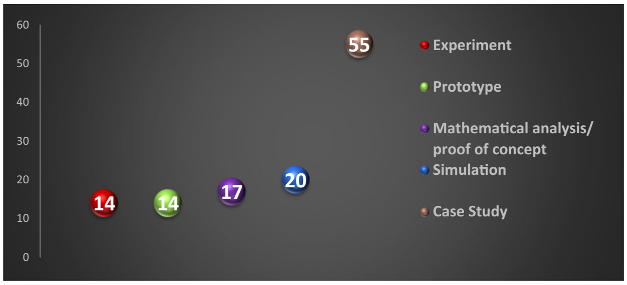

Fig. 8 Research methods in primary studies

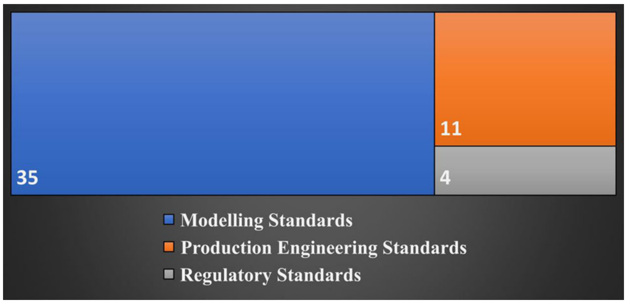

Fig. 9 Adopted industrial standards in primary studies

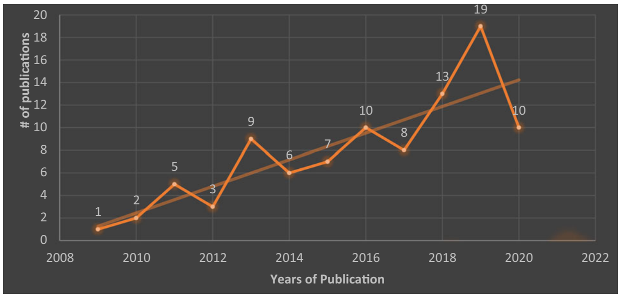

Fig. 10 Number of primary studies published per year

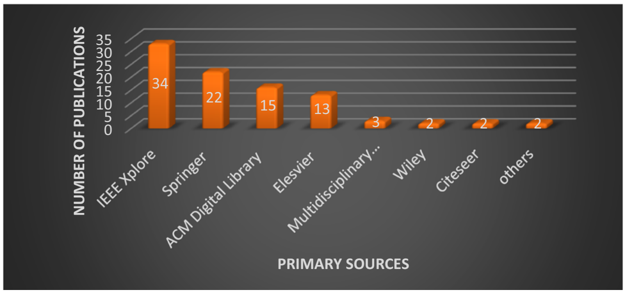

Fig. 11 Number of primary studies published by publication sources

et al. 2006) in the primary studies helped us in deciding the methods of the research.

Overall, case study is the dominant research method, whereas, experiments and prototyping are the least considered methodologies. Fifty-five (59%) primary studies used case study as a research method. On the other hand, simulation and mathematical analysis or proof of concept are carried out in 20 (21%) and 17 (18%) different studies, respectively.

Bibliography mapping

This section reports findings corresponding to the following research questions.

- RQ3.1. Which industrial standards have been adopted in identified primary studies?
- RQ3.2 How can the identified primary studies be classified according to the publication years?
- RQ3.3 What are the publication sources and venue types for the identified primary studies?

Classification of primary studies w.r.t industrial standards

The industrial standards adopted in primary studies, shown in Fig. 9, have been divided into modelling standards (Sanford et al. 2020), production system engineering standards (Lu et al. 2016) and regulatory standards. 35 (38%) of the primary studies claim to use different modelling standards followed by 11 (12%) that use production system engineering standards. Technologies integrated for manufacturing industries have to comply with regulatory standards. However, only 4 (4%) of the studies cover regulatory standards (Westman and Nyberg 2014; Sinha et al. 2018; Gomez et al. 2020; Bernardi et al. 2020).

Classification of primary studies by publication years

The line graph in Fig. 10 shows the number of publications each year. Overall, there has been a moderate increase in numbers over the last 10 years. 2013 saw the first spike of activity with the publication of 9 studies. In 2019, the maximum number of relevant studies (19, 20%) were published.

Classification of primary studies by publication sources and venue types

Fig. 11 shows the breakdown of the selected primary studies according to published sources. 34 (34%) of the publications were published in IEEE Xplore, followed by Springer, who published 22 (28%) of the relevant articles. Fig. 12 shows that most works 53 (57%) were published in conferences, followed by journals. A low number (1%) of relevant technical reports or whitepapers is another indication of low industry adoption.

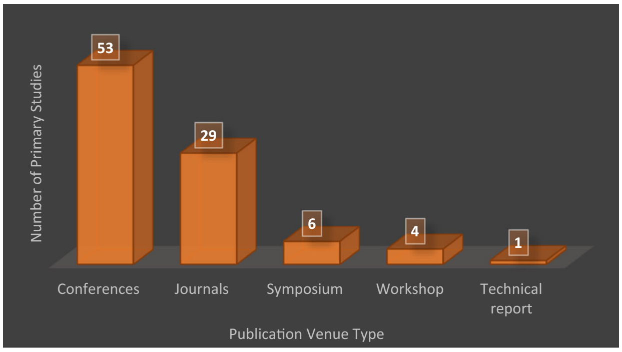

Fig. 12 Number of primary studies published by publication venue type

Discussion

A quality assessment Of this study

We evaluate the quality of our systematic mapping study by the quality criteria measures QC1–5 presented in (Khan et al. 2019). Each measure is given a score of 0, 0.5 or 1 for a SMS. The total quality score for a study is the sum of its individual scores, which is quantified as low ($0.5 \leq$ quality score ≤ 2), medium ($2.5 \leq$ quality score ≤ 3), or high ($3.5 \leq$ quality score ≤ 5).

The scoring for our SMS based on these criteria is computed as follows.

1. QC1. *Inclusion and Exclusion criteria* have been clearly defined, so we score this measure at 1.
2. QC2. *Search adequacy* is demonstrated by using four reputed digital libraries, resulting in a score of 1.
3. QC3. An explicit *synthesis method* based on a well-used methodology has been presented, resulting in a score of 1 for this measure.
4. QC4. *Quality assessment of included primary studies* was conducted but not reported, so we score this measure at 0.5.
5. QC5. *Information about the primary studies* is provided for each primary study, resulting in scoring this measure at 1.

The overall score of 4.5 means that this SMS lies in the high-quality category. Generally, as the purpose of mapping is to give a broad overview of research, so this quality assessment does not rely on a qualitative assessment of the selected primary studies (Kitchenham et al. 2010).

Gap analysis

Our survey of 93 primary studies selected out of 3,645 research papers using a multi-phase methodology (Section 3) leads to some interesting observations that reveal both current trends and areas requiring further research.

Observation 1 *Most studies explore formal methods with an increasing number looking at the integration of formal and semi-formal methods.*

Meta-analysis: Only a few (eleven) focus solely on semi-formal methods, while 31 studies integrate formal and semi-formal methods. SFM are easier to use and can reduce the time required for producing requirement specifications. FM are more structured and can be automated, but require user expertise and may result in high costs for specifying requirements. The integration allows for a balanced approach where the level of formality can be chosen depending on the criticality of the requirements being handled (Hall 2005).

Observation 2 *Most studies focus on model-based techniques using formal analysis or verification methods, simulation/co-simulation, agents, model checking and model testing, domain-specific models, semi-formal methods and iterative approaches.*

Meta-analysis: The surveyed works use model-based techniques for different RE activities. This shows that while model-oriented paradigms for ICPS are growing, but there is no accepted standard for modelling and evaluating ICPS (Derigent et al. 2020). Owing to the closed-coupling of hardware, software, and physical structure of ICPS, model-based techniques have to consider three perspectives uniformly: functionality (implemented in software), physicality (physical environment and hardware platform), and architecture. Nevertheless, most of the current model-based techniques do not cover all of these three aspects uniformly.

Observation 3 *Most works focus primarily on safety and timing requirements.*

Meta-analysis: The dynamic nature of ICPS raises challenges regarding the quality requirements of systems which needs consideration at the requirements stage. Fig. 5 shows that more attention should be given to robustness, throughput, maintainability availability and fault-tolerance. Additionally, predictability and self-awareness requirements are not covered at all.

Observation 4 *Studies that use SysML, Modelica and IEC 61499 propose methodologies useful to improve production system engineering. Few works like (Iglesias et al. 2017) aim to reduce manufacturing costs by improving information exchange among suppliers and manufacturers by using ISA 95 Standard. Some works, for example, Metsälä et al. (2017); LeMay et al. (2011) have tried to develop control and production planning strategies in order to improve robustness, reconfigurability, flexibility and security of ICPS using OPC-UA, system-level iterative approaches, formal analysis techniques or combined formal and semi-formal methods.*

Meta-analysis: Existing design and development standards are far from being sufficient for the ICPS ecosystem because they cannot keep pace with rapidly evolving requirements. New or improved standards are required which have an impact on product and production life cycles, business cycles, and supply chain management in order to improve quality, economy and productivity.

