Computing (2026) 108:68
https://doi.org/10.1007/s00607-026-01657-6

REGULAR PAPER

Surveying uncertainty representation: a unified model for cyber-physical systems

Johannes Mäkelburg[1] · **Diego Perez-Palacin**[2,3] · **Raffaela Mirandola**[2] · **Maribel Acosta**[1]

Received: 6 October 2025 / Accepted: 27 March 2026

Abstract

Cyber-Physical Systems (CPS) operate in dynamic environments, leading to different types of uncertainty that affect their design, operation, and reliability. This work provides a comprehensive review of uncertainty representations and categorizes them based on the dimensions used to represent uncertainty. Through this categorization, key gaps and limitations in existing approaches are identified, such as inconsistent terminology, the lack of systematic differentiation between CPS components, and the absence of explicit consideration of autonomy in CPS. To address these issues, a Conceptual Model of Uncertainty Representations in CPS is introduced, which unifies the terminology used in existing frameworks while introducing missing categories specifically tailored to CPS. Our model incorporates distinctions between cyber, physical, and platform components, as well as between autonomous and non-autonomous subsystems, offering a more precise characterization of uncertainty. Its applicability is demonstrated through examples from the automotive domain, showing its effectiveness in capturing and structuring uncertainty in real-world scenarios. This contribution not only harmonizes existing approaches but also establishes a foundation for future research on expressive and domain-aware representations of uncertainty in CPS.

Keywords Uncertainty · Taxonomy · Cyber-physical systems

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant SFB 1608 "CONVIDE" – 501798263.

Extended author information available on the last page of the article

Published online: 21 April 2026

1 Introduction

Cyber-physical systems (CPS) integrate computational components with physical processes through embedded computers that control the system via feedback loops [29, 30]. This integration enables continuous interaction and mutual influence between the physical and computational components [5, 29, 30]. With the rapid advancement of digitalization, CPS have become a fundamental component of modern technological infrastructures. Today, nearly all technological infrastructures can be considered CPS, with prominent examples including automotive systems [11], smart grids [51], and healthcare [20]. A key advantage of CPS lies in their ability to simulate physical processes within a cyber environment, which accelerates technological development, improves system efficiency, and reduces overall costs [5]. CPS achieve these benefits by enabling real-time access to information, supporting predictive maintenance, facilitating predefined decision-making, and optimizing operational processes [36]. However, CPS also introduce significant challenges. They are highly vulnerable to security threats [36, 43] and different types of uncertainties [30]. The literature offers diverse interpretations of uncertainty, from a complete lack of understanding to insufficient data and the gap between the information needed to complete a task and the current information [1, 39, 42, 50]. A general definition, that can be adopted also for CPS is provided by Walker [47], where uncertainty is seen as *any deviation from the unachievable ideal of completely deterministic knowledge of the relevant system*. This issue is exacerbated in CPS, which, unlike traditional software systems, must operate in dynamic and often unpredictable environments where uncertainties arise from multiple sources, including sensor inaccuracies, environmental variations, and unforeseen interactions between system components [32]. Managing these uncertainties is particularly crucial for CPS due to the tight coupling between computational and physical processes. Since CPS rely on the precise coordination of these components, uncertainties can propagate throughout the system, affecting functionality and overall reliability. In safety-critical domains such disruptions can have severe consequences, including system failures and safety risks. Moreover, CPS operate across multiple domains, where their interdisciplinary nature introduces additional sources of uncertainty, particularly between components of different domains. These uncertainties, along with the dynamic and unpredictable nature of CPS, underscore the need for a systematic approach to managing uncertainty throughout the entire system lifecycle. Effectively addressing them requires integrating uncertainty considerations from the earliest stages of software design. Addressing these uncertainties effectively requires incorporating the notion of uncertainty from the beginning of software design. A design approach that inherently accounts for uncertainties allows CPS development to be guided by principles and methodologies that explicitly consider uncertainties.

1.1 Contributions

In this work, we review how uncertainty is represented in CPS, identify the main gaps, and present a unified model to address them, demonstrated on automotive case studies; our contributions are:

1. A structured literature review that categorizes and analyzes existing uncertainty representations in CPS, identifying key gaps and limitations.
2. A harmonized and extended Conceptual Model of Uncertainty Representations that integrates and refines existing models while introducing missing categories.
3. A demonstration of the model's applicability through case studies in the automotive domain, showing its effectiveness in capturing and structuring uncertainty.

This work advances the understanding of uncertainty representation in CPS and provides a foundation for future research on more expressive and structured modeling approaches.

1.2 Paper structure

The remainder of this paper is structured as follows. Section 2 surveys existing work on uncertainty representation, grouping prior studies into model-specific, category-specific, and CPS-related perspectives. Section 3 reviews existing work on uncertainty representation in CPS, categorizing different modeling approaches and identifying key contributions and gaps. Section 4 introduces our harmonized and extended Conceptual Model, which addresses the gaps identified in the literature. Section 5 demonstrates the applicability of our conceptual model using an Autonomous Vehicle as a case study. Section 6 concludes with a summary and future research directions.

2 Related uncertainty surveys

Uncertainty representation has been extensively studied across various domains, with numerous surveys providing valuable insights. However, these works often focus on specific aspects of uncertainty, such as a particular model type, a single uncertainty dimension, or a restricted application domain. In contrast, our work aims to present a more comprehensive perspective, tailored to the challenges of uncertainty in CPS.

We group related surveys into three categories: (1) those on uncertainty in specific model types, (2) those addressing particular representation categories, and (3) those exploring uncertainty in CPS. This section reviews key contributions in each, noting their scope, limitations, and differences from our approach.

Model-Specific Uncertainty Surveys: Among existing surveys, many focus on uncertainty representation within a specific model type: belief/deep-learning combinations [19], neural networks [18], graph neural networks (GNNs) [48], machine learning Processes [16], and large language models (LLMs) [44]. These surveys provide an in-depth analysis within their model-specific domain, but do not consider combinations between models from different domains, like in CPS.

Category-Specific Uncertainty Surveys: Other surveys narrow their scope to a specific aspect of uncertainty representation, for instance, focusing on decision-making or optimization [25, 53], domain-specific applications such as smart grids [41] or power markets [21], measurements [14], or visualization techniques [23], rather than offering a broader view across multiple approaches. These surveys produce domain specific quantification methods or visual encodings, without a structured characterization of uncertainty itself. Our survey takes a broader view, addressing uncertainty representation beyond quantification and visualization.

Method-Specific Uncertainty Surveys: A few CPS-related surveys review techniques for handling uncertainty across disciplines [31] and within CPS [45]. They produce a collection of techniques and tools, rather than a structured representation of uncertainty and where it manifest. Our work complements them by organizing representations and their relationships for CPS.

3 Related work on uncertainty representation

The objective of this literature review is to identify and analyze the most relevant uncertainty representations for the domain of CPS, focusing on key contributions that provide a representative overview of the state of the art.

Our selection prioritized peer-reviewed journal articles and conference proceedings, particularly those that address uncertainty representation beyond a single modeling paradigm or application domain. Works that were limited to highly specialized techniques without broader relevance to CPS were excluded. The representativeness of our selection is based on covering different types of contributions: conceptual frameworks for uncertainty, methodological approaches spanning multiple paradigms, and applied works in CPS case studies. This results in a collection of literature that underpins our analysis of uncertainty representation in CPS. To identify relevant work, we searched major academic databases, including *IEEE Xplore*, *ACM Digital Library*, *SpringerLink*, and *Google Scholar*. Our search queries combined keywords such as *Cyber-Physical Systems*, *Uncertainty Representation*, *Uncertainty Modeling*, and *Uncertainty Quantification*. Additional relevant works were identified through backward and forward citation tracking to ensure the inclusion of foundational and influential contributions.

This selection analyses papers from various relevant domains, focusing on the domains of Software Engineering, Cyber-Physical Systems, and Coupled Models. These domains were selected based on their contributions to understanding and managing uncertainty in CPS. A complete list of all the literature included in the review can be found in Table 1. Figure 1 displays an overview of the different categories of uncertainty across various frameworks. The categories are grouped by colors: blue categories describe the characteristics of uncertainty, the grey categories are the category for handling uncertainty, and the green category is the meta dimension.

