Toward Modeling Trust Cyber-Physical Systems: A Model-based System Engineering Method

Zina Oudina[1], Makhlouf Derdour[2]
Computer Science Department, University of Badji Mokhtar Annaba, Annaba, Algeria[1]
Computer Science Department, University of Oum El Bouaghi, Oum El Bouaghi, Algeria[2]

***Abstract*—Developing trust in cyber-physical systems (CPSs) is a challenging task. Trust in CPS is needed for carrying out their intended duties and is reasonably safe from misuse and intrusion; it also enforces the applicable security policy. As an example, medical smart devices, many researches have found that trust is a key factor in explaining the relationship between individual beliefs about technological attributes and their acceptance behavior; and have associated medical device failures with severe patient injuries and deaths. The cyber-physical system is considered a trust system if the principles of security and safety, confidentiality, integrity, availability, and other attributes are assured. However, a lack of sufficient analysis of such systems, as well as appropriate explanation of relevant trust assumptions, may result in systems that fail to completely realize their functionality. The existing research does not provide suitable guidance for a systematic procedure or modeling language to support such trust-based analysis. The most pressing difficulties are achieving trust by design in CPS and systematically incorporating trust engineering into system development from the start of the system life cycle. Still, there is a need for a strategy or standard model to aid in the creation of a safe, secure, and trustworthy CPS. Model-based system engineering (MBSE) approaches for trust cyber-physical systems are a means to address system trustworthiness design challenges. This work proposes a practical and efficient MBSE method for constructing trust CPS, which provides guidance for the process of trustworthiness analysis. The SysML-based profile is supplied, together with recommendations on which approach is required at each process phase. The MBSE method is proven by expanding the autonomous car SysML and UML diagrams, and we show how trust considerations are integrated into the system development life cycle.**

Keywords*—Cyber Physical Systems (CPSs); trust CPS; system engineering (SE); model-based system engineering (MBSE); SysML

I. INTRODUCTION

The emergence of cyber-physical systems (CPSs) has an impact on people's lives and is used in health care, smart homes, commerce, and other areas. Engineering CPS necessitates a combination of approaches from several fields (mechanical, electrical, biological, etc.)[1] and computer science methods.

The cyber-physical system is considered a trust system if the principles of security and safety, confidentiality, integrity, availability, and other attributes are ensured. They defined trust in [2] as a measure of confidence or belief that the other party will refrain from opportunistic behavior and behave in an expected manner. The faith CPS is a system that meets a set of requirements and obligatory attributes that ensure the trustworthiness of CPS [3].

Inattention to trustworthiness can result in the loss of human life, long-term environmental implications, the disruption of essential infrastructure, or other catastrophic consequences such as the revelation of sensitive data, the destruction of equipment, economic loss, and reputational harm. These dangers and negative outcomes become more severe as industries become more networked and integrated.

In the literature, many directions are suggested for modeling trust in systems. However, does not provide suitable instructions for a systematic methodology or an acceptable modeling language to support such trust-based analysis. The most significant problems are achieving trust by design in CPS and systematically incorporating trust engineering into system development from the start of the system life cycle.

According to [4], a user-centered holistic technique is described for analyzing trust as the system is being developed. The process consists of five steps (the scenarios, trust analysis, peer review, scenarios refinement, guiding the design). Other lines of research target improving system trustworthiness in terms of security. They consider trust an enabler of security, and security services rely to a great extent on some notion of trust [5]. In [6], a trust case is presented with a complete and explicit argument for trust in the system under development in terms of security and safety. They used the Claim Definition Language (CDL). For trustworthy computing [7], it employs the Security Development Lifecycle (SDL) for the development of software that must withstand attacks. UMLsec [8] is a Unified Modeling Language (UML) extension and it is a UML profile that allows designers to define security features on design models. To establish the security requirements and assumptions on the system's environment, typical UML extension techniques in the form of labels, such as stereotypes and tags, are used. Trust cases that focus on the security and safety of the system are described in [6], excluding additional trust attributes such as privacy and usability, and it enables the identification of the components that are engaged in trust assumptions.

The specific challenges and limitations of existing research are about identifying trust concerns and requirements that may apply to many parties, as well as focusing purely on security quality and viewing trust as an enabler of security and security services while being unconcerned about the remaining trust

attributes; furthermore, selecting modeling approaches that can handle the complexity of trust in CPS.

Not defining a cyber-physical system's tasks. Not specifying the exact characteristics of trust in CPS. Not defining the targets of the actors and participants in the CPS, including developers, users, and potential customers. Not defining goals, especially business goals. The unpredictability of the behavior of the system component, all those are the major challenges to the development of trust in cyber-physical systems, particularly during the modeling stage.

In this work, we examine specific strategies, models, and tools that encompass design activities and preparation for modeling the trustworthiness of CPSs, as well as how these techniques can be expressed in Unified Modeling Language (UML) and SysML models. The MBSE approach and phases are discussed. By applying the autonomous vehicle SysML and UML diagrams, the MBSE method is validated.

The rest of the paper is organized as follows: Section II presents the contribution and research methodology. Model based system engineering (MBSE) is discussed in Section III. Proposed MBSE Trust CPS Method is introduced in Section IV. Discussion is given in Section V. We conclude the paper in Section VI.

II. Contribution and Methodology

This work aims to:

- Analyze specific approaches, models, and tools pertaining to design processes and preparation for modeling the trustworthiness of CPS.
- Create a viable and efficient MBSE method for developing secure systems while adhering to trust constraints.

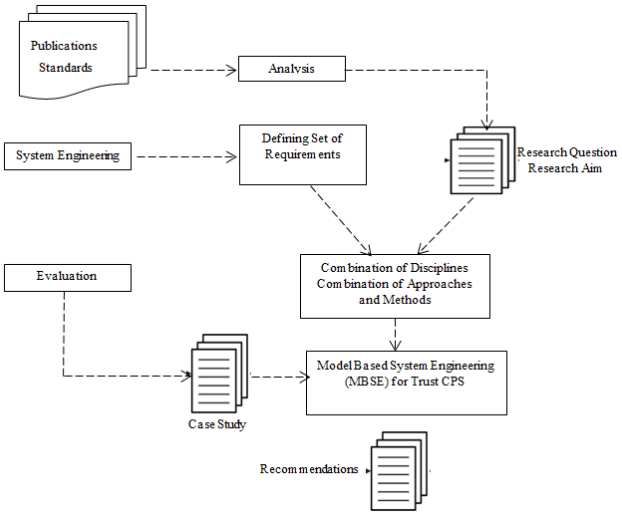

Fig. 1. Research methodology.

- Ensure the system's trustworthiness characteristics are created in accordance with end-user trust requirements.

Research Question:

RQ1- How Infusing trust attributes in CPS into systems engineering practice to decrease the complexity and increase the quality and trustworthiness of CPS?

RQ2- What are the methods and tools shall be used for modeling trust CPS?

Our methodology was based on the analysis of fund papers from publications and standards and the definition of requirements and mandatory attributes using system engineering and requirements approaches. Model-Based System Engineering (MBSE) for Trust CPS is a combination of disciplines and approaches, and methods. In Fig. 1, the research method is displayed.

III. Model Based System Engineering (MBSE)

Model-Based Systems Engineering (MBSE) is a tool used to run simulations, support system engineering operations, and improve requirements throughout process development. MBSE offers various advantages, including increased mapping, traceability, and system decomposition [9], improved communication and system management [10], complexity management and risk reduction [11], and systemized decision making [12].