Observation 5 *Most works address multiple, but not all, RE activities.*

Meta-analysis: There is still no standard or generally accepted RE process defined for ICPS.

Observation 6 *Most works focus on only problem or solution domain requirements.*

Meta-analysis: More work is needed to study the interaction between problem and solution domain requirements.

Observation 7 *Formal ontology or domain-specific languages, frameworks and tools help to capture cross-domain relationship, give domain-specific views of requirements at different levels of abstraction and promote the reusability of requirements specification in a particular domain.*

Meta-analysis: Muti-domain integration and migration is a significant challenge in ICPS. More work is needed to provide clear semantics (interfaces) to establish and maintain the relationships between different domains.

Observation 8 *Formal contracts in primary studies are either vertical contracts (design exploration, early detection of errors), horizontal contracts where rules are formalised for subsystems interaction with its environment or stochastic contracts.*

Meta-analysis: Formal contracts for different domains need continuous-time contracts (Fisher et al. 2014) that can not only differentiate between the discrete event and continuous changes but also, based on discrete constraints, can express the bounds on continuous behaviours. Formal contracts for probabilistic requirements are still in the early stages. We also require tools to formalise requirements through contracts effectively and to explore techniques to improve their suitability and scalability.

Observation 9 *Requirements elicitation is a critical RE activity in ICPS but only a few works look at eliciting requirements, removing ambiguities, and elaboration.*

Meta-analysis: Systematic, formal or semi-formal approaches for requirements elicitation and management are urgently needed in large-scale distributed ICPS. These approaches would need to strike the right balance between expressiveness and rigour to be usable in industry (Jeon et al. 2020).

Observation 10 *Privacy and trustworthiness are important quality requirements that are becoming increasingly important in ICPS (Fink et al. 2017). Unfortunately, none of the surveyed works focuses on these requirements.*

Meta-analysis: New formal or semi-formal methodologies that extend legacy methods for these concerns are needed urgently.

Observation 11 *Three papers classified in requirements analysis and one in requirements V&V use co-simulation to analyse and verify large-scale behaviours in the early stages of system development.*

Meta-analysis: FM and SFM in RE complement *simulation* as well as *co-simulation*. Simulation is a well-understood strategy to explore and test systems and finds several uses in the RE of ICPS. Akkaya et al. (2016) and Menghi et al. (2019) use simulation to analyse requirements. Kim et al. (2019) use Simulink Design Verifier (SDV) with the nuSMV model checkers for requirements V&V using simulation. Kang et al. (2018); Dang et al. (2016) use simulation for both requirements analysis and V&V purpose. Co-simulation and SysML are integrated in (Neghina et al. 2019) which shows that co-simulation can be used with semi-formal foundations, too. Testing for validation purpose involves the use of simulation for requirements testing, like in (Kim et al. 2019) and (Menghi et al. 2019). Co-simulation is a promising *modular* approach to manage complexity in ICPS (Wiesner et al. 2015) and is a priority area for future development.

Observation 12 *Only two studies integrate requirements validation and verification.*

Meta-analysis: The interplay of validation and verification is largely unexplored, with most techniques focussing on only one aspect. Future exploration may reveal optimisations and efficiencies in taking a holistic and integrated approach towards these closely related aspects of RE in ICPS.

Observation 13 *Most works employ model checking techniques for verification.*

Meta-analysis: Model checking is attractive as it is fully automated. However, user-guided verification like in theorem proving may be more desirable for large-scale ICPS where model checking does not scale. Model checking and proof-theoretic approaches can be combined to verify complex requirements of ICPS.

Observation 14 *The empirical evidence of the effectiveness of emerging semi-formal and formal methods is largely missing. The limited number of personal experience papers using formal or semi-formal methods in the industry is also deficient.*

Meta-analysis: Industry adoption is low. Processes for the rapid maturation of lab-based solutions and testing them in industrial settings are needed.

Observation 15 *Interesting combinations of different types of methods have been successfully used for various RE activities. Timed CSP is combined with a cause-effect graph in (Kim et al. 2019), Zhang (2013b) mix automata with Modelica, and Metsälä et al. (2017) use abstract state machines illustrated using 3DSiemens' Solid Edge ST9.*

Meta-analysis: These groupings indicate that the availability of specialised methods enables novel integration that may be better suited than individual methods. For instance, Modelica is a popular method to model the physical part of an ICPS and can be combined with AADL or UML to model the cyber aspects, like in (Zhang 2013c).

A conceptual model to aid practitioners

The primary drivers of the RE process in ICPS are regulatory standards, software requirements (functional and quality requirements), and stakeholders requirements. Furthermore, formalisms in selected studies belong to different illustration styles and programming paradigms. Therefore, RE of ICPS using these formalisms also adopts, indirectly, these illustration styles and programming paradigms. Thus, the relationship between RE of ICPS and formalisms is captured in the form of a conceptual model shown in Fig. 13. Adapted from (Lana et al. 2019), this model relates to illustration styles defined in (Mandayam and Steven 1995) and programming paradigms listed in (Van-Roy and Haridi 2004). This conceptual model is mapped in Table 15 to *RE activities, formalisms* (formal/ semi-formal), *types of methods* (methods can be semi-formal or formal languages or techniques. Frameworks and tools are not included here because they inherit the characteristics of the languages and techniques that they support, such as *illustration style* and *programming paradigm*.

The findings of this study indicate that the illustration style hierarchy adopted by the primary studies can be divided into two sub-styles: Property-Oriented (PO) or Model-Oriented (MO). PO sub-style depicts the properties of the system at a higher level of abstraction resulting in the less-detailed specification. It can be further categorized into algebraic-oriented and axiom-oriented styles. Axiom-oriented styles include ontology-based, contracts-based, rule-based and knowledge-based programming paradigms. MO is comprised of object-oriented, aspect-oriented, state-based, probability-based, automata-based, language-oriented and discrete-event based programming paradigms.

The findings of this study can be transformed into guidelines for academic and industrial practitioners. For academic practitioners, Section 4 determines the current trends and gaps within the scope of this study, which are also addressed in Section 5.2. For industrial industrial practitioners, Table 8, Table 11, Table 12, Table 13, Fig. 5 and Fig. 6 can assist in identifying state-of-the-art methods that can be adopted. To benefit both academic and industrial practitioners, we present a further comparative analysis between formal and semi-formal languages and techniques in Table 16 and Table 17, respectively. The agility and reflection of system engineering practices in our conceptual model and its mapping can help both academic and industrial practitioners select formal and semi-formal methods depending on programming paradigms, illustration styles or requirement types.

We illustrate the utility of our conceptual model on a real-world case study of a USB stick production line (Neghina et al. 2019). This case study includes subsystems like Human–Machine interface (HMI), Part Tracker, Warehouse, Robotic Arm, Wagons, Test Station to handle the customers' order. The detailed description of each of these subsystems and experimental report is provided in (Neghina et al. 2019). Our model provides an overview for the practitioners to select the desired methods depending on their needs. For example, if the objective is to specify the timings constraints of ICPS along with an asynchronous analysis of all involved subsystems, we can use VDM-RT as it can produce abstract functional model units for all subsystems. These model units can describe both the physical and cyber parts of the USB production system. Furthermore, to link the model units to customer order requirements, we can examine an object-oriented paradigm, such as SysML that can be used later to configure co-simulation. Similarly, requirements can also be defined as rules and analysed by Process Algebra, while Hybrid Automata can be used to describe both the cyber and physical aspects of the production system. According to these decisions, VDM-RT as a formal technique, SysML as a semi-formal language and co-simulation as a method to perform requirement analysis, are integrated. VDM-RT and SysML belong to the discrete event and object-oriented programming paradigms and adopt a model-oriented illustration style. Similarly, Process algebra is included in a rule-based programming paradigm that has a property-oriented illustration style while Hybrid Automata follow a state-based programming paradigm using a model-oriented style.