The categories presented in the following subsections reflect the most used categories across the selected literature. We focus on categories that appear repeatedly across multiple sources and provide complementary perspectives on uncertainty in CPS, ensuring that only those most relevant are included. while omitting domain-specific ones.

3.1 Location

According to Walker et al. [47], the location of uncertainty refers to the point within a complex model (i.e., a formal representation of a system) where uncertainty mani-

Table 1 Overview of used Literature and its Uncertainty Categories (Miti = Mitigation, Perp = Perspective, Redu = Reducability)

Paper	Location	Source	Type	Time	Level	Redu	Nature	Model	Pattern	Effect	Risk	Miti	Perp
A classification framework of uncer-tainty in architecture-based self-adaptive systems with multiple quality require-ments [32]	✓	✓		✓	✓		✓						
Orthogonal uncertainty modeling in the engineering of cyber-physical systems [6]	✓	✓							✓	✓		✓	
Mastering uncertainty in mechanical engineering [38]	✓									✓			
Defining uncertainty: a conceptual basis for uncertainty man-agement in model-based decision support [47]	✓				✓		✓						
Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation [39]	✓	✓			✓		✓						
Uncertainty Theories for Real-Time Sys-tems [7]		✓											
Uncertainty in self-adaptive software sys-tems [15]		✓			✓								
A taxonomy of uncertainty for dynami-cally adaptive systems [42]		✓		✓						✓		✓	
Managing uncertainty in integrated envi-ronmental modelling: The UncertWeb framework [8]		✓											
Uncertainty in coupled models of cyber-physical systems [1]	✓	✓		✓				✓		✓			
Uncertainty representation in software models: a survey [46]			✓										
Coverage of uncertainties in cyber-physical systems [13]	✓								✓	✓			
Incorporating measurement uncertainty into OCL/UML primi-tive datatypes [9]								✓					
Understanding uncertainty in cyber-physical systems: a con-ceptual model [52]	✓	✓	✓	✓	✓				✓	✓	✓		
Toward modeling and verification of uncertainty in cyber-physical systems [12]	✓	✓	✓	✓					✓	✓	✓		
Uncertainty handling in cyber-physical systems: State-of-the-art approaches, tools, causes, and future directions [3]		✓											✓
Precise Semantics for Uncertainty Mod-eling (PSUM), Ver-sion 1.0 [35]			✓		✓	✓	✓		✓	✓	✓		✓

fests. This helps identify where the uncertainty that affects the outcome is created in the model. Walker et al. outline several typical locations of uncertainty that can apply to most models: *Context*, *Model*, *Inputs*, *Parameters*, and *Outcomes*. *Context* refers to the system's boundaries and how fully it represents the real world. *Model* uncertainty includes both conceptual model uncertainty, which relates to the variables and relationships, and technical uncertainty, which concerns the computer implementation. *Inputs* pertain to the description of the reference system and external forces driving its changes. *Parameters* refer to uncertainty in the data and methods used to calibrate the model parameters. Lastly, *Outcome* uncertainty is the difference between model outcomes and true values that is significant to decision-makers. Chipman et al. [13] build on Walker et al.'s framework, categorizing uncertainty locations as *Inputs*, *Parameters*, and *Models*. *Input* uncertainties typically stem from a lack of knowledge about initial conditions or unforeseen scenarios. *Parameter* uncertainties arise when accurate values are unknown, though statistical knowledge or value ranges may be available. Here, *Model* uncertainty refers to uncertainties in the representation of the existing system.

The taxonomy by Perez-Palacin et al. [39] builds on Walker et al. [47], tailoring it to the uncertainties in the models used by self-adaptive systems. It identifies three types of uncertainty location: *Context*, *Model structural*, and *Input*. The *Context uncertainty* concerns the decided model boundaries with respect to the real world where the system operates, while *Model structural uncertainty* represents how accurately the model characterizes the modeled subset of the real world. *Input*, also called parameter, refers to uncertainty concerning the actual values of input variables or calibrating methods. Pelz et al. [38] introduce *System Design* as a category to identify uncertainty locations, including *Model* uncertainty, *Structural* uncertainty similar to Perez-Palacin et al. [39], and *Data* uncertainty arising from incomplete, unclear, or insufficient data. Acosta et al. [1] combine the understandings of Pelz et al. [38] and Perez-Palacin et al. [39] into a new notion called *Locus*, which specifies which model elements are affected and their corresponding location in the real world. The locus categories are *Parameters*, *Models*, *Analysis*, and *Decision Making*. *Parameter* includes uncertainties related to all types of parameters that characterize a model, while uncertainty related to the model definition, formalism selection, boundaries, and structure are combined in the category *Models*. The category *Analysis* comprises uncertainty from evaluating design decisions through model-based methods, and uncertainties in the category *Decision-Making* relate to the decision-making process when different options are available or when the decision is based on incomplete information and only estimates of the values of interest are available.

Mahdavi-Hezavehi et al. [32] categorize the locations of uncertainty within a system. Potential locations include *Environment*, *Model*, *Adaptation functions*, *Goals*, *Managed systems*, and *Resources*. *Environment* encompasses the execution context and human interactions affecting the system. *Model* refers to the various conceptual models representing the system. *Adaptation functions* correspond to the functionalities performed as part of MAPE-K [26]. *Goals* group the specification, modeling, and modification of system goals. *Managed Systems* organize all application-specific monitoring and adaptation systems. *Resources* include the essential factors and components required for the self-adaptive system to operate normally.

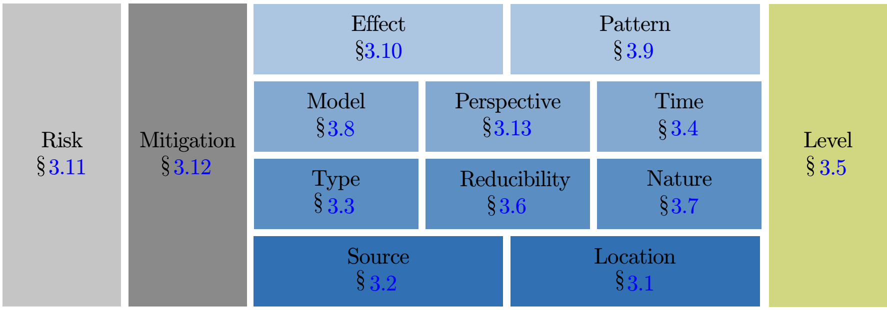

Fig. 1 Overview of different categories representing Uncertainty across various frameworks. Blue indicates characteristics, grey handling, and green the meta dimension

Other works do not further categorize uncertainty locations but provide constructs to represent them. Zhang et al. [52] propose a Belief Model, where uncertainty is the subjective state of a Belief Agent with incomplete knowledge of a statement. In the Belief Model, the location of uncertainty is defined in the Uncertainty Model as the specific place where uncertainty occurs. Bandyszak et al. [6] present an Uncertainty Ontology and use the term *Observation Point* to document the artifacts containing uncertainty, which serve as locations where the system can detect uncertainty. Each observation point may be associated with multiple data sources, representing multiple uncertainty locations.

3.2 Source

In our selected literature, Ramirez et al. [42] first describe Uncertainty Sources, identifying 26 distinct uncertainty sources within dynamically adaptive systems. These sources are organized across three critical phases of the model lifecycle, *Requirement Level*, *Design-Time*, and *Run-Time*, providing a lifecycle-oriented framework.

Building on this foundation, the FORMS reference architecture [49] has served as a lens to explore uncertainty in self-adaptive systems. FORMS breaks a self-adaptive system (SAS) into four different components: the User, the Environment, the Base-Level, which includes the main functionalities of the system, and the Meta-Level, which manages the Base-Level's behavior using a feedback control loop, like MAPE-K [26]. Based on interactions between these components, Esfahani et al. [15] identify uncertainty sources such as *simplified assumptions*, *model drift*, *noise*, *parameters in future operation*, *human in the loop*, *objectives*, *decentralization*, *context*, and *cyber-physical systems*. Rather than proposing new sources, Perez-Palacin et al. [39] classify these sources into a taxonomy designed to highlight similarities. This approach enables the application of shared mitigation methods for similar types of uncertainty. The work by Mahdavi-Hezavehi et al. [32] defines uncertainty sources as circumstances that cause a system to deviate from its expected behavior, similar to *model drift* from Esfahani et al. [15].