MBSE is a modeling formalism used to assist system requirements, design, analysis, verification, and validation activities, beginning with the conceptual design phase and progressing through the development and subsequent life cycle phases [13, 14, 15]. According to [16], the use of MBSE models reduces project risks and timelines, reduces costs, and improves product quality. Additionally, the MBSE approach's primary idea of using simulation models assures that results are acquired faster and more affordably than testing and prototyping. MBSE has become an essential component of designing complex cyber-physical systems [17,18], is popularly used [19] for: a) Capture and manage the system's requirements, architecture, design, and identify its environment. b) Ease the communication among many stakeholders by participating and providing views for different purposes. In [20], they stated that the benefit of MBSE activities is system validation and verification throughout the early stages of system design. The use of MBSE is limited to object modeling via the Systems Modeling Language (SysML) [21].

IV. Proposed MBSE Trust CPS Method

This section presents the MBSE technique, which includes activities and principles for developing trust CPS. Our goal with MBSE is to obtain descriptive models that leverage semantically rich modeling standards to provide systems abstraction, data traceability, and separation of views. The benefits of using MBSE are explained in previous section. The main goal is: How infusing trust's attributes of CPS into systems engineering practice to decrease the complexity of trustworthiness.

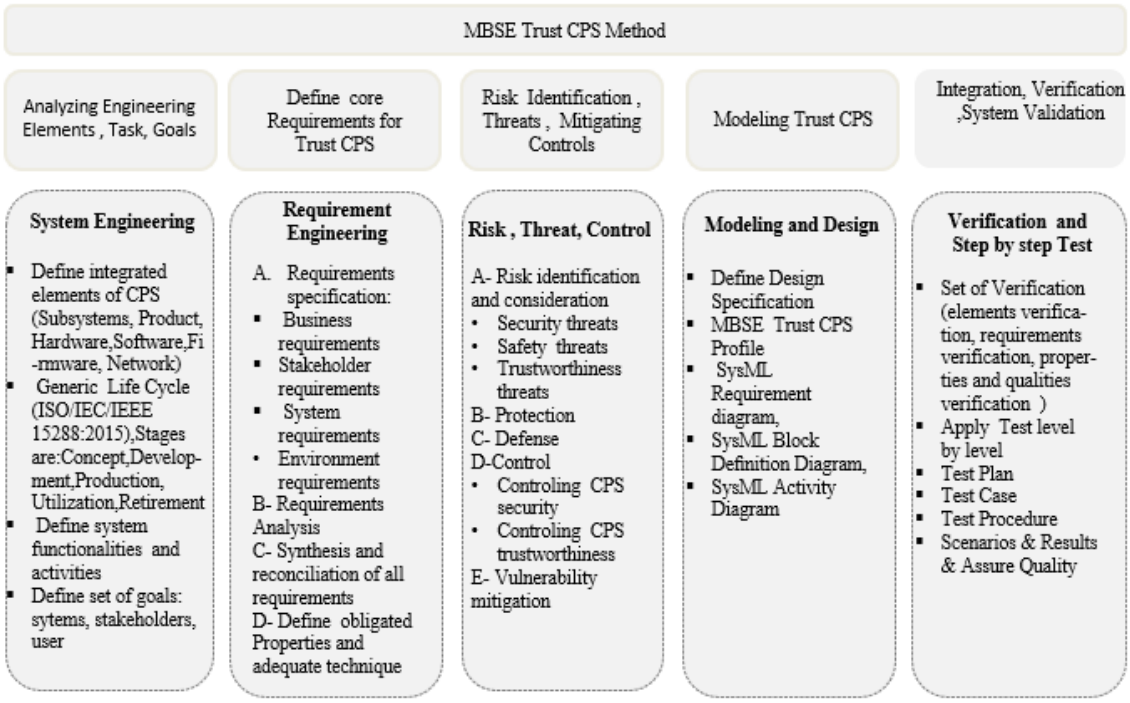

Fig. 2. The phases of model based system engineering (MBSE), activities in phases, and used methods.

To produce intuitive system descriptions, MBSE requires the synthesis of stakeholder needs into architecture models. A trust CPS technical requiring multiple disciplines to design, for this reason, our proposed MBSE method for trust cyber physical system is based on many disciplines such as system engineering that is the pillar of MBSE, requirement definition, modeling, verification as presented in Fig. 2 (source: compiled by authors) that details the definition of Phases, activities in phases and defining used methods.

A. Phases

1) Phase 1(using system engineering and analyzing engineering elements, tasks and goals): Top-down synthesis is an iterative approach used in systems engineering [22]. SE is a discipline that focuses on the overall (system) design and implementation, taking into account all the facets [23].

a) System definition: Define integrated elements of CPS (Subsystems, Product, Hardware , Software, Firmware, Network). The exterior and internal views of the system introduced the elements that belong to the system and interact with it. The functionality of system is expressed by the interaction of system with its operating environments. We recommend using the SysML Block Definition Diagram or UML Component Diagram.

b) The stages of the Generic Life Cycle (ISO/IEC/IEEE 15288:2015) are: Concept, Development, Production, Utilization, and Retirement [24].

c) Define system activities (we recommend the use of SysML Activity Diagram).

d) Define set of goals: systems and stakeholders, customer.

2) Phase2 (Engineering Requirement)

a) Identify trust requirements: Engineering requirements (RE) is a methodical, disciplined strategy for defining and managing needs [25]. In the design of software and cyber-physical systems, requirements are frequently the most critical step. They serve as a guideline for implementation as well as a reference for final product verification and validation.

During the requirements analysis stage, we are concentrating on trust modeling. We seek to integrate trust attributes in the early stages of trust CPS creation by combining a CPS cycle life with a set of trustworthiness requirements. Our primary goal is to create a model that may aid the CPS designer throughout the complete CPS design process, utilizing the information available in the requirements specification and the availability of necessary characteristics to test and monitor the final implementation's behavior. The standard SysML requirements diagram is recommended. They described a basic need for trust CPS in [3] as follows:

- Engineering requirements: For determining goals, needs, and available resources to aid in the development of a system that meets the requirements.
- Cyber-physical system requirements: Understanding the aim, specifications, and pre-designated purpose of CPS, as well as accumulating a huge number of system information.
- Trust (as a system quality) requirements: collect system, organization, and user trust needs.
- Business Requirement

b) *Defining mandatory attributes for trust CPS:* This activity is carried out during the requirements engineering

phase of CPS design. It uses objective and business process models to examine and refine trustworthiness requirements.

The cyber-physical system is considered a trust system if the principles of security and safety, confidentiality, integrity, availability, and other attributes are assured. As previously stated [3] secure, trustworthy, and trusted are system qualities that differ in terms of concerns, requirements, properties, and human interaction. However, there is a link and connection in that each quality can be complimentary to the other by adding characteristics and features to it (security and safety, in addition to a set of attributes, are both necessary to a sufficient degree to build a trustworthy cyber-physical system)[3]. The ability of a secured system to be trusted or trustworthy to some extent is a feature of the system in context.

Safety and security are interdependent, and these dependencies should be considered at the CPS design phase. Interdependence can be classified into four types: a) conditional dependencies: safety is a prerequisite for security, and vice versa; b) reinforcement: safety and security countermeasures can support each other; c) antagonism: they can undermine each other; and d) independence: there is no relationship between safety and security.

In [3] they presented an attribute classification based on system functionality and obligation, as well as a set of judgment criteria. As shown in Table I, this classification of CPS characteristics and quality of service covers the most significant indicators for CPS security and trustworthiness. The terminology glossary [26] defines the majority of the attributes. Some necessary attributes are dependent on the precision and service quality of stakeholders or consumers.

3) Phase 3 (Identification of risk and threats and mitigating controls): CPS should be constructed with controls for threats that may compromise its functions. During this step, the set of assets, threats, or controls should be determined. Risk management is based on norms in industries, and some companies have their own set of procedures and standards.

a) Risk identification: NIST defined a threats as a potential risk to a computer system that could lead to the interruption of the system or the interception, manipulation, obstruction, or destruction of computational resources.