Conclusions and future work

As the use of ICPS has grown, researchers and practitioners have become more inclined to employ new or improved requirements engineering methods for developing quality ICPS. However, the scale, heterogeneity, and complexity of ICPS, as well as their evolutionary nature and the involvement of a multitude of stakeholders have made RE of ICPS a challenging task. A comprehensive landscape of methods

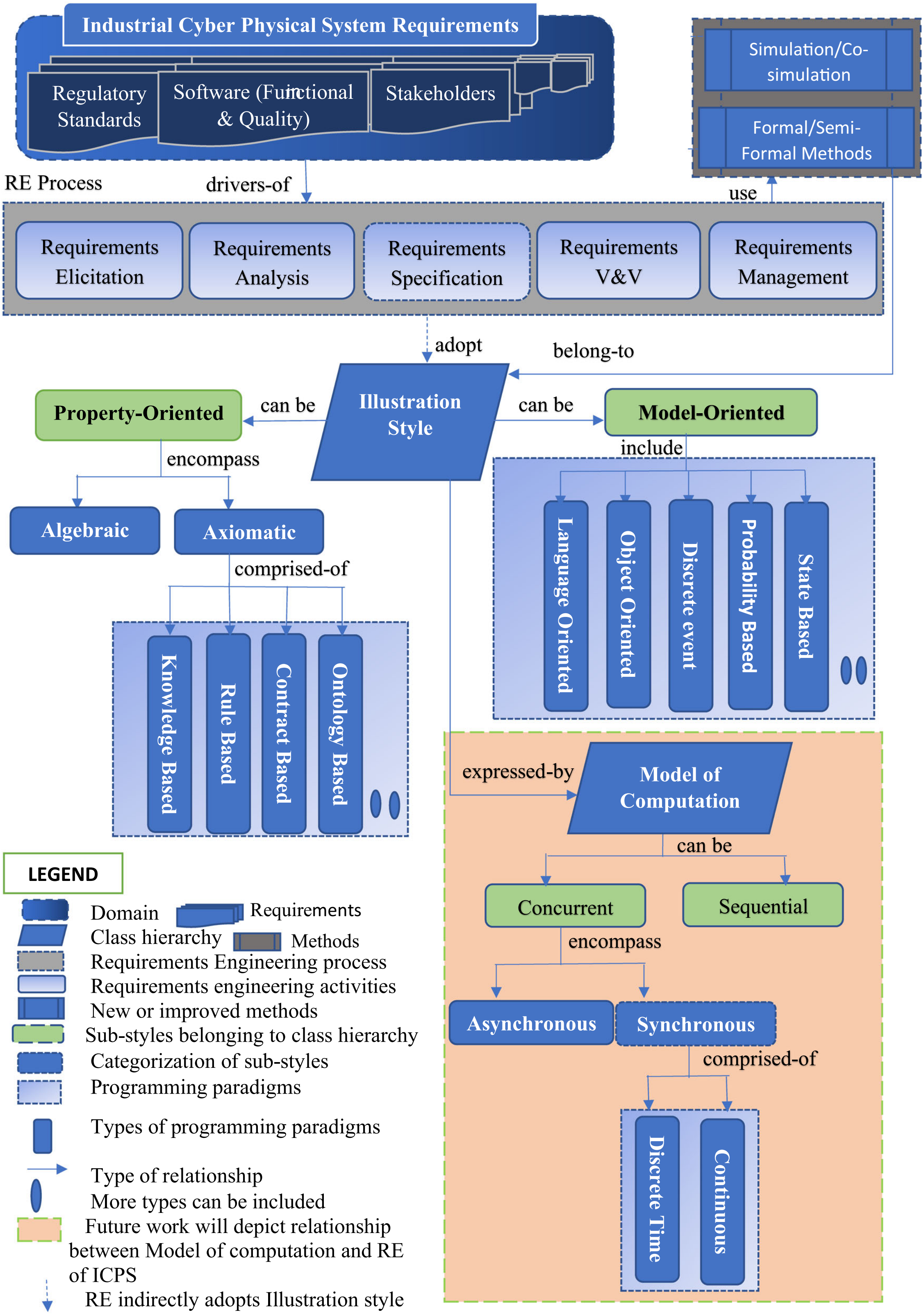

Fig. 13 Contextualization based on illustration style, programming paradigms and improved methodologies to show the relationship between formalisms and requirements engineering of Industrial cyber-physical system

Table 15 Mapping of conceptual model on the basis of requirements engineering activities, formalisms (semi-formal/formal), types of methods (techniques/languages), method name, illustration style (property-oriented/model-oriented), programming paradigms

RE Activities	Formalism	Types of Methods	Method Name	Illustration Style	Programming Paradigm
Requirements Elicitation	Semi-Formal	Techniques	Prototype	MO	Language-Based
			Graphic Notations	MO	Discrete Event
Requirements Analysis	Formal	Languages	PA	PO	Rule-Based
			Alloy	PO	Rule-Based
	Semi-Formal		AADL	MO	Object-Oriented
			UML		Object-Oriented
			SysML		Object-Oriented
			MARTE		Object-Oriented/ Language-Based
			Modelica		Object-Oriented
			POOSL		Object-Oriented
	Formal	Techniques	Abstract State Machine	MO	Automata-Based
			Hybrid Timed Automata		Automata Based/ State Based
			Cellular Automata		Discrete-Event
			Graph theory		Automata-Based
			Formal Contract	PO	Contract-Based
			Formal Ontology		Ontology-Based
			VDM-RT	MO	Discrete-Event
			PDRTA		Automata-Based/ Probability-Based
			Stochastic Timed Automata		Probability-Based
	Semi-Formal		Use-Cases	MO	Object-Oriented
			Domain Model		Object-Oriented
			Meta-Model		Object-Oriented
			Object Diagram		Object-Oriented
			Parametric Diagram		Object-Oriented

Table 15 continued

RE Activities	Formalism	Types of Methods	Method Name	Illustration Style	Programming Paradigm
Requirements Specification	Formal	Languages	Event B	MO	State-Based
			Z Language		State-Based
			Object Z	MO	Object-Oriented/ Aspect-Oriented
			Temporal Logic	PO	Rule-Based
			CASL	PO	Algebraic
			SMV		State-Based
			First Order Logic		Rule-Based
			ASLan++	MO	Language-Based
			Timed CSP	PO	Rule-Based
			Value Specification Language	MO	Language-Based
			RTCM	MO	Language-Based
			Form-L	PO	Rule-Based/ Language-Oriented
Requirements Verification and Validation	Formal	Languages	Real-Time Maude	PO	Rule-Based
			HCSP	MO	Discrete-Event
			STL	MO	Discrete-Event
			ETL	PO	Rule-Based / Language-Based
		Techniques	Formal Contract	PO	Contract-Based
			State invariants	PO/MO	Rule-Based/ Discrete-Event
			Cause-effect Graphing	MO	Automata-Based
			Knowledge-Based(Quantative Fitness Function)	PO	Knowledge-Based
			CIPNs	MO	State-Based
Requirements Management	Semi-Formal	Technique	Traceability Graph	MO	Automata -Based
			Non-conflicting check	PO	Rule-Based/ Algebraic
			Traceability Matrix	MO	Object-Oriented
	Formal		Process Petri-Nets	MO	Discrete-Event

Table 16 Advantages and disadvantages of formal and semi-formal languages

Languages	Advantages	Disadvantages
AADL	model hardware and software components of ICPS, check components consistency in discrete-time	does not model the spatial-temporal features of transportation, does not verify the components in continuous time
UML	help practitioners to tackle complex software structures, model functional structures of software abstracted from inner details of system	not able to customize description rules for extended requirements, modeling and management of physical resources and concurrency are challenging
SysML	general-purpose modelling language used for specification, analysis, design, V&V of ICPS	unable to represent mechanical, physical components and description of system resources through static and dynamic diagrams
MARTE	model hardware and software parts of ICPS, provide interoperability between different tools of used for specification, design and verification	does not have specific methodology, physical models are not modelled by it
Modelica	model and simulate physical parts of ICPS, increase robustness, involve in hybrid and multi-domain modelling	graphical formalisms are not available, hard to find programming and modelling errors
POOSL	expressive in nature, model for analysis of ICPS, supports flexible and reusable designs.	unable to combine inheritance and concurrency in a flexible way
PA	suitable to describe the order of occurrence of events, facilitate the modular composition of process.	unable to handle complex operations, does not support performance and function analysis
Alloy	have strong analytical capability, very expressive in defining complex structure and behavior of ICPS	limited support for numerical constraint, lack built in support for dynamic system modelling
Event-B	used for system level modelling and analysis of ICPS, provides different level of abstraction of system, support formal refinements, can prove timing requirements	stochastic behavior are not supported well
Object Z	express complex data operations, support modularity	operations are atomic, no direct way to determine that how much time an operation will take to complete
LTL	expressive, formal and compact notation to state safety and liveness requirements	cannot represent the non-deterministic state or transitions
CTL	can represent non-deterministic states or transitions	cannot express fairness requirements directly in formalism, does not cater stochastic features
SMV	provide modular and hierarchical description of ICPS, supports reusability	graphical formalism is not available