Unlike previous works, Bastin et al. [8] categorize the uncertainty into two primary origins: *Model Input* and *Model Structure*, each subdivided into more specific sources. *Model Input* uncertainties include *Measurement*, *Representativity*, *Sensor*

Model, and *Transmission Uncertainties*. *Model Structure* uncertainties encompass *Mechanism*, *Representation*, *Parameter*, and *Numerical Uncertainties*.

The concept of *Indeterminacy*, introduced by Zhang et al. [52] in their Belief Model, describes uncertainty as a lack of confidence in belief statements due to insufficient knowledge originating from *Indeterminacy Sources*. Chatterjee et al. [12] adopt this framework.

Expanding on these ideas, Bandyszak et al. [6] incorporate the notion of uncertainty sources into their Uncertainty Ontology, which they refer to as *Uncertainty Rationale*. Their subsequent work [7] introduces the ECDC model to address uncertainty in real-time computing. This model distinguishes various kinds of uncertainty based on four core concepts: *Execution platform*, which includes the hardware and operating system for running real-time embedded software; *Communication infrastructure*, referring to resources for data exchange and coordination between devices; *Data Processing*, encompassing uncertainties in processing data from sensors or other technical devices; and *Coordination*, addressing uncertainties from collaboration between autonomous systems. This model emphasizes the operational contexts in which uncertainty arises.

Further broadening the scope, Acosta et al. [1] distinguish between system-related and environment-related uncertainty sources, offering an abstract perspective that complements the more detailed frameworks. Lastly, Asmat et al. [3], in a literature review specific to CPS, synthesize earlier approaches, including Zhang et al.'s Belief Model and Bandyszak et al.'s Uncertainty Ontology. Their taxonomy organizes uncertainty causes into three categories: *Human Behavior*, *Natural Processes*, and *Technological Processes*, underscoring the diverse origins of uncertainty in CPS.

3.3 Uncertainty types

The concept of *Uncertainty Types* is included in the model of Zhang et al. [52] encompasses five categories: Occurrence, Environment, Content, Geographical Location, and Time Uncertainty. This model has influenced subsequent work, such as Chatterjee et al. [12] and the PSUM-Metamodel [35], which adopt the concept of uncertainty types.

Troya et al. [46] further explore uncertainty in software models, reviewing various methods for representing it, including Zhang et al.'s Uncertainty Model. Their analysis expands the classification into a taxonomy comprising six uncertainty types: *Spatiotemporal*, *Measurement*, *Occurrence*, *Design*, *Behavior*, and *Belief Uncertainty*. This taxonomy reflects the growing need to address uncertainty in increasingly complex systems and provides new perspectives on its categorization.

One uncertainty type that both Zhang et al. [52] and Troya et al. [46] have in common is *Occurrence Uncertainty*. Zhang et al. describe this as the lack of confidence in the occurrence of events mentioned in a Belief Statement, whereas Troya et al. interpret it as uncertainty about the existence of an entity. Zhang et al. also define Geographical Location Uncertainty and Time Uncertainty, representing a lack of confidence in spatial and temporal aspects, respectively. Troya et al., however, combine these dimensions into a single type, Spatiotemporal Uncertainty, highlighting a more integrated view of uncertainty related to space and time.

Beyond these shared aspects, Zhang et al. [52] identify distinct types of uncertainty, for example, environment and content uncertainty. *Environment Uncertainty* refers to a Belief Agent's lack of confidence in the surroundings of a physical system. *Content Uncertainty* pertains to uncertainty in the details of a Belief Statement's content.

Complementary, Troya et al. [46] introduce additional types of uncertainty that address specific aspects of system modeling. *Measurement Uncertainty* relates to the possible states or outcomes of a measurement, typically associated with probabilities. *Design Uncertainty* captures ambiguity in system design decisions, including uncertainties about user requirements, operating conditions, and potential solutions. *Behavior Uncertainty* involves the unpredictability of system or environmental behavior, encompassing actions, motivations, timing, and parameters. Finally, *Belief Uncertainty* represents uncertainty in any statement made about the system or its environment, reflecting a broad, overarching category that applies to various aspects of modeling and analysis.

3.4 Time

Ramirez et al. [42] propose a template for representing uncertainty in systems, including a *Classification* category that links uncertainty to the phase of the system where it occurs: the *Requirement Level*, which captures uncertainties during requirement setting; *Design-Time*, where uncertainties emerge during the development process; and *Run-Time*, which encompasses uncertainties encountered after deployment. Acosta et al. [1] and Mahdavi-Hezavehi et al. [32] also include a temporal categorization, distinguishing only between *Design-Time* and *Run-Time*. These categorizations capture how uncertainties can manifest and evolve in the lifecycle of a system.

Zhang et al. [52] and Chatterjee et al. [12] introduce an additional perspective by examining the duration of uncertainty and describing its "lifetime": *Temporal Uncertainty* occurs within a specific time interval, and *Persistent Uncertainty*, remains unresolved until specific actions or events take place. This classification provides a dynamic view of uncertainty, capturing its temporal behavior.

3.5 Level

Walker et al. [47] propose a spectrum of knowledge levels to categorize uncertainties, ranging from deterministic knowledge to total ignorance in five levels: *determinism*, *statistical uncertainty*, *scenario uncertainty*, *recognized ignorance*, and *total ignorance*. In this context, *statistical uncertainty* refers to uncertainty that can be characterized using statistical terms, while *scenario uncertainty* arises when only a range of possible outcomes is identifiable, with limited understanding of the mechanisms leading to them. Building on Walker's framework, Esfahani et al. [15] introduce the term *Spectrum of Uncertainty* and redefine the endpoints as *certainty* and *ignorance*, drawing from Aughenbaugh [4]. They add two intermediate states: *Current Information*, representing the present level of knowledge, and *Complete Information*, marking the theoretical limit of knowable facts. Mahdavi-Hezavehi et al. [32] align with

Walker's and Esfahani's definitions but focus primarily on distinguishing *statistical uncertainty* and *scenario uncertainty*, adopting Walker's original terminology.

Furthermore, Zhang et al. [52] differentiate three *Levels of Occurrence* of uncertainty in their Belief Model: the Application Level, resulting from events within the CPS application; the Infrastructure Level, resulting from interactions among physical units; and the Integration Level, which results from interactions between uncertainties either within a level or across different levels.

Perez-Palacin et al. [39] adopt Armour's *orders of ignorance* framework [2], which offers a broader perspective on uncertainty. This approach categorizes uncertainty into five levels: the 0th order (complete certainty), the 1st order (known uncertainty, such as statistical uncertainty), the 2nd order (gaps in knowledge), the 3rd order (unawareness of these gaps and a lack of methods to address them), and the 4th order (meta uncertainty), which reflects uncertainty about the existence of the previous levels. Walker et al. [47] consolidate the higher orders under the term *Total Ignorance*.

3.6 Reducibility

The PSUM-Metamodel [35] introduces the concept of Reducibility Level to classify uncertainty based on its potential for reduction. This dimension provides a way to represent how much uncertainty can be mitigated through additional information or analysis. Fully Reducible Uncertainty refers to situations where uncertainty, although present, can be entirely reduced until full certainty is achieved. Partially Reducible Uncertainty describes uncertainty that can only be reduced to a limited extent. Finally, Irreducible Uncertainty refers to situations where uncertainty cannot be reduced at all.

3.7 Nature

Walker et al. [47], Perez-Palacin et al. [39], Mahdavi-Hezavehi et al. [32] and the PSUM-Metamodel [35] all use the nature of uncertainty to describe uncertainty. The works by Pelz et al. [38] and Troya et al. [46] also explore this dimension, though not as an explicit category.

A consistent theme across these works is the distinction between epistemic and aleatory uncertainty. *Epistemic Uncertainty* emerges due to the imperfection of knowledge or understanding of a phenomenon or system. *Aleatory Uncertainty*, results from the inherent variability of a phenomenon or system. As a result, Mahdavi-Hezavehi et al. [32] refer to it as *Variability Uncertainty* instead.