- Security threats: CPS security deals with attacks and mostly attacks threats the security attributes. A denial of service (DoS) attack, for example, renders a control system's components unreachable in some or all circumstances. This DoS can infect devices and prevent them from transferring data, as well as attack routing protocols and communication channels [34]. The common attacks on cyber-physical systems are attacks on data integrity that include false data injection attacks. Integrity is compromised when an attacker deletes or edits crucial data unintentionally or maliciously, leading the receivers to believe the modified or deleted data is accurate. Integrity in CPS could be defined as the ability to achieve physical goals by avoiding, detecting, or defeating attempts to tamper with data transmitted and received by sensors, actuators, and controllers [35]. Some attacks are presented in Table II.

TABLE I. MANDATORY ATTRIBUTES FOR TRUST CPS

Attributes	Definition	Sub-attributes
Safety [27][28]	The ability to operate without causing danger or harm to people or the system's environment.	fault tolerance, robustness
Security [29][30][31]	A software and hardware system's ability to protect entities from attacks and misuse as well as to safeguard resource access is referred to as security.	availability, accountability, auditability, assurance, traceability, integrity, confidentiality, non-repudiation
Compatibility [32]	The ability of software and hardware from multiple sources to work together without having to be modified.	Openness
Performance [32][33]	Describe the effectiveness of a service. The quantity of event responses handled within an interval is referred to as throughput. Response Time: The amount of time it takes for the service to complete a single transaction.	throughput, Response Time
Dependability [31][32]	That ability can be justified and placed on the service it provides. The anticipated execution was correct and predictable.	accuracy, availability, robustness, reliability, scalability, maintainability
Privacy [27][33]	The system's capability and functionality that allows users to govern the use of their personal information or data,	No one else can access or utilize their personal information.
Usability [ISO 9241-210][33]	Is a set of attributes that can be meet in same time for unique product or system. It refers to the ease with which a user can learn to operate, prepare input for, and interpret the output of the service. Positive attitudes towards the use of the service	satisfaction, learnability, effectiveness, efficiency of use
Correctness [32]	Correctness refers to whether a system behaves in a way that satisfies user needs, especially those related to trust expectations and trustworthiness requirements.	Examine whether a system's behavior complies with the requirements of the user.

TABLE II. SOME ATTACKS

Attack	Paper
Denial of service (DoS)	[36]
False data injection	[37]
Covert attack	[38]
Generic deception	[39]
Eavesdropping	[40]
Packet scheduling attack	[41]
Load redistribution attack	[37]

- Safety threats: Concerns about safety arise from interactions between the environment and the CPS, within the CPS, and between the CPS and authorized users. CPS safety is related to the CPS's ability to assure the absence of catastrophic repercussions on the lives, health, property, or data of stakeholders and the physical environment. Safety is an essential concern that affects process industries, and IEC 61508 is regarded as a basic safety standard that applies to all industries. In [42], they discussed how to improve safety awareness and demonstrated that individuals' safety, actions, and ability to deal with dangers at work are heavily influenced by their consciousness and behaviors.
- Trustworthy threats: Some threats are connected to privacy, which is the ability of the CPS to prohibit entities (people, machines) from having access to data stored in, created by, or transiting a CPS or its components. Privacy is the system's ability and functionality that allows users to regulate the use of their private information or personal data. It is a significant contributor to the system's trustworthiness. Privacy attacks are a type of passive assault that target data collecting and can be used to leak sensitive information and reduce the visibility and control of the user over his private information. Mostly attacks threat the trustworthy attributes. The denial of service (DoS) threats availability that is the readiness for usage, and the reliability that is the continuity of service [43]. The attacks result service delay when companies delay providing services on time due to the problems in the system. Unauthorized users or hackers may gain access to specific data and extract confidential information [44].
- Trust concerns:
  - Usability concerns relating to CPS's capacity to be used successfully to meet functional objectives and user satisfaction (adapted from ISO 9241-210) Meeting usability requirements becomes more difficult when physical and cyber components are combined in complicated systems.
  - Correctness involves system behavior in compliance with user needs, including user trust expectations and trustworthiness criteria.

b) Defense: CPS security corresponds to defense against attacks. The defense strategies is based on protection and detection , and mitigation. The prvention led to reducing attacks [45]. Several study in literature presented protection-based approaches and discussing many approches against CPS attacks such as: Control [46], protection-based[47], security metrics[48], state estimation[49].

- Protection
  - Functional Protection: Functional security safeguards the software system against harmful, infiltrating code from both the outside and inside the company.
  - Information protection: Information security safeguards the confidentiality, integrity, and availability (CIA) of computer system data and functionality against unauthorized and harmful access.
  - Safety Protection: Protection against faults, errors, and failures, damage to life, health or society, or injury to the environment. Fault-tolerance, availability, and fail-safe states are examples of safety quality sub-attributes.
- Control: This part consist of deciding trust objectives and controling security and trustworthiness of CPS.
  - Controlling CPS security: In [50], authors categorized three security defense mechanisms and stated that: a) prevention is used to delay the onset of an attack; b) resilience to close the operation of the attack; c) detection to identify the source of the attack; and d) isolation of corrupted subsystems and speedy restoration of normal mode. The defense plan should rely on three mechanisms to avoid: a) the period between the commencement of the attack and discovery, which results in system damage. b) inability to protect against spoofing attacks. An example of a failure to detect [51]. Some control strategies in the literature, such as (observer-based techniques, watermarking, baiting, and learning-based anomaly detection), were categorised in [50]. The authors of [52] proposed an authentication strategy to secure the integrity of devices and utility servers and to avoid tampering attempts using cryptography techniques.
  - Controlling CPS trustworthiness: The major tasks for measuring system quality and trust attributes are the identification of threats that may occur and affect user trust, as well as corresponding controls that may be undertaken to minimize the threats. The use of the risk-based method to identify threats to trustworthiness on an abstract level and computational approaches to evaluate end-to-end system trustworthiness in terms of several trustworthiness metrics as an example of trustworthy evaluation in design time [53].
- Vulnerability mitigation: The following are the most commonly used techniques and controls [54]:
  - Tamper resistance controls on field devices
  - Trusted procurement procedures
  - Patching and updating
  - Encryption
  - Penetration testing and internal audit
  - Network segmentation
  - Use of different technologies

4) Phase 4 (Modeling and Design): In literature, there is a variety of modeling methods and languages for CPS and its

quality. The integration of models that capture software and computational behavior with the physical environments is a challenged task a long with the inability to integrate discrete-event and continuous-time modeling paradigms for improving the ability to provide trustworthy CPSs in the future [55,56].

Architectural modeling tools for CPS are frequently used to depict full systems, including graphical notations such as SysML and UML that are useful for considering how the CPS is organized, as well as how the constituent elements interact and share data. Continuous-time modeling paradigms and discrete-event modeling paradigms are the two most well-known modeling methodologies. Modeling techniques that rely on mathematical notations are capable for representing continuous-time behaviors. Continuous-time modeling is required for the creation of a physics model capable of precisely predicting a system's interactions with its physical environment. It captures dynamic behavior of a system by utilizing iterative methods of integration and differential equations [57]. When it comes to physical processes and analog circuits, it used continuous-time techniques [58].

Some systems modeling methodologies and tools for CPS design and analysis include hybrid discrete-event and continuous simulations [59], inductive constraint logic programming [60], hybrid timed automaton [61], ontologies [62], information schema [63], UML [64], and SysML [65].and information dynamics modeling [66], meta-model for multimedia software architecture (MMSA) that enables the description of software architectures [67], SMSA (Security Meta-model for Software Architecture)[68], trustworthy collaboration [69].

As languages that have been designed for modeling holistic embedded systems and CPSs : Stateow/Simulink, Modelica [70], hybrid CSP [71], and HyVisual [72]. Comodeling (collaborative modeling) is an approach that focuses on creating system models composed of separate models [73]. An approach cosimulation engine called Crescendo[57].