Table 16 continued

Languages	Advantages	Disadvantages
Signal First Order Logic	powerful language, can capture time and magnitude continuous behavior of ICPS, able to handle uncertainties due to ICPS-environment interaction	non-deterministic exponential time is hard to monitor
ASLan++	flexible, expressive, easy to use and formally specify security requirements of ICPS	does not cover all modes of encryption for security
Timed CSP	provide the facility to analyse run-time behaviors, has strong ability to model process control	time-consuming, modelling is tedious
RTCM	have user friendly template, set of keywords, and rules for writing test cases specifications.	specific to the application so applicability is limited
FORM-L	cater stochastic aspects and deterministic behaviour , deal with spatiotemporal constraints	need the expertise to understand the language.
Real-Time Maude	easy to use and specify safety and timing requirements of ICPS.	no guarantee of complete search and model checking
HCSP	expressive, easy to use, model hybrid behaviour of ICPS	involve the sequential composition of operation which make interruption difficult to handle
ETL	capable of describing continuous real-time physical aspects and stochastic behaviours	dependent on high-level languages in order to model
STL	formalise control-theoretic properties, express timing constraint	online monitoring is not efficient, optimisation problem, cannot intend for frequency-domain analysis

Table 17 Advantages and disadvantages of formal and semi-formal techniques

Techniques	Advantages	Disadvantages
Prototype	allows iterative development of ICPS for better understanding and communication among stakeholders	complexity and scale of ICPS make prototyping harder, impractical to prototype the physical system whose existence is unknown in advance
Abstract State Machine	enables high level analysis and design, effective when different analysis and validation techniques may be applied to the same model	not easy to construct accurate ground models of requirements
Hybrid Automata	describes systems with mixed continuous and discrete dynamics	provides limited support to represent non-linear and spatial-temporal features of ICPS
Cellular Automata	very efficient in model spatial-temporal requirements of ICPS, allows efficient parallel computation	emergent behaviors of ICPS can lead to redundant results, not suitable for the environment that generates unpredictable results
Graph Theory	models the topology of ICPS, demonstrate spatial-temporal requirements	cannot describe heterogeneity of nodes
Formal Contract	enhance reusability of specification, efficient for large and hybrid design-space exploration	has scalability problem in case of probability requirements
Formal Ontology	use to overcome cross-domain barrier, efficient for semantic interoperability	data is structured in such a way that it does not allow to add inconsistent data which can be made consistent later
Cause-effect Graphing	detect the ambiguity and incompleteness by generating testcases.	expertise is required, need to be very focused, difficult to work with large specifications
CIPNs	efficient for communication with environment by signals, concurrency can be shown graphically	need expertise to understand methodology
VDM-RT	model timings constraints of ICPS, support asynchronous analysis	VDM models are not accurate in the sense of physical implementation

Table 17 continued

Techniques	Advantages	Disadvantages
Stochastic Timed Automata	generic technique to model spatial behaviours in various domains and provide accurate semantics of requirements	result in random delays, unlikely behaviours can be ignored due to selection of interactions.
PDRTA	model the discrete events of ICPS based on probability	does not perform well in practice
State invariant	validates the states of ICPS components, helps in making a decision and monitoring cyber process	As ICPS have the mass of state invariants so one fault in the system may result in an abundance of broken invariants.
Traceability Graph	useful for multi-level tracing, visualisation tool for a large set of requirements	Modelling effort and ensuring coherence between requirements and graph models
Traceability Matrix	helps the testing team to understand the level of testing activities done for the specific product.	not suitable for higher dimensions trace links
Process Petri-Nets	manage the complexity of ICPS by providing a clear separation of technological model and resource sufficiency	are system dependent, are not equipped with a notion of physical distribution, do not portray self-organizing cyber and physical production
Use Cases	user-oriented, manage the complexity, provide a base to specify end-to-end temporal requirements.	For ICPS having infinite interactions with its environment, a large number of use cases have to be created. If creations are limited to only important scenarios, then few use cases lead to insufficient specification.
Domain Model	gives a better understanding of the complex domain, helps in improving communication among teams	time-consuming, requires domain expertise
Requirement Diagram	helps in analysis and traceability of formal and quality requirements.	lack precise decomposition semantics, relatively immature diagram, relationships allocation are incomplete and ambiguous
Meta-Model	helps in understanding and describing the large system, supports reusability, assure consistency among teams	can be difficult and challenging to define right abstractions and structure them for reusability, face compatibility problems among multi-domains, time-consuming
Parametric Diagrams	models the constraints and mathematical relationship between components in order to fulfil the performance requirements.	Parametric constraints are not clearly understandable, immature as compared to other SysML diagrams.

that can identify, analyse, verify or manage ICPS requirements has been missing. This research addresses this by reporting a systematic mapping study on available formal and semi-formal methods for the RE of ICPS. Semi-formal and formal methods promise rigour and structure that are seen as essential ingredients in building robust, repeatable and scalable requirements engineering processes for building ICPS. The findings of the study result in a novel conceptual model that highlights current trends and research gaps in the area.

Our findings have identified several new research directions as future work: Firstly, comprehensive comparisons of each formal/semi-formal technique, language, tool or method utilised in different activities of requirements engineering can be analysed. Next, co-simulation techniques can be optimised to analyse the different types of requirements such as performance, probability, trust and privacy and fault tolerance for providing customer-specific solutions and for resolving constraints on ICPS. Thirdly, practitioners can combine model checking and proof-theoretic approaches to verify the complex requirements of ICPS. Also, more work is needed to provide clear semantics (interfaces) to establish and maintain the relationships between different domains. Lastly, formal contracts for probabilistic requirements are still in the early stages and new methods are required to formalise such requirements.

For our future work, we will extend our conceptual model to further analyse the relationship between RE of ICPS and models of computation (MoCs). This analysis will not only articulate their benefits to the industrial community but also help them identify and choose the appropriate MoCs in order to comply with industrial standards. Furthermore, we are developing design patterns to enable easy integration of security requirements from standards into ICPS component and system software. The key challenge in this direction is to sufficiently secure a system without sacrificing performance or overwhelming the limited computation powers of ICPS hardware components like PLCs.

References

Aceto, L., Ingólfsdóttir, A., Larsen, K G., & Srba, J. (2007). Reactive systems: Modelling, specification and verification (1st ed.). Cambridge University Press.

Adepu, S., Kang, E., Jackson, D., & Mathur, A. (2016, 05). Model-based security analysis of a water treatment system. In *2nd international workshop on software engineering for smart cyber-physical systems (sescps)*. Austin, Texas. https://doi.org/10.1145/2897035.2897041

Adepu, S., & Mathur, A. (2016a, 01). Introducing cyber security at the design stage of public infrastructures: A procedure and case study. In *Complex systems design & management asia* (Vol. 426, pp. 75–94). ChamSpringer. https://doi.org/10.1007/978-3-319-29643-2_6

Adepu, S., & Mathur, A. (2016b, 01). An investigation into the response of a water treatment system to cyber attacks. In *Ieee 17th international symposium on high assurance systems engineering (hase)*. Orlando, United States. https://doi.org/10.1109/HASE.2016.14

Adepu, S., & Mathur, A. (2016c, 05). Using process invariants to detect cyber attacks on a water treatment system. In *Ifip international conference on information security and privacy protection* (pp. 91–104). Gent, Belgium. https://doi.org/10.1007/978-3-319-33630-5_7

Ahmad, E., Dong, Y., Larson, B., Lü, J., Tang, T., & Zhan, N. (2015). Behavior modeling and verification of movement authority scenario of Chinese train control system using AADL. *Science China Information Sciences*, *58*(11), 1–20.

Ahmed, R., & Robinson, S. (2007). *Simulation in business and industry: how simulation context can affect simulation practice? In Proceedings of the 2007 spring simulation multiconference-volume 3* (pp. 152–159). USA: Virginia.

Akella, R., & McMillin, B M. (2009). Model-checking BNDC properties in cyber-physical systems. In *2009 33rd annual ieee international computer software and applications conference* (Vol. 1, pp. 660–663). Seattle, Washington, USA.

Akkaya, I., Derler, P., Emoto, S., & Lee, E. A. (2016). Systems engineering for industrial cyber-physical systems using aspects. *Proceedings of the IEEE*, *104*(5), 997–1012.

Askarpour, M., Ghezzi, C., Mandrioli, D., Rossi, M., & Tsigkanos, C. (2019). Formal methods in designing critical cyber-physical systems. In *From software engineering to formal methods and tools, and back* (Vol. 11865, pp. 110–130). Porto, PortugalSpringer. https://doi.org/10.1007/978-3-030-30985-5_8

BA, K., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering (Tech. Rep.). The Pennsylvania State UniversityKeele University and Durham University Joint Report.

Bae, K., Krisiloff, J., Meseguer, J., & Ölveczky, P. (2015). 06). Designing and verifying distributed cyber-physical systems using multirate pals: An airplane turning control system case study. *Science of Computer Programming*, *103*, 13–50. https://doi.org/10.1016/j.scico.2014.09.011.

Balasubramaniyan, S., Srinivasan, S., Buonopane, F., Subathra, B., Vain, J., & Ramaswamy, S. (2016). Design and verification of cyber-physical systems using truetime, evolutionary optimization and uppaal. *Microprocessors and Microsystems*, *42*, 37–48.