3.8 Type of model

Uncertainty representation can be tackled in models with different properties, e.g., software or mathematical models. Bertoa et al. [9] present methods for incorporating measurement uncertainty into OCL/UML primitive data types, such as Boolean, Integer, and String. Perez-Palacin et al. [39] use UML diagrams and Markovian models to represent uncertainty in software behavior and performance. These works establish the foundations for uncertainty-aware software engineering models. Pelz et al.[38] use a mechanical engineering perspective by employing mathematical models, such as functions and ordinary differential equations, to address uncertainty in physical processes. In the domain of CPS, Acosta et al. [1] comprises models from both mechanical [38] and software engineering [39]. This work emphasizes the importance of integrating techniques that account for uncertainty across diverse components of these systems.

3.9 Pattern

The characterization by Chipman et al. [13] distinguishes two fundamental patterns of uncertainty: static and dynamic. Static uncertainties describe unknown variables that are constant over time, whereas dynamic uncertainties must be determined for every use since they change to a given law. Consequently, a dynamic uncertainty can be defined as a sequence of multiple static uncertainties.

Building on this distinction, Zhang et al. [52] propose a framework called the *Occurrence Pattern of Uncertainty*, which categorizes how uncertainty arises. Their model identifies two primary types: random uncertainty, which lacks any discernible pattern, and temporal uncertainty, which is characterized by its adherence to temporal patterns. Temporal patterns are further refined into two categories: systematic patterns, which are mathematically predictable and can be either persistent (ongoing indefinitely) or periodic (recurring at regular intervals), and aperiodic patterns, which occur irregularly and are classified as sporadic (occasional) or transient (temporary). The PSUM-Metamodel [35] adopts and integrates these pattern types from Zhang et al.'s framework. Chatterjee et al. [12] expand on Zhang et al.'s model by introducing a third primary type of uncertainty: spatial. Spatial uncertainties depend on location rather than time and differ from temporal uncertainties in that they only manifest systematically, either persistently or periodically.

Further refinement of these concepts is found in Bandyszak et al.'s Uncertainty Ontology [6], which refers to the Occurrence Pattern of Uncertainty as the *Activation Condition*. This term describes the specific circumstances under which uncertainties are triggered during runtime.

3.10 Uncertainty effect

Ramirez et al. [42] introduce the concept of *Impact* in their template to describe uncertainties. This dimension specifically addresses how uncertainty influences the design or execution of a system. Chipman et al. [13] further differentiate between probabilistic and non-deterministic models to capture the nature of uncertainty, and then distinguish between continuous and discrete models to address its structure.

Zhang et al. [52] present a *Measure Model* with three categories to describe the effects of uncertainty: *Ambiguity*, involves measuring uncertainty in terms of ambiguous or imprecise observations; *Probability*, applies probabilistic measures to quantify uncertainty; *Vagueness* uses fuzzy methods or qualitative measures to represent uncertainty. Both Chatterjee et al. [12] and Bandyszak et al. [6] adopt and adjust this model, with Bandyszak et al. framing it under the term *Uncertainty Effect*.

Pelz et al.[38] offer another perspective on representing the effects of uncertainty by introducing a tiered framework. They distinguish between cases where the effect of uncertainty is known versus unknown. When unknown, it is labeled as *ignorance*. When known, further differentiation is made: if the probability of the effect is known, it is called *stochastic uncertainty*; if not, it is called *incertitude*. Acosta et al. [1] align with this classification. However, they diverge in their third classification, suggesting an uncharacterized form of uncertainty to represent cases where further delineation is unnecessary or impractical.

Finally, the PSUM-Metamodel [35] frames the effect of uncertainty as a direct consequence of uncertainty within a belief statement. This perspective emphasizes how uncertainty can impact the interpretation and reliability of such statements, such as when misinterpretation arises due to ambiguous or incomplete information.

3.11 Uncertainty risk

Zhang et al. [52] highlight that not all uncertainties pose the same level of risk. To address this, they employ ISO 31000 [22] to categorize uncertainties into four risk levels: Low, Medium, High, and Extreme. This classification considers both the probability of occurrence and the potential impact, with the latter being assessed through a Risk Matrix [28]. Building on this foundation, Chatterjee et al. [12] and the PSUM-Metamodel [35] integrate this classification system into their frameworks.

3.12 Uncertainty mitigation

Ramirez et al. [42] include a category called *Mitigation Strategies* in their template for uncertainties in dynamically adaptive systems. This category enumerates available techniques to resolve specific sources of uncertainty. Building on this work, the Uncertainty Ontology [6] features a similar concept called *Uncertainty Mitigation*. It comprises methods aimed at proactively preventing uncertainties from arising, rather than merely addressing them once they occur.

Asmat et al. [3] contribute further by proposing a taxonomy based on their literature review of uncertainty sources. Their taxonomy identifies key dimensions where tools and approaches can be applied to manage uncertainty. These dimensions include Physical Units, Heterogeneous Networks, External Entities such as human behavior, and the Physical Environment. These dimensions provide a structure for addressing uncertainties across diverse system components.

3.13 Uncertainty perspective

The PSUM Metamodel [35] introduces the concept of the perspective, which categorizes uncertainty as *subjective* or *objective* based on the viewpoint of observing agents. A subjective perspective arises when uncertainty is shaped by the observations and reasoning processes of the agent, reflecting an interpretation that varies across different observers. In contrast, an objective uncertainty is independent of any specific observer's influence or interpretation.

3.14 Identified gaps and limitations

Our literature review identified several gaps and limitations within the existing research:

Gap 1: Jungle of Terminology Limiting Clarity and Integration: Our literature review revealed that a significant gap lies in the considerable variation in terminology across different frameworks. Many papers use different terms for the same concepts and, conversely, the same terms for different concepts. These inconsistencies create a jungle of terminology, making it difficult to connect, align, and compare approaches effectively. Figure 2 shows how terms for the *Location of Uncertainty* have evolved, highlighting terminological inconsistencies. In total, nine different definitions and five different names of the Location of Uncertainty are used across the related literature. In Fig. 2, all the reused category names are highlighted in the same blue tone. Across these nine definitions, a total of 26 locations are identified, with 15 having different names and 24 varying in meaning. Similar to the category names, all the locations with the same name but different meanings are marked in the same green tone. This variation highlights the significant complexity associated with just one category.

Additionally, such variability can lead to misunderstandings and complicate efforts to establish a unified framework, as readers and researchers must navigate this *jungle of terminology* before making comparisons. This is not only a theoretical issue: the same concept may be labeled "data uncertainty" in one framework and "measurement uncertainty" in another, making it difficult to reuse methods or compare results and ultimately hindering development and safety analysis.

Gap 2: Lack of Explicit Component Differentiation in CPS Frameworks: An additional gap identified in our literature review is the absence of a CPS framework that explicitly differentiates between the distinct component types: the Cyber Component, the Physical Component, and the Platform Component. While some works, such as those by Amsat et al. [3], Acosta et al. [1], and Zhang et al. [52], explore the domain of CPS, none provides a comprehensive framework that addresses these distinct components. Yet each component introduces distinct uncertainties that must be handled differently [40]. For example, sensor noise in a physical component can often be reduced through calibration or filtering [34], whereas execution noise in a platform component must be handled through scheduling and resource control [33]. Without such distinctions, frameworks offer overly generic guidance and fail to support effective mitigation.

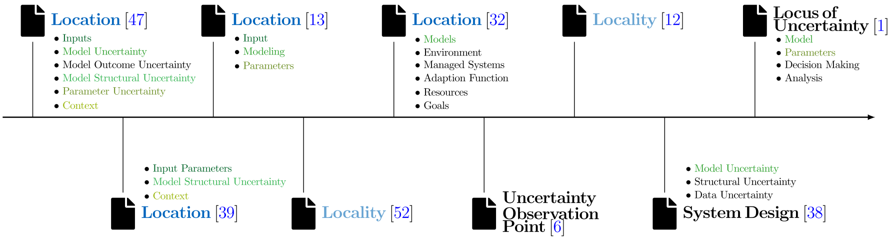

Fig. 2 Overview of terms describing the Location of Uncertainty across various frameworks. Colors indicate terms with identical names but differing meanings across frameworks

Gap 3: Lack of Explicit Autonomy Differentiation in CPS Frameworks: The final gap we identified in our literature review is the absence of a framework that differentiates between autonomous and non-autonomous components within a system. This distinction is essential, particularly for autonomous CPS like self-driving cars, where some components function autonomously while others depend on external control or oversight. Autonomous components, such as perception and planning modules, face uncertainties that differ fundamentally from those in non-autonomous components like actuators or platform infrastructure. [27] shows that autonomous decision-making introduces safety challenges that cannot be addressed with the same approaches used for traditional vehicle components. Without this distinction, frameworks may overlook these differences and fail to provide clear guidance on how to represent, quantify, and mitigate the uncertainties of each component type.