In previous phases, we explained that a trust CPS technical requiring multiple disciplines to design which have be combined. For trust CPS modeling, we attempted to capture customer objectives and requirements about trust and reflect them in system functioning. Our suggested activities and tools for MBSE of trust CPS are presented in Table III.

In Fig. 3, our profile diagram is presented. The SysML profile is used on package to incorporate stereotypes. It depicts the MBSE method's phases and underlying trust methods. The MBSE suggests several tasks, including (SysML Block Definition Diagram, SysML Requirement Diagram, SysML Activity Diagram, SysML Use Case Diagram). UML's diagrams can be used also for the suggested activities [74].

5) Phase 5 (Verification and Step by step test): In this phase a set of verification (elements verification, requirements verification, attributes and qualities verification) is done and operated in parallel with pervious phases.

A test step by step is applied to ensure the quality of modeling and level by level verification to ensure the design of entire system as mentioned in Fig. 4. The major objective is to conduct more analysis to see if the system satisfies trust qualities and allows for quick feedback on requirements and design choices.

System requirements are verified against the stakeholder and customer requirements and in the line with set of trust requirements. This step may results change or delete of requirements. The second level of verification targets system realization. The third level targets the use of system. A combination of test plan with test case and scenarios can be applied.

TABLE III. SOME ACTIVITIES IN MBSE

Activities	Implementation in SysML	Purpose
Trust Requirements Definition	SysML Requirement diagram,	Identifies both functional and non-functional trust criteria
Trust Structure Definition	SysML Block Definition Diagram,	Defines system components as well as their contents (attributes, Behaviors, Constraints), interfaces, and relationships.
Security Constraints Definition	SysML Block Definition Diagram	Captures policies pertaining to trust.
Trust Processes Definition	SysML Activity Diagram	Determines trust controls

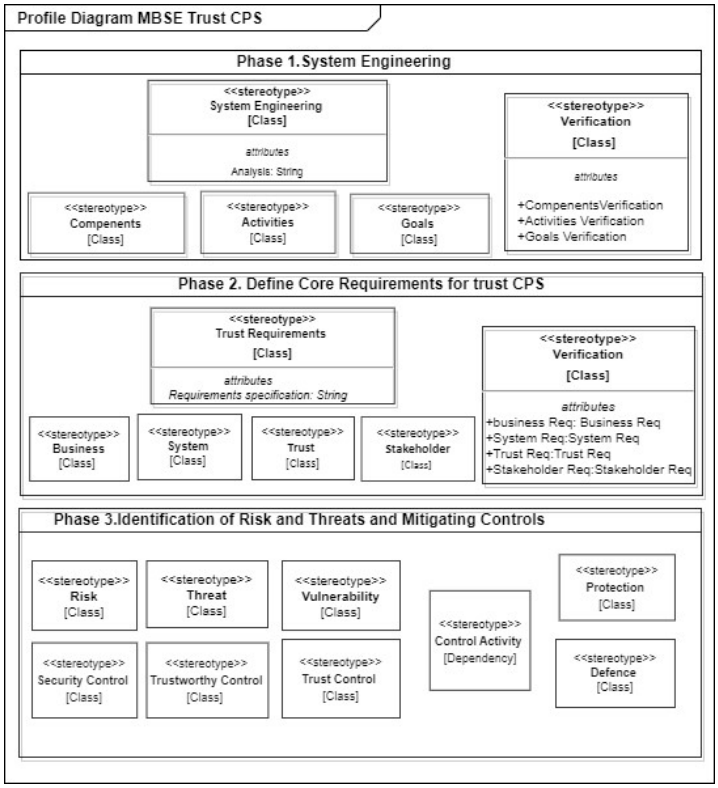

Fig. 3. The MBSE trust profile.

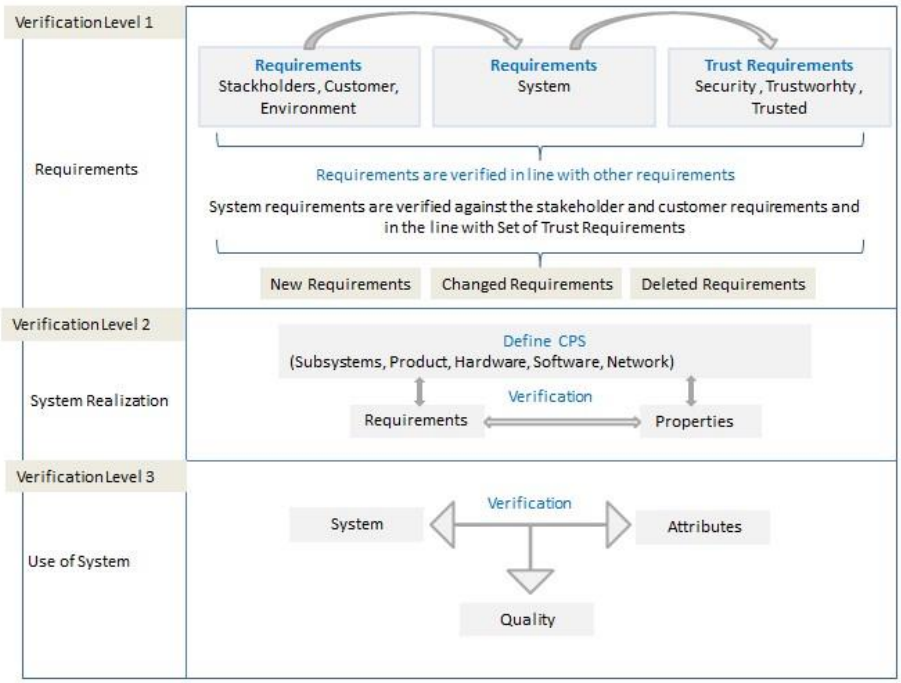

Fig. 4. Verification Phase, Test Step by Step (Elements verification, requirements verification, attributes and qualities verification).

B. *Case Study: Can we trust autonomous vehicles (AVs) on the road?*

One of the largest system engineering challenges to date is the development of autonomous vehicles (AVs). For demonstrating the proposed MBSE method usage, we applied the autonomous vehicle (AV) that enables the preparation for MBSE and presents an initial guideline for establishing the trustworthiness of AVs.

1) Trust AVs: Trust is the first foundation for acceptance of AV and plays a critical part in fostering the relationship between the user and automation and increasing people's desire to use or interact with it [75]. Different surveys showed that people were fascinated by AV but hesitated to trust it. In [76], They discovered that 65% of their participants were concerned about the dependability of self-driving cars. According to [77], 43% of participants are terrified about driving in an autonomous vehicle. A survey [78] found that 22% of respondents couldn't envisage riding in a fully automated car. The authors acknowledged in the quoted polls that the majority of participants had not yet experienced any automated driving functions, making a true judgment difficult.

AV has six levels of autonomy, with the human driver monitoring the driving environment from L0 to L2, and the automated driving system monitoring the driving environment from L3 to L5 [79]. Trust is a key aspect in the evaluation of autonomous systems and influences user behavior [80, 81], and a supportive user interface is vital, particularly during the transition period to automated driving, when the "driver" must relinquish control in favor of an unfamiliar feature.

2) MBSE for trust AVs: We are applying MBSE method to two key aspects of developing trust systems: (1) ensuring trust through the use of requirements set that related to the interactor's needs, and (2) infusing trust attributes into systems engineering practice.

The UML Component Diagram for AV is presented in Fig. 5.

a) Component diagram for AV: The first phase of system engineering and analyzing engineering elements suggests system definition and integrated elements of CPS. The important components of AV are presented in Table IV.