Bartocci, E., Manjunath, N., Mariani, L., Mateis, C., Ničković, D., & Pastore, F. (2020). CPSDebug: a tool for explanation of failures in cyber-physical systems. In *Proceedings of the 29th acm sigsoft international symposium on software testing and analysis* (p. 569–572). New York, NY, USA. https://doi.org/10.1145/3395363.3404369

Bernardi, S., Gentile, U., Marrone, S., Merseguer, J., & Nardone, R. (2020). Security modelling and formal verification of survivability properties: Application to cyber–physical systems. *Journal of Systems and Software*, 110–746. Retrieved on 22 October 2020

Bourque, P., & Fairley, R E. (2014). Guide to the software engineering body of knowledge (swebok (r)): version 3.0. IEEE Computer Society Press.

Bouskela, D., & Jardin, A. (2018). Etl: a new temporal language for the verification of cyber-physical systems. In *Annual ieee international systems conference (syscon)* (pp. 1–8). Vancouver, BC, Canada. https://doi.org/10.1109/SYSCON.2018.8369502

Bouskela, D., Nguyen, T., & Jardin, A. (2017). Toward a rigorous approach for verifying cyber-physical systems against requirements. *Canadian Journal of Electrical and Computer Engineering*, *40*(2), 66–73.

Bray, T. (2017, December). The JavaScript Object Notation (JSON) data interchange format (No. 8259). RFC 8259. RFC Editor. Retrieved from https://rfc-editor.org/rfc/rfc8259.txt https://doi.org/10.17487/RFC8259

Bu, L., Wang, Q., Chen, X., Wang, L., Zhang, T., Zhao, J., et al. (2011). Toward online hybrid systems model checking of cyber-physical systems' time-bounded short-run behavior. *ACM SIGBED Review*, *8*(2), 7–10.

Cengic, G., & Akesson, K. (2010). On formal analysis of IEC 61499 applications, part b: Execution semantics. *IEEE Transactions on Industrial Informatics*, *6*(2), 145–154. https://doi.org/10.1109/TII.2010.2040393.

Chen, Y., Dai, W., Zhang, Z., Pang, C., & Vyatkin, V. (2018). A case study on knowledge driven code generation for software-defined industrial cyber-physical systems. In *Iecon 2018-44th annual conference of the ieee industrial electronics society* (pp. 4687–4692). Washington, DC, USA.

Clarke, E M., & Zuliani, P. (2011). Statistical model checking for cyber-physical systems. In *International symposium on automated technology for verification and analysis* (pp. 1–12). Taipei, Taiwan.

Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., et al. (2007). *All about maude - a high-performance logical framework: How to specify, program and verify systems in rewriting logic*. Berlin: HeidelbergSpringer-Verlag.

Colombo, A. W., Bangemann, T., Karnouskos, S., Delsing, J., Stluka, P., Harrison, R., et al. (2014). Industrial cloud-based cyber-physical systems. *The IMC-AESOP Approach*, *22*, 4–5. https://doi.org/10.1007/978-3-319-05624-1.

Colombo, A. W., Karnouskos, S., Kaynak, O., Shi, Y., & Yin, S. (2017). Industrial cyberphysical systems: A backbone of the fourth industrial revolution. *IEEE Industrial Electronics Magazine*, *11*(1), 6–16. https://doi.org/10.1109/MIE.2017.2648857.

Dang, T., Mady, A E.D., Boubekeur, M., Kumar, R., & Moulin, M. (2016). Validation of industrial cyber-physical systems: an application to hvac systems. In *International conference on complex systems design & management* (pp. 57–69). Paris, France.

Davis, J A., Clark, M., Cofer, D., Fifarek, A., Hinchman, J., Hoffman, J., ... Wagner, L. (2013). Study on the barriers to the industrial adoption of formal methods. In *International workshop on formal methods for industrial critical systems* (pp. 63–77). Madrid, Spain.

Denno, P O., & Blackburn, M. (2014). Virtual design and verification of cyber physical systems: industrial process plant design. In *Conference on systems engineering research (cser 2014).* CA, USA.

Derigent, W., Cardin, O., & Trentesaux, D. (2020). Industry 4.0: contributions of holonic manufacturing control architectures and future challenges. *Journal of Intelligent Manufacturing*, 1–22. https://doi.org/10.1007/s10845-020-01532-x

Drozdov, D., Patil, S., Dubinin, V., & Vyatkin, V. (2019). Towards formal ASM semantics of timed control systems for industrial CPS. In *24th ieee international conference on emerging technologies and factory automation (etfa)* (pp. 1682–1685). Zaragoza, Spain.

Drozdov, D., Patil, S., & Vyatkin, V. (2017). Formal modelling of distributed automation cps with cp-agnostic software. *Service Orientation in Holonic and Multi-Agent Manufacturing*, 35.

Du, D., Huang, P., Jiang, K., & Mallet, F. (2018). pCSSL: A stochastic extension to MARTE/CCSL for modeling uncertainty in cyber physical systems. *Science of Computer Programming*, *166*, 71–88.

Dyba, T., Dingsoyr, T., & Hanssen, G K. (2007). Applying systematic reviews to diverse study types: an experience report. In *First international symposium on empirical software engineering and measurement (esem 2007)* (pp. 225–234). NW Washington, DC,United States.

Elmqvist, H., Boudaud, F., Broenink, J., Brück, D., Ernst, T., Fritzson, P., ... Mattsson, S. (1999). ModelicaTM-a unified object-oriented language for physical systems modeling. *Tutorial and Rationale, versión*, **1**, .

Ezio, B., N, M., L, M., Cristinel, M., & D, N. (2019). Automatic failure explanation in CPS models. In *17th international conference on software engineering and formal methods* (Vol. 11724, pp. 69–86). Oslo, Norway. https://doi.org/10.1007/978-3-030-30446-1_4

Feiler, P. H., & Gluch, D. P. (2012). *Model-based engineering with aadl: an introduction to the sae architecture analysis & design language*. Addison-Wesley.

Felizardo, K R., Nakagawa, E Y., Feitosa, D., Minghim, R., & Maldonado, J C. (2010). An approach based on visual text mining to support categorization and classification in the systematic mapping. In *14th international conference on evaluation and assessment in software engineering (ease)* (pp. 1–10). Swindon,United Kingdom.

Ferrante, O., Di Guglielmo, L., Senni, V., & Ferrari, A. (2017). Application of model-based safety assessment to the validation of avionic electrical power systems. In *International symposium on model-based safety and assessment* (pp. 243–254). Trento, Italy.

Fink, G A., Edgar, T W., Rice, T R., MacDonald, D G., & Crawford, C E. (2017). Security and privacy in cyber-physical systems. In *Cyber-physical systems* (pp. 1–23). BostonAcademic Press. https://doi.org/10.1016/B978-0-12-803801-7.00009-2

Fisher, A., Jacobson, C A., Lee, E A., Murray, R M., Sangiovanni-Vincentelli, A., & Scholte, E. (2014). Industrial cyber-physical systems – iCyPhy. In *Proceedings of the fourth international conference on complex systems design & management* (pp. 21–37). France, Paris. https://doi.org/10.1007/978-3-319-02812-5_2

France, R., Evans, A., Lano, K., & Rumpe, B. (1998). The UML as a formal modeling notation. *Computer Standards & Interfaces*, *19*(7), 325–334.

Franceschini, F., Maisano, D., & Mastrogiacomo, L. (2016). Empirical analysis and classification of database errors in Scopus and Web of Science. *Journal of Informetrics*, *10*(4), 933–953. https://doi.org/10.1016/j.joi.2016.07.003.

Fuchs, A., Gürgens, S., Weber, D., Bodenstedt, C., & Ruland, C. (2010). Formalization of smart metering requirements. In *Proceedings of the international workshop on security and dependability for resource constrained embedded systems* (pp. 1–6). Vienna, Austria.

Gabmeyer, S., Kaufmann, P., Seidl, M., Gogolla, M., & Kappel, G. (2019). A feature-based classification of formal verification techniques for software models. *Software & Systems Modeling*, *18*(1), 473–498.

Garcia, L., Mitsch, S., & Platzer, A. (2019). HyPLC: Hybrid Programmable Logic Controller Program Translation for Verification. In *Proceedings of the 10th acm/ieee international conference on cyber-physical systems* (pp. 47–56). Montreal Quebec, Canada. https://doi.org/10.1145/3302509.3311036

Gawanmeh, A., Alwadi, A., & Parvin, S. (2017). Formal verification of control strategies for a cyber physical system. In *Ieee 37th international conference on distributed computing systems workshops (icdcsw)* (pp. 91–96). Atlanta, GA, USA. https://doi.org/10.1109/ICDCSW.2017.59

Geraldes, A., Geretti, L., Bresolin, D., Muradore, R., Fiorini, P., Mattos, L., & Villa, T. (2018, 09). Formal verification of medical CPS: A laser incision case study. *ACM Transactions on Cyber-Physical Systems*, **2**(4), 1–29. https://doi.org/10.1145/3140237

Gomez, F., Aguilera, M., Olsen, S., & Vanfretti, L. (2020, 04). Software requirements for interoperable and standard-based power system modeling tools. *Simulation Modelling Practice and Theory*, **103**, 102095. https://doi.org/10.1016/j.simpat.2020.102095

Goorden, M., van de Mortel-Fronczak, J., Reniers, M., Fokkink, W., & Rooda, J. (2019). The impact of requirement splitting on the efficiency of supervisory control synthesis. In *International workshop on formal methods for industrial critical systems* (pp. 76–92). Amsterdam, The Netherlands.