To fill these gaps, we need a harmonizing framework that integrates the different definitions and proposals from the literature into a single representation. We have identified the PSUM Metamodel [35] as a starting point to build such a framework.

4 Harmonizing and extended conceptual model

Building on the gaps and limitations identified in Sect. 3.14, we present our Model for Representing Uncertainty to address these challenges. We adopt the PSUM Metamodel [35], published by OMG in 2023, as our foundation because it contains fundamental uncertainty categories, Effect, Uncertainty Perspective, and Pattern, along with the attributes Kind, Level, and Nature. These categories provide a precise and interoperable basis, reducing the terminological inconsistencies discussed in Sect. 3.14 and ensuring compatibility with related domains. However, PSUM is domain-independent and does not capture CPS-specific aspects such as component differentiation or autonomy. Our contribution is to extend PSUM with CPS-specific concepts, ensuring its applicability to CPS.

Figure 3 provides an overview of the model. Concepts in white are adopted from the PSUM Metamodel, those in blue are from existing literature discussed in Sect. 3, and the green concepts represent novel contributions introduced in this work.

We address the limitations of existing uncertainty categorizations by providing a unified definition of each category, along with its options and connections to other categories in Section 4. Additionally, we introduce novel categories to address the limitations identified in the current uncertainty categorizations. In Section 4, we present categories that partition the complete CPS into smaller components, enabling a fine-grained characterization of uncertainty within the CPS. In Section 4, we introduce a category that differentiates between autonomous and non-autonomous components.

4.1 Kind

To classify the manifold types of uncertainties in the real world, we take the name of the category *Uncertainty Kind* and its position as one of the three uncertainty attributes from the PSUM Metamodel [35]. For specific types, we incorporate the

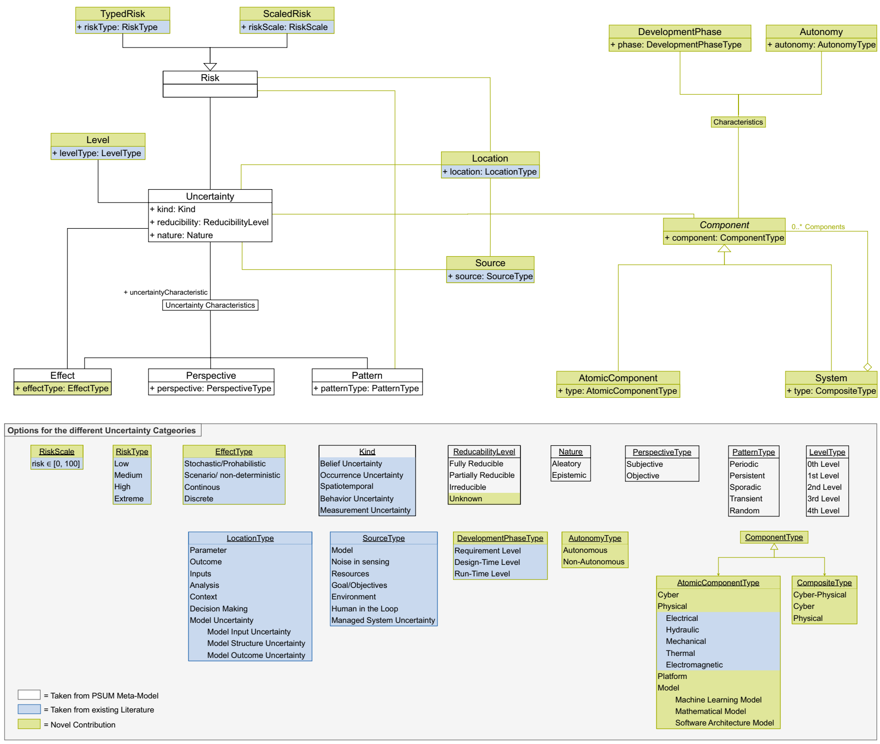

Fig. 3 Harmonized and Extended Conceptual Uncertainty Representation Model

most relevant kinds identified by Troya et al. [46], including Belief-, Occurrence-, Behavior-, Measurement-, and Spatiotemporal-Uncertainty.

4.2 Reducibility

Understanding whether uncertainty can be reduced plays a critical role in determining how it should be addressed. To capture this dimension, we adopt the Reducibility Level category from the PSUM Metamodel [35]. Within this framework, Reducibility is defined as an attribute of uncertainty and is classified into three levels: Fully Reducible, Partially Reducible, and Irreducible. We introduce an additional category, Unknown, to capture cases where the reducibility of uncertainty cannot be determined.

4.3 Nature

Following the PSUM Metamodel framework [35], the final uncertainty attribute is the Nature of Uncertainty. The literature consistently defines this category and distinguishes between two types: epistemic and aleatory.

4.4 Effect

Once uncertainty is present in a system, it can significantly affect behavior, influence its operations, outcomes, and decision-making processes. This influence can manifest in various ways, all of which are captured in the category Uncertainty Effect.

In alignment with the PSUM Metamodel framework [35], we treat the Uncertainty Effect as a characteristic of uncertainty. To represent this category, we adopt the approach proposed by Chipman et al. [13]. Their framework defines four categories, each offering mathematical methods to quantify the impact of uncertainty. These categories cover all relevant concepts identified in the literature and are structured around two key distinctions: continuous vs. discrete and non-deterministic vs. probabilistic.

4.5 Perspective

Uncertainty can be viewed from different perspectives. For example, some uncertainties are identifiable through data analysis, while others can only be recognized through personal observations. To capture this distinction, we adopt the Uncertainty Perspective category from the PSUM Metamodel [35] as the next characteristic of uncertainty.

4.6 Pattern

Once it is established that uncertainty is present in a CPS, an important question is when this uncertainty arises. To address this, we use the Uncertainty Pattern category, which is the final uncertainty characteristic in the PSUM Metamodel [35]. The Pattern of Uncertainty provides a framework for understanding how uncertainty manifests over time.

To capture the various patterns of uncertainty occurrence, we adopt the classifications proposed by Zhang et al. [52].

4.7 Level

Not all uncertainties have the same level of knowledge associated with them. Some are well understood, while others lack sufficient knowledge or methods for discovery. To address these differences, we classify them into levels. We adopted the approach proposed by Perez Palacin [39], as it captures the core ideas of other approaches in a clear and accessible manner. Instead of using the term "orders of uncertainty" from the original framework, we refer to these categories as levels of uncertainty.

4.8 Risk

When uncertainty is introduced into a system, it can significantly affect its behavior, influencing its operations, outcomes, and decision-making processes. Given that systems often rely on expected functionality, any deviation in one system's functionality can impact others and, in the worst case, lead to system failure. Thus, uncertainty inherently introduces risk, which we assess using two complementary metrics: Risk Type and Risk Scale.

The Risk Type is adapted from Zhang et al. [52], which builds upon ISO 31000 [22] to classify uncertainties into four levels: low, medium, high, and extreme.

The numerical Risk Scale is a normalized indicator that combines likelihood and severity by multiplying the probability of occurrence (0–1) with a severity ranking (1–10) and scaling the result to a 0–100 range. This simple approach aligns with established practices such as FMEA [10] and ISO 26262 [37] while remaining easy to apply across CPS domains For example, a component failure with probability 0.05 and severity 8 yields a risk score of 40.