TABLE IV. AUTONOMIOUS VEHICLE'S COMPONENTS

Main Components	Electronic Control Unit	Sensors/ Actuators System
Central Gateway	Electronic engine control	Brake System
Vehicle control unit	Airbag Control Unit	Electric Power Pack
Driver assistance system domain controller (DASY]	ESP Unit	Vehicle Motion and position sensors
Information domain computer	Electronic Immobilizer	Ultrasonic sensor
V2X Connectivity control unit	Steering Control Unit	Near range camera sensor
Body Computer Module		Mid-Range Radar sensor
		Multi-Purpose Camera

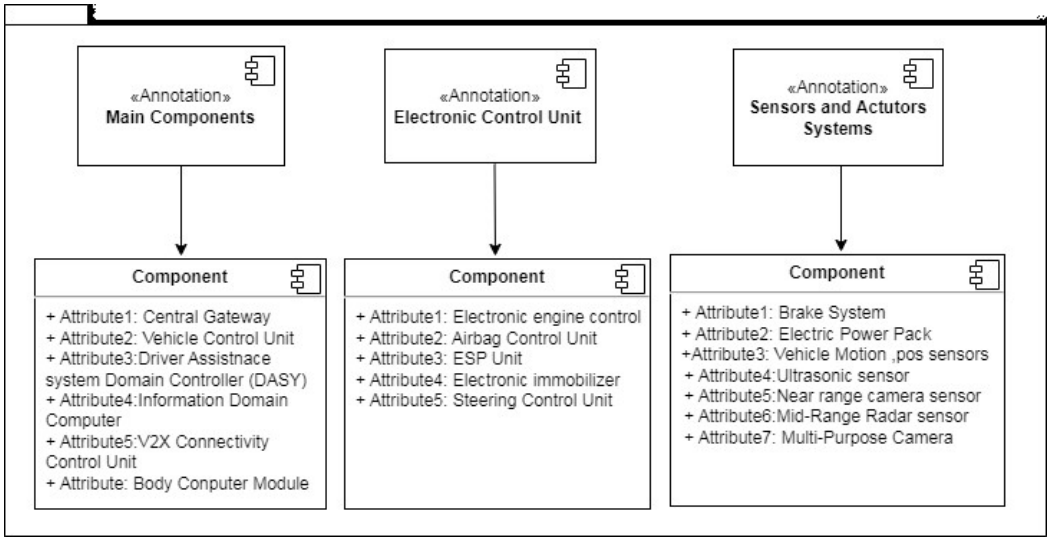

Fig. 5. UML component diagram for AV.

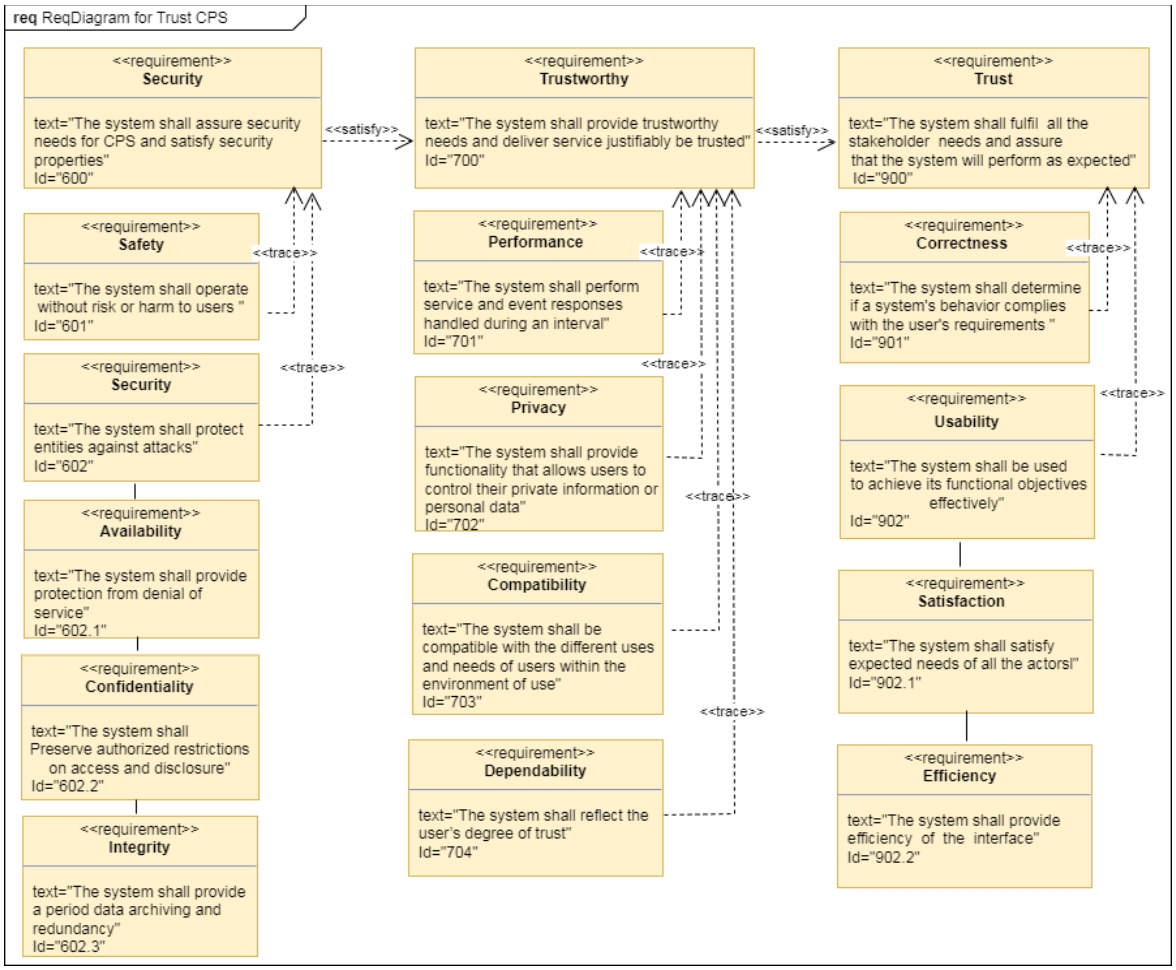

Fig. 6. SysMlL requirement diagram (General set of trust requirements for the AVs).

b) SysML requirement diagram for trust AVs: In the second phase of "trust requirements", we identify the general set of trust requirements for the AVs as presented in Fig. 6.

This step consists of determining the set of trust requirements (secured, trustworthy, trust) and, in the case of trust AV, how to integrate trust consideration into the requirement diagram. The requirements that will be merged are as follows:

- Secured requirement in which the system must ensure the AV's security and safety needs and satisfy all relevant attributes. The safety property is satisfied when the AV operates on the road without putting the system, the driver, or the passengers at risk. The AV must defend entities from attacks in order to fulfill the security property. The availability is ensured by offering protection against denial-of-service (DoS) assaults.
- Trustworthy requirement in which the system shall provide trustworthy needs and deliver service justifiably be trusted. Performance, privacy, and dependability must all be met.
- A trust need that is directly tied to human concern, and the system must fulfill all stockholder needs and ensure that the system performs as intended. All of the following criteria must be met: correctness, usability, satisfaction, and efficacy.

We used the requirement element as an element to present the requirement view, and as connectors, we used "satisfy" to present that to attend trustworthy requirements should satisfy security requirements, and to attend trust requirements should satisfy trustworthy requirements. The second connector "traces to present the relation between the requirement view and the attributes."

- SysML Requirement diagram over functionality of AVs: This diagram shows the expectation of AV driving to a given destination, from its start position, without colliding with encountered obstacles, in the shortest possible time as mentioned in Fig. 7.