Gracia, T J H., & García, A C. (2018). Sustainable smart cities. creating spaces for technological, social and business development. *Boletín*

Científico de las Ciencias Económico Administrativas del ICEA, **6**(12), https://doi.org/10.1007/978-3-319-40895-8

Grobelna, I. (2020). Formal verification of control modules in cyber-physical systems. *Sensors*, *20*(18), 51–54. https://doi.org/10.3390/s20185154.

Gräßler, I., Bodden, E., Pottebaum, J., Geismann, J., & Roesmann, D. (2020, 01). Security-oriented fault-tolerance in systems engineering: a conceptual threat modelling approach for cyber-physical production systems. In *Advanced, contemporary control* (Vol. 1196, pp. 1458–1469). ChamSpringer. https://doi.org/10.1007/978-3-030-50936-1_121

Guttag, J. V., Horning, J. J., Garland, S., Jones, K., Modet, A., & Wing, J. (1993). *Larch: languages and tools for formal specification*. Springer.

Hachicha, M., Halima, R. B., & Kacem, A. H. (2019). Formal verification approaches of self-adaptive systems: A survey. *Procedia Computer Science*, *159*, 1853–1862. https://doi.org/10.1016/j.procs.2019.09.357.

Hall, A. (2005). Realising the benefits of formal methods. In *7th international conference on formal engineering methods, icfem 2005*. Berlin, Heidelberg. https://doi.org/10.1007/11576280_1

Hissam, S.A., Chaki, S., & Moreno, G A. (2015). High assurance for distributed cyber physical systems. In Proceedings of the. (2015). *European conference on software architecture workshops* (pp. 1–4). Dubrovnik Cavtat: Croatia.

Hofmann, M., & Klinkenberg, R. (2013). Rapidminer: Data mining use cases and business analytics applications. *Chapman and Hall/CRC*,. https://doi.org/10.1201/b16023.

Hopcroft, J E., Motwani, R., & Ullman, J D. (2000). Introduction to automata theory, languages, and computation, 2nd edition. *SIGACT News*, **32**, 60–65.

Huang, J., Zhu, Y., Cheng, B., Lin, C., & Chen, J.(2016). A petrinet-based approach for supporting traceability in cyber-physical manufacturing systems.*Sensors*, **16**(3), 382.

Huang, L., Liang, T., & Kang, E. Y.(2019). Tool-supported analysis of dynamic and stochastic behaviors in cyber-physical systems. In *Ieee 19th international conference on software quality, reliability and security (qrs)* (pp. 228–239).Sofia, Bulgaria.

Huth, M., & Ryan, M.(2004). Logic in computer science: Modelling and reasoning about systems. USACambridge University Press.

Iglesias, A., Lu, H., Arellano, C., Yue, T., Ali, S., & Sagardui, G.(2017, 09). Product line engineering of monitoring functionality in industrial cyber-physical systems: A domain analysis. In *Proceedings of the 21st international systems and software product line conference* (pp. 195–204).Sevilla, Spain. https://doi.org/10.1145/3106195.3106223

Jalali, S., & Wohlin, C. (2012). Systematic literature studies: database searches vs. backward snowballing.In Proceedings of the. (2012). *Acm-ieee international symposium on empirical software engineering and measurement* (pp. 29–38). Sweden: Lund.

Jeon, B., Yoon, J. S., Um, J., & Suh, S. H.(2020). The architecture development of industry 4.0 compliant smart machine tool system (smts).*Journal of Intelligent Manufacturing*, **31**(8), 1837–1859. https://doi.org/10.1007/s10845-020-01539-4

Jue, W., Yineng, S., Wu, X., & Dai, W.(2019, 10). A semi-formal requirement modeling pattern for designing industrial cyber-physical systems. In *45th annual conference of the ieee industrial electronics society* (pp. 2883–2888).Lisbon, Portugal. https://doi.org/10.1109/IECON.2019.8926665

Kallel, S.(2011).Specifying and monitoring non-functional properties (Unpublished doctoral dissertation) Technische Universität.

Kang, E. Y., Huang, L., & Mu, D.(2018). Formal verification of energy and timed requirements for a cooperative automotive system.In *Proceedings of the 33rd annual acm symposium on applied computing* (p. 1492–1499).New York, NY, USA. https://doi.org/10.1145/3167132.3167291

Kang, E. Y., Mu, D., Huang, L., & Lan, Q.(2018). Model-based verification and validation of an autonomous vehicle system (Vol. abs/1803.06103). arXiv preprint, arXiv:1803.06103

Kaur, A., Gulati, S., Samridhi & Singh.(2012). A comparative study of two formal specification languages: Z-notation and B-method. In *Proceedings of the second international conference on computational science, engineering and information technology* (pp. 524–531). Coimbatore UNK, India.

Keshav, S.(2007, July). How to read a paper.*SIGCOMM Computer Communication Review*, **37**(3), 83–84. https://doi.org/10.1145/1273445.1273458

Khan, M. U., Sherin, S., Iqbal, M. Z., & Zahid, R. (2019). Landscaping systematic mapping studies in software engineering: A tertiary study. *Journal of Systems and Software*, *149*, 396–436. https://doi.org/10.1016/j.jss.2018.12.018.

Kim, J., Chon, S., Park, J., & (2019). Suggestion of testing method for industrial level cyber-physical system in complex environment.In,. (2019). *IEEE international conference on software testing, verification and validation workshops (icstw)* (pp. 148–152). Xian: China.

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O P., Turner, M., Niazi, M., & Linkman, S.(2010). Systematic literature reviews in software engineering – a tertiary study.*Information and Software Technology*, **52**(8), 792–805. https://doi.org/10.1016/j.infsof.2010.03.006

Knüppel, A., Jatzkowski, I., Nolte, M., Thüm, T., Runge, T., & Schaefer, I.(2020). Skill-based verification of cyber-physical systems.In *International conference on fundamental approaches to software engineering. lecture notes in computer science* (Vol. 12076, pp. 203–223). https://doi.org/10.1007/978-3-030-45234-6_10

Krueger, M., Walden, D., & Hamelin, R.(2011). Systems engineering handbook: A guide for system life cycle processes and activities (v. 3.2. 1).International Council on Systems Engineering (INCOSE), San Diego, CA.

Kulvatunyou, B., Wallace, E., Ivezic, N., & Lee, Y.(2014, 09). Toward manufacturing system composability analysis: A use case scenario.In *Advances in production management systems* (Vol. 439, pp. 658–666).Ajaccio, FranceSpringer. https://doi.org/10.1007/978-3-662-44736-9_80

Kumar, P., Goswami, D., Chakraborty, S., Annaswamy, A., Lampka, K., & Thiele, L.(2012). A hybrid approach to cyber-physical systems verification.In *Dac '12: The 49th annual design automation conference 2012* (pp. 688–696). San Francisco, California.

Lana, C A., Guessi, M., Antonino, P O., Rombach, D., & Nakagawa, E Y.(2019). A systematic identification of formal and semi-formal languages and techniques for software-intensive systems-of-systems requirements modeling.*IEEE Systems Journal*, **13**(3), 2201–2212. https://doi.org/10.1109/JSYST.2018.2874061

LeMay, E., Ford, M D., Keefe, K., Sanders, W H., & Muehrcke, C. (2011). Model-based security metrics using adversary view security evaluation (advise). In *8th international conference on quantitative evaluation of systems(qest)* (pp. 191–200).Aachen, Germany.

Li, F., Zhang, P., Huang, H., & Chen, G.(2016, 03). A model-based service-oriented integration strategy for industrial CPS.In *International conference on industrial iot technologies and applications* (Vol. 173, pp. 222–230).GuangZhou, China. https://doi.org/10.1007/978-3-319-44350-8_22

Li, N., Tsigkanos, C., Jin, Z., Hu, Z., & Ghezzi, C. (2020). Early validation of cyber-physical space systems via multi-concerns integration. *Journal of Systems and Software*, *170*, 110–742.

Lima, B., & Faria, J P.(2018). Towards real-time patient prioritization in hospital emergency services.In *Ieee 20th international conference on e-health networking, applications and services (healthcom)* (pp. 1–4).Ostrava, Czech Republic.