4.9 Location

Understanding the location within a system where uncertainty is observed is important, as it provides valuable context for analyzing and managing the uncertainty effectively. For instance, the location of uncertainty can significantly influence the associated risk. To address this, we identified the framework of Walker et al. [47] and Acosta et al. [1] as the most suitable for representing uncertainty locations in CPS. Their focus on categorization through decision-based and coupled models highlights key aspects that significantly influence the CPS development. Based on these frameworks, we define seven categories: *Parameter*, *Outcome*, *Inputs*, *Analysis*, *Context*, *Decision Making*, and *Model Uncertainty* which can be further split up into *Model Input Uncertainty* and *Model Structure Uncertainty*. For Parameter Uncertainty, Outcome Uncertainty, and Input Uncertainty, we adopt the definition provided by Walker et al. [47]. From Acosta et al. [1], we adopt the definitions of Analysis Uncertainty and Decision-Making Uncertainty. The definitions of Context and Model Uncertainty follow Walker et al. [47].

4.10 Source

Uncertainty Location refers to the part of a system where uncertainty is observed. Yet, the observed location is not always the root cause of the uncertainty. To address this distinction, we introduce the category of Uncertainty Source, which represents the origin of the uncertainty. The relationship between sources and locations is not one-to-one: a single Uncertainty Source can affect multiple locations, and conversely, a single Uncertainty Location can be influenced by multiple sources, mediated through different uncertainty instances.

We highlight a selection of common sources of uncertainty, or classes of such sources, primarily based on the classification by Mahdavi-Hezavehi et al. [32]. This list is not exhaustive, as identifying all sources is domain-specific and remains future work. Tailored to CPS in general, we focus on the following five sources: Model, Resources, Goal/Objectives, Environment, and Managed System Uncertainty, all based on the descriptions by Mahdavi-Hezavehi et al. [32].

4.11 Relating uncertainty to CPS components

Identifying the part of the system where uncertainty manifests and is observed, i.e., the Uncertainty Location, as well as the source from which it arises, i.e., the Uncertainty Source, requires a concrete description of the relevant parts in the CPS rather than viewing the CPS as a whole.

To effectively represent the uncertainty aspects in concrete CPS parts while engineers can keep their autonomy to decide the appropriate modeling granularity for each CPS part, we apply the Composite pattern [17]. The types of elements that represent the CPS following the composite pattern are called the Atomic Component, which is the finest modeling granularity in the CPS, and the System, which is the composite entity that can contain other Systems and Atomic Components.

4.11.1 Atomic component

An atomic component represents the finest granularity unit of a CPS considered in our representation model. For CPS, the possible atomic component types are: Cyber, Physical, Platform parts, and the Model type.

The Physical Part of a CPS includes components that interact directly with the environment, enabling the system to engage with physical processes. To refine the Physical Part, we follow the approach proposed by Karsai [24] and classify it into the following categories: Electrical, Hydraulic, Mechanical, Thermal, and Electromagnetic.

The Cyber Part refers to the digital and computational components responsible for processing, analysis, decision-making, and control within the system.

A critical aspect of a CPS is the Platform Part, which facilitates interaction between the two other components. This interaction often introduces additional complexity, and thus potential uncertainty, into the system. To emphasize its significance, the interaction platform is treated as a separate atomic component type, as it plays a key role in influencing overall system behavior and uncertainty propagation.

Our model also allows the representation of the uncertainty related to the various types of models used in the CPS. The types of models include Mathematical Models, Software Architecture Models, and Machine Learning Models, which represent diverse approaches to analyzing system behavior. It is important to note that this list reflects current observations from the reviewed literature but remains incomplete and open to future extensions.

4.11.2 System

While atomic components provide the fundamental building blocks of a CPS, some functionalities require more complex groupings of these building blocks. Hence, we introduce the System entity as the aggregator of other entities. This hierarchical structure of the Composite pattern allows a System to be viewed as a CPS in its own right. Consequently, we introduce a new composite type, the Cyber-Physical, to capture the combined nature of cyber and physical elements at the entity container level.

At the root of the composite structure, the System entity represents the whole CPS, which is composed of other lower granularity Systems and or Atomic Components.

4.12 Development phase

Following the literature, we also introduce a category that represents the time when uncertainties can arise. To tailor this concept more specifically to CPS, we renamed this category the Development Phase of a CPS, as it captures the stages of development where uncertainties tend to occur more accurately. Following the differentiation proposed by Ramirez et al. [42], we incorporate three development phases in our unified model: Requirement-Level, Design-Time, and Run-Time.

4.13 Autonomy

CPS are increasingly integrating both autonomous and non-autonomous components. Since these parts have fundamentally different characteristics, uncertainties must be handled according to their respective status. To address this, differentiating between autonomous and non-autonomous parts becomes a crucial addition to uncertainty representation. Autonomous parts are capable of making decisions independently, often relying on internal models, real-time sensor data, and adaptive mechanisms. In contrast, non-autonomous parts operate based on explicit commands or inputs from external controllers. We assign this characteristic to each component to enable flexibility in defining the autonomy level of system parts. This approach allows both atomic components and entire systems to be classified as either autonomous or non-autonomous.

4.14 Summary of contributions

This section summarizes how we addressed the gaps and limitations identified in Sect. 3.14 and highlights the novel parts of our harmonized conceptual model.

Contribution 1: Systematic Refinement of Terminology for Uncertainty Representation in CPS. The first gap we identified is the inconsistency in terminology, where numerous overlapping or redundant terms complicate the understanding of uncertainty categorization. We tackled this issue by systematically analyzing each term and concept by identifying equivalences, overlaps, and hierarchical relationships. This process ensured that only the essential terms and concepts remained while still preserving a comprehensive overview. For instance, in the category of Uncertainty Location, Fig. 2 illustrates that existing literature provides nine different definitions and 26 different locations. Through our analysis, we consolidated these into a single definition with 10 distinct locations, as illustrated in Fig. 4. The timeline in Fig. 2 has been extended to incorporate our proposed categorization, highlighted in orange, making the reduction in redundancy and the improved structure clearly visible.

Contribution 2: Structured Classification of CPS Components for Uncertainty Representation. Addressing the gap of a systematic CPS component classification, Section 4 introduces a structured framework that distinguishes between Cyber, Physical, and Platform components. Unlike existing approaches that consider CPS as a

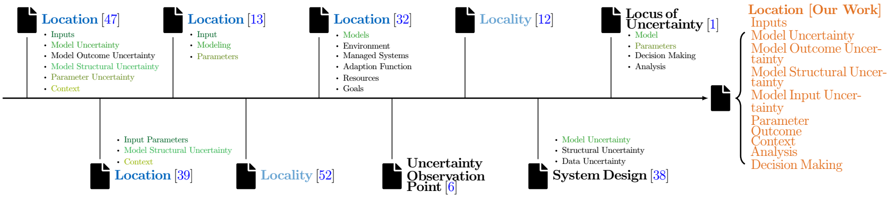

Fig. 4 Overview of terms describing the Location of Uncertainty across various frameworks. Colors indicate terms with identical names but differing meanings across frameworks, while orange highlights the introduced consistent terminology

monolithic system, our model introduces a hierarchical decomposition into Atomic Components, Components, and Systems. This structure enables precise localization of uncertainty by linking it directly to specific CPS elements rather than relying on a generic classification. Additionally, we introduce the Platform Component as a distinct entity to capture interactions between Cyber and Physical elements, a crucial yet often overlooked aspect in uncertainty representation.

Contribution 3: Classification of Autonomous and Non-Autonomous Components in CPS. The distinction between autonomous and non-autonomous components introduced in Section 4 extends the CPS characterization and addresses the final gap identified in our literature review. Autonomous components operate independently using internal models and real-time data, while non-autonomous components rely on external commands. This refinement enhances uncertainty representation by capturing the differing decision-making properties within CPS.

5 Conceptual model application example

We illustrate the application of the conceptual model to a CPS, using the Autonomous Vehicle (AV) as a CPS example. Since the AV includes a multitude of components, we concentrate on the model of a set of components in the self-driving pipeline. Among other components, the self-driving system pipeline includes components for data *Acquisition*, *Perception* of objects, traffic signs, pedestrians, etc., and *Actuation* on the vehicle physical devices such as steering, throttle, brakes, etc. We assume that the *Perception* system is purely software, and therefore, the componentType is *cyber*; the *Acquisition* componentType is cyber-physical since it includes the physical devices that capture data from the real world as well as the digital representation of such data; while the *Actuation* system components are purely physical. Figure 5 shows the representation of subsystems and atomic components of the AV instantiating the model in Fig. 3.