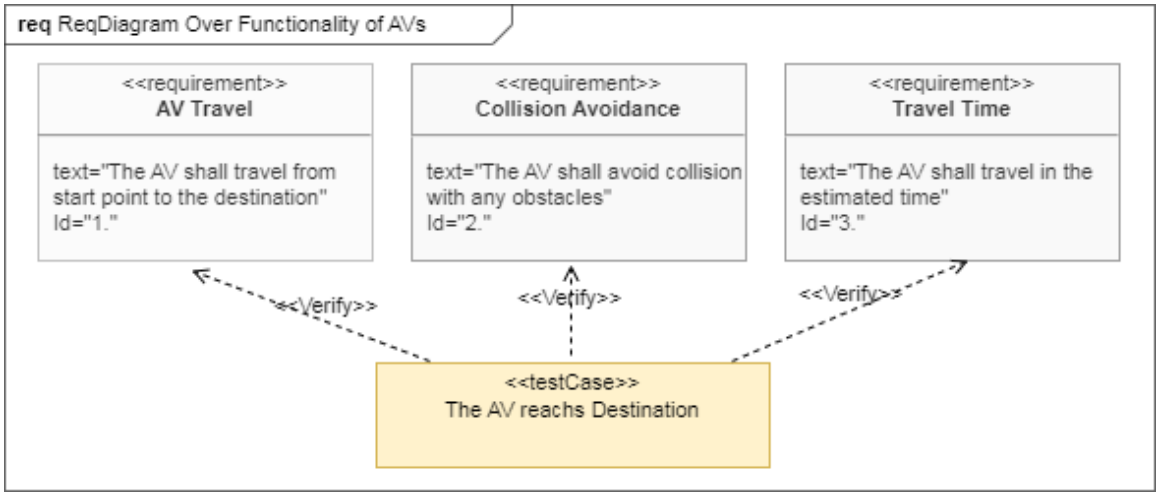

Fig. 7. SysML requirement diagram over functionality of AVs.

c) The risk definition diagram for AVs (driving style, environment, visualisation): The third phase is the identification of risk and threats. In the field of the automotive industry, the lack of traceability of security threats and their effects with safety hazards is a source of risk, for that reason is necessary to determine this traceability. We prioritized and mentioned the essential threat that rely to the driveless of AVs in the road as presented in Fig. 8, the risk definition diagram.

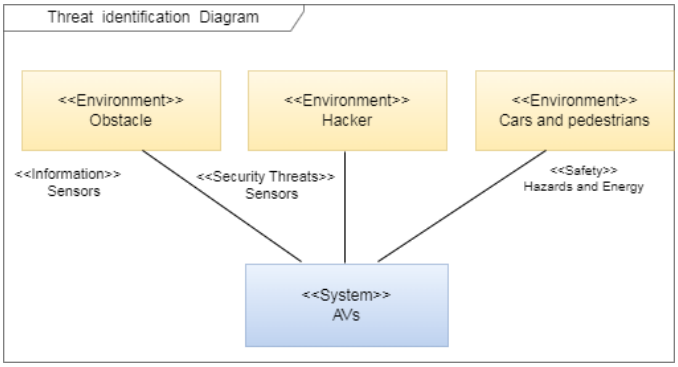

Fig. 8. Risk definition diagram for AVs.

Identifying risks related to phase 3 is used to eliminate security concerns at an early stage of system development and boost final user awareness, which leads to the discovery of suitable defense solutions. This activity incorporates security and safety concerns into a risk identification diagram.

d) SysML uses diagram reflecting how the interactors trust AV: The human must be convinced of AV's functionality, safety and reliability. In the case of AV, human can be the designer or the user, or the passenger, pedestrians and drivers on the roads.

Trust and confidence must be won at the component, systems, vehicle, and V2X communication levels, AV's driverless in the road. We have to understand how people trust AV and how interact with it. For our case, we will focus on trust of AV's driverless in the road. Drivers can be either autonomous vehicles or human passengers in autonomous vehicles.

People who participate in the traffic environment on foot, via bicycle, or by other manual transportation methods are referred to as pedestrians. The SysML use case diagram is presented in Fig. 9.

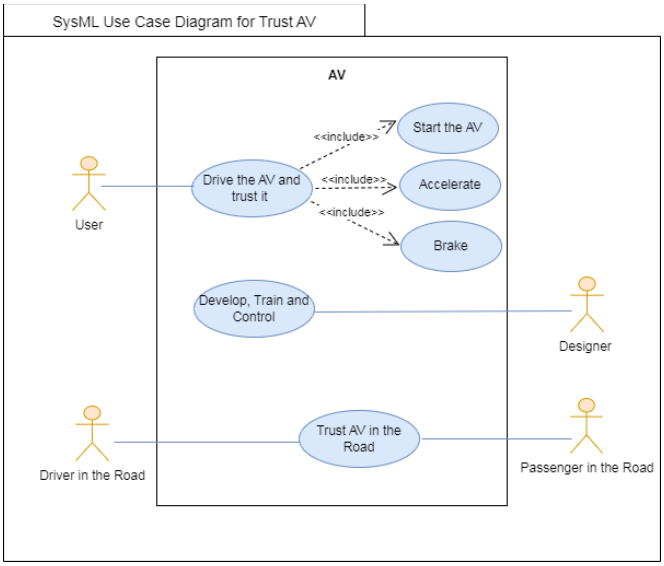

Fig. 9. SysML use case diagram for trust AVs.

The usability and satisfaction properties are included into the trust concern in the use case diagram. How users drive and trust AVs, how designers build, train, and control AVs, how drivers and passengers trust AVs on the road.

V. Discussion

MBSE was utilized to manage the complexity of developing trust in CPS in terms of system requirements, design, analysis, verification, and validation activity. Many studies have shown that security requirements and risks can be addressed in the MBSE model, which is required to address the complicated multidisciplinary, multi-domain process of CPS, and that modeling may be successfully reused in the multidisciplinary sector.

MBSE has become a key component of developing complex cyber-physical systems [17, 18], and it is widely utilized [82] for collecting and controlling the system's requirements [83], architecture, design [84], and environment identification [85]. Also, by participating and expressing views for various objectives, it can help to ease communication among many stakeholders.

The MBSE method requires the design of different partial models for the different aspects of a technical system. For the use case, we applied our proposed phases to AV usage on the road. In this model, the system structure, the components of the system are modeled in the UML component diagram, and the interactions between users and the system are modeled as use case diagrams. The important risks are identified in the threat definition diagram. Trustworthiness is a holistic property of CPS, and heterogeneous system is modeled as a requirement diagram. The expectation of AV driving is modeled as a requirement diagram over functionality.

Some points are revealed:

- Targeting the balance between trust and trustworthiness is an important task during development.
- Cyber-physical systems should be made capable of presenting trustworthiness attributes.
- A well-structured method is required for modeling and developing a trustworthy CPS.
- MBSE could be leveraged in order to mitigate security risks and assure trustworthiness requirements at an early stage of system development.

VI. Conclusion

In this paper, an analysis of specific methodologies and tools that cover design activities for developing MBSE for trust CPS are presented. We introduced the phases of MBSE as well as activities in phases, and used methods. The MBSE method consists of the SysML/UML-based profile, trust requirements definition, and risk and threats definition, modeling the interactions of users and recommendations. The proposed MBSE method usage is presented by AV and how it can be trusted in the road and how the different actors interact with and trust it.

The use of MBSE is recommended, and companies must adopt their methodologies. MBSE is a formalized application of modeling to support system requirements, analysis, design, and validation and verification (V&V) activities, beginning in the conceptual design phase and continuing throughout development and later lifecycle phases. The MBSE approach is necessary to address the complex multidisciplinary, multi-domain process of CPS.