Lin, Q., Adepu, S., Verwer, S., & Mathur, A.(2018). Tabor: a graphical model-based approach for anomaly detection in industrial control systems.In *Proceedings of the 2018 on asia conference on computer and communications security* (pp. 525–536). Incheon Republic of Korea. https://doi.org/10.1145/3196494.3196546

Loucopoulos, P., & Karakostas, V. (1995). System requirements engineering. McGraw-Hill, Inc.

Loucopoulos, P., Kavakli, E., & Chechina, N.(2019). Requirements engineering for cyber physical production systems. In *31st international conference on advanced information systems engineering* (pp. 276–291).Rome, Italy.

Lu, Y., Morris, K C., & Frechette, S.(2016). Current standards landscape for smart manufacturing systems.*National Institute of Standards and Technology, NISTIR*, **8107**, 39.

Mancini, T., Mari, F., Melatti, I., Salvo, I., Gruber, J., Hayes, B., ... Elmegaard, L.(2018, 10). Parallel statistical model checking for safety verification in smart grids.In *Ieee international conference on smart grid communications (smartgridcomm)* (pp. 1–6). Aalborg, Denmark. https://doi.org/10.1109/SmartGridComm.2018.8587416

Mandayam K., S., & Steven P., M.(1995). Formal verification of an avionics microprocessor (Tech. Rep.).CSL-95-04Technical report, SRI International Computer Science Laboratory.

Mann, C.(2009). A practical guide to sysml: the systems modeling language.*Kybernetes*, **38**, . https://doi.org/10.1108/k.2009.06738aae.004

Mashkoor, A., & Hasan, O.(2012). Formal probabilistic analysis of cyber-physical transportation systems. In *International conference on computational science and its applications* (Vol. 7335, pp. 419–434).Salvador de Bahia, Brazil,. https://doi.org/10.1007/978-3-642-31137-6_32

Menghi, C., Nejati, S., Gaaloul, K., & Briand, L C.(2019). Generating automated and online test oracles for simulink models with continuous and uncertain behaviors.In *Proceedings of the 2019 27th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering* (p. 27–38). Tallinn, Estonia. https://doi.org/10.1145/3338906.3338920

Meseguer, J., & Ölveczky, P. (2012). Formalization and correctness of the pals architectural pattern for distributed real-time systems. *Theoretical Computer Science*, *451*, 1–37. https://doi.org/10.1016/j.tcs.2012.05.040.

Metsälä, S., Gulzar, K., Vyatkin, V., Gröhn, L., Väänänen, E., Saikko, L., & Nyholm, M.(2017). Simulation-enhanced development of industrial cyber-physical systems using OPC-UA and IEC 61499. In *International conference on industrial applications of holonic and multi-agent systems* (pp. 125–139).Lyon, France.

Michael, T., Atif, M., Andreas, D., & Alexander, E.(2020). Ensuring safe and consistent coengineering of cyber physical production systems: A case study.*Journal of Software Evolution and Press*, **32**(2), . https://doi.org/10.1002/smr.2308

Misson, H A., Gonçalves, F S., & Becker, L B.(2019). Applying integrated formal methods on CPS design. In *Ix brazilian symposium on computing systems engineering (sbesc)* (pp. 1–8).Natal, Brazil. https://doi.org/10.1109/SBESC49506.2019.9046084

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., ... Ueda, K.(2016). Cyber-physical systems in manufacturing.*Cirp Annals*, **65**(2), 621–641. https://doi.org/10.1016/j.cirp.2016.06.005

Mühlfelder, M.(2018). Analysis and design of a cyber-physical production system (CPPS) in sensor manufacturing. A case study.In *Proceedings of the 20th congress of the international ergonomics association (iea 2018)* (Vol. 822, pp. 391–400).Florence, ItalySpringer. https://doi.org/10.1007/978-3-319-96077-7_41

Nägele, T., Broenink, T., Hooman, J., Broenink, J., & (2019). Early analysis of cyber-physical systems using co-simulation and multi-level modelling.In,. (2019). *IEEE international conference on industrial cyber physical systems (icps)* (pp. 133–138). Taipei: Taiwan.

Neghina, M., Zamfirescu, C. B., & Pierce, K.(2019). Early-stage analysis of cyber-physical production systems through collaborative modelling.*Software and Systems Modeling*, 1–20.

Nejati, S., Gaaloul, K., Menghi, C., Briand, L C., Foster, S., & Wolfe, D. (2019). Evaluating model testing and model checking for finding requirements violations in simulink models. In *Proceedings of the 2019 27th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering* (p. 1015–1025). Tallinn, Estonia. https://doi.org/10.1145/3338906.3340444

Nuzzo, P., Li, J., Sangiovanni-Vincentelli, A L., Xi, Y., & Li, D. (2019). Stochastic assume-guarantee contracts for cyber-physical system design.*ACM Transactions on Embedded Computing Systems (TECS)*, **18**(1), 1–26.

Nuzzo, P., Lora, M., Feldman, Y. A., Sangiovanni-Vincentelli, A. L., & (2018). CHASE: contract-based requirement engineering for cyber-physical system design.In,. (2018). *Design, automation & test in europe conference & exhibition (date)* (pp. 839–844). Germany: Dresden.

Object Management Group. (2011). UML profile for modeling and analysis of real-time and embedded systems (MARTE).Object Management Group.

Ölveczky, P C., & Meseguer, J.(2007). Semantics and pragmatics of real-time maude.*Higher-order and symbolic computation*, **20**(1-2), 161–196.

Öztemel, E., & Gursev, S. (2020). Literature review of industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*, 127–182. https://doi.org/10.1007/S10845-018-1433-8.

Pagliari, L., Mirandola, R., & Trubiani, C.(2019, 07). Engineering cyber-physical systems through performance-based modelling and analysis: A case study experience report.*Journal of Software: Evolution and Process*, **32**(1), . https://doi.org/10.1002/smr.2179

Penzenstadler, B., & Eckhardt, J.(2012). A requirements engineering content model for cyber-physical systems. In *2012 second ieee international workshop on requirements engineering for systems, services, and systems-of-systems (ress)* (pp. 20–29).Chicago, USA.

Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, *64*, 1–18. https://doi.org/10.1016/j.infsof.2015.03.007.

Rashid, A., & Hasan, O.(2020). Formal analysis of the continuous dynamics of cyber–physical systems using theorem proving.*Journal of Systems Architecture*, **112**, .Retrieved on 22 October 2020 https://doi.org/10.1016/j.sysarc.2020.101850

Rashid, A., Siddique, U., & Tahar, S.(2019). Formal verification of cyber-physical systems using theorem proving.In *International workshop on formal techniques for safety-critical systems* (pp. 3–18). Shenzhen,China. https://doi.org/10.1007/978-3-030-46902-3_1

Ribeiro, F G C., Rettberg, A., Pereira, C E., & Soares, M S.(2016). An analysis of the value specification language applied to the requirements engineering process of cyber-physical systems.*IFAC-PapersOnLine*, **49**(30), 42–47. https://doi.org/10.1016/j.ifacol.2016.11.123

Rocchetto, M., & Tippenhauer, N O. (2017). Towards formal security analysis of industrial control systems.In Proceedings of the. (2017). *Acm on asia conference on computer and communications security* (pp. 114–126). Abu Dhabi: United Arab Emirates.

Roscoe, B.(1998). The theory and practice of concurrency. Prentice Hall.

Ruchkin, I. (2015) . Towards integration of modeling methods for cyber-physical systems.In The doctoral symposium at the 18th acm, ieee international conference of model-driven engineering languages and systems, . (2015). *(models 2015*. Ottawa: Canada.

Sakarovitch, J.(2009). Elements of automata theory. USACambridge University Press.

Saleh, M S., Althaibani, A., Esa, Y., Mhandi, Y., & Mohamed, A A. (2015). Impact of clustering microgrids on their stability and resilience during blackouts. In *International conference on smart grid and clean energy technologies (icsgce)* (pp. 195–200).Offenburg, Germany.

Sanford, F., Dov, D., & Yaniv, M.(2020).Modeling Standards. https://www.sebokwiki.org/wiki/Modeling_Standards. [Online; accessed 27-October-2020]

Sanwal, M U., & Hasan, O.(2013). Formal verification of cyber-physical systems: coping with continuous elements. In *International conference on computational science and its applications* (pp. 358–371).Ho Chi Minh, Vietnam.

Seceleanu, C. C., Johansson, M. E., Suryadevara, J., Sapienza, G., Seceleanu, T., Ellevseth, S. E., et al. (2017). Analyzing a wind turbine system: From simulation to formal verification. *Science of Computer Programming*, *133*, 216–242. https://doi.org/10.1016/j.scico.2016.09.007.

Sepúlveda, S., Cravero, A., & Cachero, C. (2016). Requirements modeling languages for software product lines: A systematic literature review. *Information and Software Technology*, *69*, 16–36.