Regarding the Autonomy associated with Components, The *Actuation* and *Perception* systems are *Autonomous*. For vehicles that are not possible to actuate manually (e.g., autonomous delivery robots that are not possible to steer manually but move based on their defined destination), the *Actuation* system is *Autonomous* too. However, if we assume that the AV is one of the current cars with some self-driving capabilities, the *Actuation* system is *non-autonomous* since it demands oversight and external control sometimes. In that case, the *non-autonomous* Autonomy of the

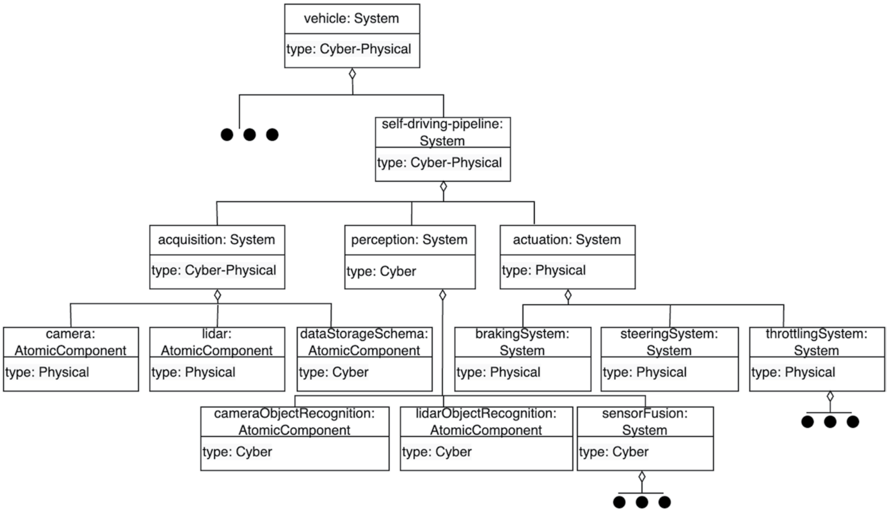

Fig. 5 Model of a selection of components in the autonomous vehicle

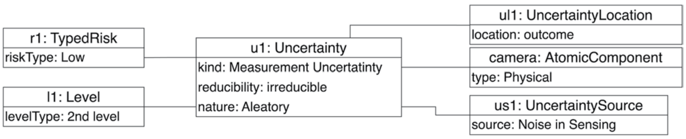

Fig. 6 Model of a selection of uncertainties in the camera component

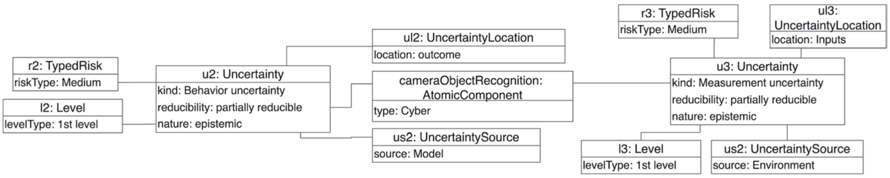

Fig. 7 Model of a selection of uncertainties in the object recognition from the camera information

Actuation system propagates upstream in the composite structure [17]. Regarding the DevelopmentPhase, the illustrated example concentrates on the *Run-Time level*.

Figures 6 and 7 illustrate the information of a few uncertainties associated with Components. Figure 6 represents the uncertainty regarding the noise captured by the camera component. It models that it is a *Measurement Uncertainty* since it comes from capturing information from reality, and hence its source of the uncertainty is the *Noise in Sensing*. The location of the uncertainty lies in the camera component output. We assume that the camera is unaware of its uncertain behavior and does not provide any information about its noise in the images captured, then getting assigned a 2nd level. As typically happens with the noise, it is assumed to be aleatory, and this uncertainty about the reality values cannot be eliminated with more data. Since the components that use the camera as input are well aware of the existing noise when acquiring images with cameras, and they behave according to the existence of noise, the risk caused by the noise in sensing in the camera is considered *Low*.

In turn, Fig. 7 shows two uncertainties associated with the cameraObjectRecognition Component. The first uncertainty, named *u2* in the figure, refers to the imperfection of the algorithms that classify elements in images. Therefore, its kind is the *Behavior Uncertainty*, and its location is the output of the object classification algorithms. Since it is possible to reduce the uncertainty in the classification by providing more training data or a better training set, the uncertainty reducibility is *partially reducible* and its nature is *epistemic*. We assign *1st level* to this uncertainty because, in modern systems, there is awareness that automated classifications are not always perfect and mitigation techniques are implemented. For example, our example already considered a sensorFusion subsystem that combines results from the object recognition software and aims at mitigating the impact of their imperfect classifications. Since the sensorFusion exists, we assigned only a *Medium* risk to the cameraObjectRecognition. However, if this program was the only authority to classify elements on the road, its assigned risk would have been higher. To show how a component may hold more than one uncertainty, the right part of the figure shows an additional associated uncertainty, called *u3*. This uncertainty refers to the quality of the *Inputs* that cameraObjectRecognition receives. Images may be of low quality due to the presence of dust on the camera, or because an element in the environment partially occludes the actual scene to process, or others. Therefore, the nature of uncertainty *u3* is epistemic, as it originates from incomplete knowledge about the observed scene and can be partially reduced through improved positioning of the camera.

6 Conclusion and future directions

In this paper, we have conducted a literature review of uncertainties in CPS and identified some gaps in the current classification schemes. We then proposed a new uncertainty representation model that addresses limitations in previous classifications and tries to overcome the "jungle of terminology" problems. Our work has highlighted the multifaceted nature of uncertainties in CPS and paved the way for future research in this domain.

Based on our research, we conclude that incorporating the concept of uncertainty from the outset of software design is crucial for the engineering of future CPS. This approach would facilitate the development of CPS that are inherently aware of uncertainty, guided by a set of design principles and methodologies that intrinsically address various forms of unpredictability. Furthermore, this perspective opens up several promising research avenues that warrant exploration to achieve this overarching objective. These directions are briefly described below.

6.1 Shared understanding

Our work is a first step towards a unified uncertainty terminology. Additional work in this direction is necessary to provide a common basis for uncertainty representation, measurement, and reporting across different CPS domains, facilitating better communication and comparison of results.

6.2 Quantification methods

The resulting model and the different facets of uncertainty highlighted emphasize the need to investigate and develop sophisticated techniques to quantify and measure various types of uncertainties in CPS, particularly those that are difficult to capture using traditional probabilistic approaches.

6.3 Propagation and Interaction

The complexity of the CPS architecture requires investigation of how uncertainties propagate through the different components and how uncertainties in one domain (e.g., cyber) affect uncertainties in another (e.g., physical). This study would also lead to the development of new comprehensive mitigation strategies.