Cyber-physical systems (CPS) have been employed in a number of operations in the oil sector, where petroleum CPS optimization approaches can aid in petroleum exploration, production, and management. Several hazards confront the energy business, with the potential to interrupt critical supply lines, hurt the environment, and trigger a financial catastrophe. The scientific community is focusing on how to confidently realize a trust CPS. There is no clear description of all types of trust concerns and requirements in the literature, particularly in the sphere of oil and gas, and the subject of cyber security for oil and gas assets is not frequently addressed.

Our future work will focus on how to align the proposed MBSE method with the security and safety standards of specific industries, such as oil and gas and how to enhance the modeling of trustworthiness by adding a survey of the judgment by real participants, and computing and analyzing their acceptance.

References

[1] E. A. Lee, “The Past, Present and Future of Cyber-Physical Systems: A Focus on Models,” Sensors, vol. 15, no. 3, Art. no. 3, Mar. 2015, doi: 10.3390/s150304837.

[2] O. E. Williamson, ‘Calculativeness, trust, and economic organization”’, Journal of Law and Economics, pp. 453–486, 1993.

[3] Z. Oudina, M. Derdour, R. Boudour, A. Dib, and M. A. Yakoubi, ‘Trust cyber physical systems: Trust degree framework and evaluation’, Int. J. Saf. Secur. Eng., vol. 13, no. 2, pp. 213–225, Apr. 2023.

[4] S. L. Presti, ‘Holistic Trust Design of E-Services”’, in Trust in E-Services: Technologies, Practices and Challenges, 2006, pp. 113–139.

[5] J. Viega, T. Kohno, and B. Potter, ‘Trust (and Mistrust) in Secure Applica- tions”’, Commun. ACM, vol. 44, pp. 31–36, 2001.

[6] J. Gorski, 'Trust case: Justifying trust in an IT solution"', Reliability Engineering & System Safety, vol. 89, no. 1, pp. 33–47, 2005.

[7] S. Charney, Trustworthy Computing Next". Microsoft, 2012.

[8] J. Jrjens, Secure Systems Development with UML. Berlin, Heidelberg: Springer-Verlag, 2010.

[9] M. D. S. Soares and J. Vrancken, 'Model-Driven User Requirements Specification using SysML', J. Softw., vol. 3, no. 6, Jun. 2008.

[10] J. Murray, 'Model-based systems engineering (MBSE) media study', International Council on System Engineering, 2012.

[11] D. Mažeika and R. Butleris, 'Integrating security requirements engineering into MBSE: Profile and guidelines', Secur. Commun. Netw., vol. 2020, pp. 1–12, Mar. 2020.

[12] T. Amorim, A. Vogelsang, F. Pudlitz, P. Gersing, and J. Philipps, 'Strategies and best practices for model-based systems engineering adoption in embedded systems industry', in 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 2019.

[13] J. Hallqvist and J. Larsson, 'Introducing MBSE by using systems engineering principles', INCOSE Int. Symp., vol. 26, no. 1, pp. 512–525, Jul. 2016.

[14] D. Kaslow et al., 'Developing a CubeSat model- based system engineering (MBSE) reference model-interim status', in 2015 IEEE Aerospace Conference, IEEE, 2015, pp. 1–16.

[15] M. Broy, W. Damm, S. Henkler, K. Pohl, A. Vogelsang, and T. Weyer, 'Introduction to the SPES modeling framework', in Model-Based Engineering of Embedded Systems, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 31–49.

[16] T. Huldt and I. Stenius, "State-of-practice survey of model-based systems engineering," Systems Engineering, vol. 22, no. 2, pp. 134–145, 2019, doi: 10.1002/sys.21466.

[17] A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities," Systems Engineering, vol. 21, no. 3, pp. 172–190, 2018, doi: 10.1002/sys.21438.

[18] A. Morkevicius, A. Aleksandraviciene, D. Mazeika, L. Bisikirskiene, and Z. Strolia, 'MBSE grid: A simplified SysML-based approach for modeling complex systems', INCOSE Int. Symp., vol. 27, no. 1, pp. 136–150, Jul. 2017.

[19] 'International Council on Systems Engineering. Systems Engineering Handbook; Version 3.1; International Council on Systems Engineering', 2007.

[20] A. Madni and S. Purohit, 'Economic analysis of model-based systems engineering', Systems, vol. 7, no. 1, p. 12, Feb. 2019.

[21] OMG Systems Modeling Language (OMG SysML) Version 1.3, vol. 1. Needham, MA, USA, 2012.

[22] H. Eisner, Essentials of project and systems engineering management, 3rd ed. Nashville, TN: John Wiley & Sons, 2011.

[23] Systems Engineering Manual, Version 3.1. Federal Aviation Administration. 2006.

[24] Systems and Software Engineering-System Life Cycle Processes. Geneva, Switzerland: International Organization for Standardization, 2015.

[25] Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant. Rocky Nook, Inc, 2016.

[26] M. Glinz, Standard Glossary of the Certified Professional for Requirements Engineering (CPRE) Studies and Exam, Version, 1. 2011.

[27] L. Pietre-Cambacedes and M. Bouissou, 'Modeling safety and security interdependencies with BDMP (Boolean logic Driven Markov Processes)', in 2010 IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 2010.

[28] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, 'Basic concepts and taxonomy of dependable and secure computing', IEEE Trans. Dependable Secure Comput., vol. 1, no. 1, pp. 11–33, Jan. 2004.

[29] R. F. Babiceanu and R. Seker, 'Trustworthiness requirements for manufacturing cyber-physical systems', Procedia Manuf., vol. 11, pp. 973–981, 2017.

[30] A. Jaquith, Security metrics: replacing fear, uncertainty, and doubt. Pearson Education, 2007.

[31] O. E. Williamson, 'Calculativeness, Trust, and Economic Organization', J. Law Econ., vol. 36, no. 1, Part 2, pp. 453–486, Apr. 1993.

[32] N. G. Mohammadi et al., 'An Analysis of Software Quality Attributes and Their Contribution to Trustworthiness', in CLOSER, 2013, pp. 542–552.

[33] H. Mei, G. Huang, and T. Xie, "Internetware: A Software Paradigm for Internet Computing," Computer, vol. 45, no. 6, pp. 26–31, Jun. 2012, doi: 10.1109/MC.2012.189.

[34] S. Amin, A. A. Cárdenas, and S. S. Sastry, 'Safe and secure networked control systems under denial-of-service attacks', in Hybrid Systems: Computation and Control, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 31–45.

[35] S. Bi and Y. J. Zhang, 'False-data injection attack to control real-time price in electricity market', in 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, 2013.

[36] S. Soltan, M. Yannakakis, and G. Zussman, 'Joint cyber and physical attacks on power grids: Graph theoretical approaches for information recovery', in Inter- national Conference on Measurement and Modeling of Computer Systems, ACM, 2015, pp. 361–374.

[37] R. S. Smith, 'Covert misappropriation of networked control systems: Presenting a feedback structure', IEEE Control Syst, vol. 35, no. 1, pp. 82–92, 2015.

[38] S. M. Djouadi, A. M. Melin, E. M. Ferragut, J. A. Laska, and J. Dong, 'Finite energy and bounded attacks on control system sensor signals', in 2014 American Control Conference, Portland, OR, USA, 2014.

[39] K. Kogiso and T. Fujita, 'Cyber-security enhancement of networked control sys- tems using homomorphic encryption', in 54th IEEE Conference on Decision and Control (CDC). IEEE, 2015, pp. 6836–6843.

[40] Y. Shoukry, P. Nuzzo, N. Bezzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, 'Secure state reconstruction in differentially flat systems under sensor attacks using satisfiability modulo theory solving', in 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, 2015.

[41] J. Madden, B. Mcmillin, and A. Sinha, 'Environmental Obfuscation of a Cyber Physical System - Vehicle Example"', in Workshop on 34th Annual IEEE Computer Software and Applications Conference, 2010.