Sharma, A., & Singh, M. (2013). Comparison of the formal specification languages based upon various parameters.*IOSR Journal of Computer Engineering (IOSR-JCE)*, **11**(5) 37–39.

Simon, F., Felex, W., Jivka, O., & (2019). A guideline for the requirements engineering process of SMEs regarding to the development of CPS.In,. (2019). *8th international conference on industrial technology and management (icitm)* (pp. 85–94). Cambridge: United Kingdom.

Singh, N. K., Wang, H., & (2019). Virtual environment model of glucose homeostasis for diabetes patients.In,. (2019). *IEEE international conference on industrial cyber physical systems (icps)* (pp. 417–422). Taipei: Taiwan.

Sinha, R., Dowdeswell, B., Zhabelova, G., & Vyatkin, V.(2018). Torus: Scalable requirements traceability for large-scale cyber-physical systems.*ACM Transactions on Cyber-Physical Systems*, **3**(2), . https://doi.org/10.1145/3203208

Sinha, R., Pang, C., Martínez, G S., Kuronen, J., & Vyatkin, V.(2015). Requirements-aided automatic test case generation for industrial cyber-physical systems. In *20th international conference on engineering of complex computer systems (iceccs)* (pp. 198–201).Gold Coast, Australia.

Strauss, A., & Corbin, J.(1998). Basics of qualitative research techniques.Sage publications Thousand Oaks, CA.

Sun, H., Liu, J., Chen, X., & Du, D.(2015). Specifying cyber physical system safety properties with metric temporal spatial logic. In *Asia-pacific software engineering conference (apsec)* (pp. 254–260).New Delhi, India.

Takbiri, Y., & Amini, A.(2019, 11). A survey on large-scale requirements engineering. In *4th international conference on combinatorics, cryptography, computer science and computing*.Tehran, Iran.

Theelen, B., Florescu, O., Geilen, M., Huang, J., Putten, P., & Voeten, J.(2007, 05). Software/hardware engineering with the parallel object-oriented specification language. In *5th ieee/acm international conference on formal methods and models for codesign (memocode 2007)* (pp. 139–148).Washington, DC,United States. https://doi.org/10.1109/MEMCOD.2007.371231

Van-Roy, P., & Haridi, S.(2004). Concepts, techniques, and models of computer programming. MIT press.

Vegendla, A., Duc, A N., Gao, S., & Sindre, G.(2018). A systematic mapping study on requirements engineering in software ecosystems.*Journal of Information Technology Research (JITR)*, **11**(1), 49–69.

Vogel-Heuser, B., Schütz, D., Frank, T., & Legat, C.(2014). Model-driven engineering of manufacturing automation software projects–A SysML-based approach.*Mechatronics*, **24**(7), 883–897.

von Birgelen, A., & Niggemann, O.(2018). Anomaly detection and localization for cyber-physical production systems with self-organizing maps.In *Improve-innovative modelling approaches for production systems to raise validatable efficiency* (Vol. 8, pp. 55–71).Springer Vieweg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-57805-6_4

Wang, J.(2007). Petri nets for dynamic event-driven system modeling.*Handbook of Dynamic System Modeling*, **1**, 24.

Wang, J., Song, Y., Wu, X., & Dai, W.(2019). A semi-formal requirement modeling pattern for designing industrial cyber-physical systems. In *45th annual conference of the ieee industrial electronics society* (pp. 2883–2888).Lisbon, Portugal.

Wang, R., Song, X., Zhu, J., & Gu, M.(2011). Formal modeling and synthesis of programmable logic controllers.*Computers in Industry*, **62**(1), 23–31.

Westman, J., & Nyberg, M.(2014). Specifying and structuring requirements on cyber-physical systems using contracts (Tech. Rep.). Machine Design (Dept.), Mechatronics.KTH, School of Industrial Engineering and Management (ITM).

Wieringa, R., Maiden, N., Mead, N., & Rolland, C.(2006). Requirements engineering paper classification and evaluation criteria: A proposal and a discussion.*Requirements Engineering*, **11**(1), 102–107.

Wiesner, S., Hauge, J B., & Thoben, K. D.(2015). Challenges for requirements engineering of cyber-physical systems in distributed environments.In *Ifip international conference on advances in production management systems* (pp. 49–58). Tokyo, Japan.

Wiesner, S., Marilungo, E., & Thoben, K. D.(2017). Cyber-physical product-service systems–challenges for requirements engineering.*International Journal of Automation Technology*, **11**(1), 17–28.

Wisniewski, R., Grobelna, I., & Karatkevich, A.(2020). Determinism in cyber-physical systems specified by interpreted petri nets.*Sensors*, **20**(19), 55–65. https://doi.org/10.3390/s20195565

Wortmann, A., Barais, O., Combemale, B., & Wimmer, M.(2019). Modeling languages in industry 4.0: an extended systematic mapping study.*Software and Systems Modeling*, 1–28. https://doi.org/10.1007/s10270-019-00757-6

Wu, X., Goepp, V., & Siadat, A.(2020). Concept and engineering development of cyber physical production systems: A systematic literature review.*The International Journal of Advanced Manufacturing Technology*, 1–19.

Xu, B., & Zhang, L. (2013). Formal specification of cyber physical systems: three case studies based on clock theory.In *Ieee international conference on green computing and communications and ieee internet of things and ieee cyber, physical and social computing* (pp. 804–811). https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.143

Ye-Jing, L., Ming-Cai, C., Guang-Quan, Z., Yu-zhen, S., Fei, F., & Xing-hua, H.(2013). A model for vehicular cyber-physical system based on extended hybrid automaton.In *8th international conference on computer science & education* (pp. 1305–1308).Colombo, Srilanka.

You, J., Li, J., Xia, S., & (2012) . A survey on formal methods using in software development. In Iet international conference on information science and control engineering, . (2012). (icisce 2012. *Shenzhen, China.* https://doi.org/10.1049/cp.2012.2353.

Yu, W., Dillon, T., Mostafa, F., Rahayu, W., & Liu, Y.(2019). Implementation of industrial cyber physical system: challenges and solutions. In *Ieee international conference on industrial cyber physical systems (icps)* (pp. 173–178).Taipei, Taiwan.

Yue, T., Ali, S., Zhang, M. (2015). RTCM: a natural language based, automated, and practical test case generation framework.In Proceedings of the. (2015). *International symposium on software testing and analysis* (pp. 397–408). USA: Baltimore MD.

Zhan, H., Lin, Q., Wang, S., Talpin, J. P., Xu, X., & Zhan, N.(2019). Unified graphical co-modelling of cyber-physical systems Using AADL and Simulink/Stateflow.In *7th international symposium on unifying theories of programming 2019* (Vol. 11885, pp. 109–129).Porto, PortugalSpringer. https://doi.org/10.1007/978-3-030-31038-7_6

Zhang, L.(2011, 08). Formal specification for real time cyber physical systems using aspect-oriented approach. In *Fifth international conference on theoretical aspects of software engineering* (pp. 213–216). https://doi.org/10.1109/TASE.2011.37

Zhang, L.(2013a, 08). Aspect-oriented modeling for railway control systems.In *Ieee international conference on information and automation, icia 2013* (pp. 236–241).Yinchuan, China. https://doi.org/10.1109/ICInfA.2013.6720302

Zhang, L.(2013b). Modeling railway cyber physical systems based on aadl. In *19th international conference on automation and computing* (pp. 1–6).London, United Kingdom.

Zhang, L., & (2013c). Requirement analysis method for vehicular cyber physical systems.In Ieee 10th international conference on high performance computing and communications &,. (2013). *IEEE international conference on embedded and ubiquitous computing* (pp. 2096–2103). Okayama: Japan.

Zhang, L.(2013d). Requirement specification for transportation cyber physical systems.In *Ieee international conference on green computing and communications and ieee internet of things and ieee cyber, physical and social computing* (pp. 1486–1491).Beijing, China.

Zhang, L.(2013e). Specifying and modeling automotive cyber physical systems. In *Ieee 16th international conference on computational science and engineering* (pp. 603–610).Washington, DC, United States.

Zhang, L., & (2014). Modeling large scale complex cyber physical control systems based on system of systems engineering approach.In,. (2014). *20th international conference on automation and computing* (pp. 55–60). Cranfield: UK.

Zhang, L., He, J., & Yu, W.(2013). Test case generation from formal models of cyber physical system.*International Journal of Hybrid Information Technology*, **6**(3), 15–24.

Zheng, X., Julien, C., & (2015). Verification and validation in cyber physical systems: research challenges and a way forward.In,. (2015). *ieee/acm 1st international workshop on software engineering for smart cyber-physical systems* (pp. 15–18). Florence: Italy.

Zheng, X., Julien, C., Kim, M., & Khurshid, S.(2015). Perceptions on the state of the art in verification and validation in cyber-physical systems.*IEEE Systems Journal*, **11**(4), 2614–2627.