6.4 Human-system interaction

The increasing diffusion of CPS and their usage together with humans requires an in-depth investigation of how human-system interactions affect and /or mitigate uncertainties in CPS. This, in turn, would require the development of methods for effective human–machine collaboration under uncertainty.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by all authors. The first draft of the manuscript was written by J.M., and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was supported by the German Research Foundation (DFG) – SFB 1608 – Project-ID 501798263.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Acosta M, Hahner S, Koziolek A, et al (2022) Uncertainty in coupled models of cyber-physical systems. In: Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. ACM, pp 569–578
2. Armour PG (2000) The five orders of ignorance. Commun ACM 43(10):17–20
3. Asmat MN, Khan SUR, Hussain S (2023) Uncertainty handling in cyber-physical systems: State-of-the-art approaches, tools, causes, and future directions. Journal of Software Evolution and Process 35(7):e2428
4. Aughenbaugh JM (2006) Managing Uncertainty in Engineering Design Using Imprecise Probabilities and Principles of Information Economics. Georgia Institute of Technology
5. Baheti R, Gill H (2011) Cyber-physical systems. The Impact of Control Technology 12(1):161–166
6. Bandyszak T, Daun M, Tenbergen B et al (2020) Orthogonal uncertainty modeling in the engineering of cyber-physical systems. IEEE Trans Autom Sci Eng 17(3):1250–1265
7. Bandyszak T, Weyer T, Daun M (2022) Uncertainty theories for real-time systems. In: Handbook of Real-Time Computing. Springer, p 99–132
8. Bastin L, Cornford D, Jones R et al (2013) Managing uncertainty in integrated environmental modelling: The uncertweb framework. Environmental Modelling & Software 39:116–134
9. Bertoa MF, Burgueño L, Moreno N et al (2020) Incorporating measurement uncertainty into ocl/uml primitive datatypes. Softw Syst Model 19(5):1163–1189
10. Carbone TA, Tippett DD (2004) Project risk management using the project risk fmea. Eng Manag J 16(4):28–35
11. Chakraborty S, Al Faruque MA, Chang W et al (2016) Automotive cyber-physical systems: A tutorial introduction. IEEE Design & Test 33(4):92–108
12. Chatterjee A, Reza H (2020) Toward modeling and verification of uncertainty in cyber-physical systems. In: 2020 IEEE International Conference on Electro Information Technology (EIT). IEEE, pp 568–576
13. Chipman W, Grimm C, Radojicic C (2015) Coverage of uncertainties in cyber-physical systems. In: ZuE 2015; 8. GMM/ITG/GI-Symposium Reliability by Design. VDE, pp 1–8
14. Cuzzolin F (2024) Uncertainty measures: A critical survey. Information Fusion p 102609
15. Esfahani N, Malek S (2013) Uncertainty in self-adaptive software systems. In: Software Engineering for Self-Adaptive Systems II: International Seminar, Dagstuhl Castle, Germany, October 24–29, 2010 Revised Selected and Invited Papers. Springer, pp 214–238
16. Fakour F, Mosleh A, Ramezani R (2024) A structured review of literature on uncertainty in machine learning & deep learning. arXiv:2406.00332 accessed: 2025-09-16
17. Gamma E, Helm R, Johnson R et al (1994) Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, USA
18. Gawlikowski J, Tassi CRN, Ali M et al (2023) A survey of uncertainty in deep neural networks. Artif Intell Rev 56(Suppl 1):1513–1589
19. Guo Z, Wan Z, Zhang Q, et al (2022) A survey on uncertainty reasoning and quantification for decision making: Belief theory meets deep learning. arXiv:2206.05675, accessed: 2025-09-16
20. Haque SA, Aziz SM, Rahman M (2014) Review of cyber-physical system in healthcare. Int J Distrib Sens Netw 10(4):217415
21. Haugen M, Farahmand H, Jaehnert S, et al (2023) Representation of uncertainty in market models for operational planning and forecasting in renewable power systems: a review. Energy Systems pp 1–36
22. International Organization for Standardization (2009) Iso 31000: Risk management. https://www.iso.org/standard/43170.html, accessed: 2024-11-07
23. Kamal A, Dhakal P, Javaid AY et al (2021) Recent advances and challenges in uncertainty visualization: a survey. J Visualization 24(5):861–890
24. Karsai G (2015) Modeling cyber-physical systems: Challenges and recent advances. In: Seminar at University of Connecticut
25. Keith AJ, Ahner DK (2021) A survey of decision making and optimization under uncertainty. Ann Oper Res 300(2):319–353
26. Kephart JO, Chess DM (2003) The vision of autonomic computing. Computer 36(1):41–50
27. Koopman P, Wagner M (2017) Autonomous vehicle safety: An interdisciplinary challenge. IEEE Intell Transp Syst Mag 9(1):90–96

28. Lansdowne ZF (1999) Risk matrix: an approach for prioritizing risks and tracking risk mitigation progress. Proceedings of the 30th Annual Project Management Institute, Philadelphia, PA, October pp 10–16
29. Lee EA (2008) Cyber physical systems: Design challenges. In: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). IEEE, pp 363–369
30. Lee EA (2015) The past, present and future of cyber-physical systems: A focus on models. Sensors 15(3):4837–4869
31. Li Y, Chen J, Feng L (2012) Dealing with uncertainty: a survey of theories and practices. IEEE Trans Knowl Data Eng 25(11):2463–2482
32. Mahdavi-Hezavehi S, Avgeriou P, Weyns D (2017) A classification framework of uncertainty in architecture-based self-adaptive systems with multiple quality requirements. In: Managing Trade-Offs in Adaptable Software Architectures. Elsevier, p 45–77
33. Minaeva A, Akesson B, Hanzálek Z et al (2017) Time-triggered co-scheduling of computation and communication with jitter requirements. IEEE Trans Comput 67(1):115–129
34. Nikitin AV, Davidchack RL (2019) Hidden outlier noise and its mitigation. IEEE Access 7:87873–87886
35. Object Management Group (OMG) (2023) Precise semantics for uncertainty modeling (psum), version 1.0. Object Management Group (OMG) Specification, https://www.omg.org/spec/PSUM/1.0/ Accessed 16 Jul 2025
36. Oztemel E, Gursev S (2020) Literature review of industry 4.0 and related technologies. J Intell Manuf 31(1):127–182
37. Palin R, Ward D, Habli I, et al (2011) Iso 26262 safety cases: Compliance and assurance. In: 6th IET International Conference on System Safety 2011. IET, p B12
38. Pelz PF, Groche P, Pfetsch ME, et al (2021) Mastering Uncertainty in Mechanical Engineering. Springer Nature
39. Perez-Palacin D, Mirandola R (2014) Uncertainties in the modeling of self-adaptive systems: a taxonomy and an example of availability evaluation. In: Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering. ACM, pp 3–14
40. Pinto A (2023) Analysis and design of uncertain cyber-physical systems. In: Computation-Aware Algorithmic Design for Cyber-Physical Systems. Springer, p 25–53
41. Quan H, Khosravi A, Yang D et al (2019) A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids. IEEE Transactions on Neural Networks and Learning Systems 31(11):4582–4599
42. Ramirez AJ, Jensen AC, Cheng BHC (2012) A taxonomy of uncertainty for dynamically adaptive systems. In: 2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). IEEE, pp 99–108
43. Reddy YB (2015) Security and design challenges in cyber-physical systems. In: 2015 12th International Conference on Information Technology - New Generations. IEEE, pp 200–205
44. Shorinwa O, Mei Z, Lidard J, et al (2024) A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. arXiv:2412.05563 Accessed 16 Jul 2025
45. Tao X, Broo DG, Törngren M, et al (2020) Uncertainty management in situation awareness for cyber-physical systems: State of the art and challenge. In: Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence. ACM, pp 424–430
46. Troya J, Moreno N, Bertoa MF et al (2021) Uncertainty representation in software models: a survey. Softw Syst Model 20(4):1183–1213
47. Walker WE, Harremoës P, Rotmans J et al (2003) Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support. Integr Assess 4(1):5–17
48. Wang F, Liu Y, Liu K, et al (2024) Uncertainty in graph neural networks: a survey. arXiv:2403.07185, accessed: 2025-09-16
49. Weyns D, Malek S, Andersson J (2010) Forms: a formal reference model for self-adaptation. In: Proceedings of the 7th International Conference on Autonomic Computing. ACM, pp 205–214
50. Weyns D, Calinescu R, Mirandola R et al (2023) Towards a research agenda for understanding and managing uncertainty in self-adaptive systems. ACM SIGSOFT Software Engineering Notes 48(4):20–36
51. Yu X, Xue Y (2016) Smart grids: A cyber-physical systems perspective. Proc IEEE 104(5):1058–1070

52. Zhang M, Selic B, Ali S, et al (2016) Understanding uncertainty in cyber-physical systems: a conceptual model. In: Modelling Foundations and Applications: 12th European Conference, ECMFA 2016, Held as Part of STAF 2016, Vienna, Austria, July 6–7, 2016, Proceedings. Springer, pp 247–264
53. Zio E, Pedroni N (2013) Literature review of methods for representing uncertainty. Tech. rep, FonCSI

Authors and Affiliations

Johannes Mäkelburg[1] · Diego Perez-Palacin[2,3] · Raffaela Mirandola[2] · Maribel Acosta[1]

✉ Johannes Mäkelburg
johannes.maekelburg@tum.de

Diego Perez-Palacin
diego.perez@lnu.se

Raffaela Mirandola
raffaela.mirandola@kit.edu

Maribel Acosta
maribel.acosta@tum.de

1 Technical University of Munich, Munich, Germany

2 Karlsruhe Institute of Technology, Karlsruhe, Germany

3 Linnaeus University, Växjö, Sweden