[42] A. A. Marzooq and A. Hatim, 'The Impact of Safety Priorities on the Economic Management of Projects: A Review".2023', 2023. International Journal of Safety and Security Engineering, vol. 13, no. 1, pp. 21–29, 2023.

[43] K. Pelechrinis and M. Iliofotou, Denial of Service Attacks in Wireless Networks: The case of Jammers". 2006.

[44] P. C. Bhaskar and R. K. Kamat, 'Assessing the Guilt Probability in Intentional Data Leakage', Int. J. Comput. Sci. Inf. Technol, vol. 3, no. 3, 2012.

[45] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, 'Secure control systems: a quantitative risk management approach', IEEE Control Syst, vol. 35, no. 1, pp. 24–45, 2015.

[46] Y. Yuan and Y. Mo, 'Security in cyber-physical systems: Controller design against known-plain text attack', in 54th IEEE Conference on Decision and Control (CDC), IEEE, 2015, pp. 5814–5819.

[47] M. Garcia, A. Giani, and R. Baldick, 'Smart grid data integrity attacks: Observable islands', in 2015 IEEE Power & Energy Society General Meeting, Denver, CO, USA, 2015.

[48] J. M. Hendrickx, K. H. Johansson, R. M. Jungers, H. Sandberg, and K. C. Sou, "Efficient Computations of a Security Index for False Data Attacks in Power Networks," IEEE Transactions on Automatic Control, vol. 59, no. 12, pp. 3194–3208, Dec. 2014, doi: 10.1109/TAC.2014.2351625.

[49] M. Pajic, P. Tabuada, I. Lee, and G. J. Pappas, 'Attack-resilient state estimation in the presence of noise', in 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, 2015.

[50] M. Seyed Mehran Dibaji, 'A systems and control perspective of CPS security".2019', Annual Reviews in Control, vol. 47, pp. 394–411, 2019.

[51] J. Slay and M. Miller, 'Lessons learned from the maroochy water breach', in IFIP International Federation for Information Processing, Boston, MA: Springer US, 2007, pp. 73–82.

[52] N. Chadalavada Naga Priyanka, 'Analysis on Secured Cryptography Models with Robust Authentication and Routing Models in Smart Grid".2023', International Journal of Safety and Security Engineering, vol. 13, no. 1, pp. 69–79, 2023.

[53] T. Nazila Gol Mohammadi and A. Bandyszak, Combining Risk-Management and Computational Approaches for Trustworthiness Evaluation of Socio-Technical Systems". .

[54] G. Stergiopoulos, D. A. Gritzalis, and E. Limnaios, 'Cyber-attacks on the oil & gas sector: A survey on incident assessment and attack patterns', IEEE Access, vol. 8, pp. 128440–128475, 2020.

[55] E. A. Lee, CyberPhysicalSystems:DesignChallenges. 2008.

[56] M. Broy, M. V. Cengarle, and E. Geisberger, 'Cyber-Physical Systems: Imminent Challenges', in Large-Scale Complex IT Systems. Development, Operation and Management, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1–28.

[57] J. Fitzgerald, P. G. Larsen, and M. Verhoef, Collaborative Design for Embedded Systems. Berlin Heidelberg: Springer- Verlag, 2014.

[58] J. Liu, Continuous Time and Mixed-Signal Simulation in Ptolemy II.TechnicalReportNo. UCB/ERL M98/74. 1998.

[59] E. A. Lee, M. Niknami, T. S. Nouidui, and M. Wetter, 'Modeling and simulating cyber-physical systems using CyPhySim', in 2015 International Conference on Embedded Software (EMSOFT), Amsterdam, Netherlands, 2015.

[60] N. Saeedloei and G. Gupta, 'A logic-based modeling and verification of CPS', ACM SIGBED Rev., vol. 8, no. 2, pp. 31–34, Jun. 2011.

[61] M. Burmester, E. Magkos, and V. Chrissikopoulos, 'Modeling security in cyber–physical systems', Int. J. Crit. Infrastruct. Prot., vol. 5, no. 3–4, pp. 118–126, Dec. 2012.

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

[63] S. Pourtalebi and I. Horváth, 'Information schema constructs for instantiation and composition of system manifestation features', Front. Inf. Technol. Electron. Eng., vol. 18, no. 9, pp. 1396–1415, Sep. 2017.

[64] G. Magureanu, M. Gavrilescu, D. Pescaru, and A. Doboli, 'Towards UML modeling of cyber-physical systems: A case study for gas distribution', in IEEE 8th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 2010.

[65] E. Palachi, C. Cohen, and S. Takashi, 'Simulation of cyber physical models using SysML and numerical solvers', in 2013 IEEE International Systems Conference (SysCon), Orlando, FL, 2013.

[66] Y. Wang, 'Probabilistic modeling of information dynamics in networked cyber–physical–social systems', IEEE Internet Things J., vol. 8, no. 19, pp. 14934–14947, Oct. 2021.

[67] M. Derdour, G. Zine, P. Roose, M. Dalmau, and A. Alti, 'UML-profile for multimedia software architectures', Int. J. Multimed. Intell. Secur., vol. 1, no. 3, p. 209, 2010.

[68] M. Derdour, A. Alti, M. Gasmi, and P. Roose, 'Security architecture metamodel for Model Driven security', J. Innov. Digit. Ecosyst., vol. 2, no. 1–2, pp. 55–70, Dec. 2015.

[69] Y. Wang, 'Trustworthiness in Designing Cyber-Physical Systems', in Proceedings of the 12th International Symposium on Tools and Methods of Competitive Engineering (TMCE2018), Las Palmas, Gran Canaria, Spain, 2011, pp. 27–40.

[70] V. Fritzson, 'Modelica-A unied object-oriented language for system mod- elling and simulation', in ECCOP '98: Proceedings of the 12th European Conference on Object- Oriented Programming, Springer-Verlag, 1998, pp. 67–90.

[71] H. Jifeng, From csp to hybrid systems, A Classical Mind: Essays in Honour of CAR Hoare. 1994.

[72] E. A. Lee and H. Zheng, 'Operational Semantics of Hybrid Systems', in Hybrid Systems: Computation and Control, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 25–53.

[73] J. F. M- Y Ni, 'A co-modelling method for solving incompatibilities during co-design of mechatronic devices', Advanced Engineering Informatics, vol. 28, no. 3, pp. 232–240, 2014.

[74] J. Gabay, Merise et UML: Pour la modélisation des systèmes d'information. 2004.

[75] M. Ghazizadeh, J. D. Lee, and L. N. Boyle, 'Extending the technology acceptance model to assess automation', Cogn Tech Work, vol. 14, no. 1, pp. 39–49, 2012.

[76] "Public opinion on automated driving: Results of an international questionnaire among 5000 respondents - ScienceDirect." https://www.sciencedirect.com/science/article/abs/pii/S136984781500077 (accessed Jun. 23, 2023).

[77] "YouGov." https://yougov.de/topics/_/articles-reports/2016/07/19/unfalle-mit-%20selbstfahrenden-autos-%20fur-viele-%20ist-de (accessed Jun. 23, 2023).

[78] "A survey of public opinion about autonomous and self-driving vehicles in the U.S., the U.K., and Australia." https://deepblue.lib.umich.edu/handle/2027.42/108384 (accessed Jun. 23, 2023).

[79] "SAE J1100 - (R) Motor Vehicle Dimensions | GlobalSpec." https://standards.globalspec.com/std/1205131/sae-j1100 (accessed Jun. 23, 2023).

[80] "Measuring Trust of Autonomous Vehicles: A Development and Validation Study | SpringerLink." https://link.springer.com/chapter/10.1007/978-3-319-21383-5_102 (accessed Jun. 23, 2023).

[81] J. D. Lee and K. A. See, "Trust in Automation: Designing for Appropriate Reliance," Human Factors, 2004.

[82] Huldt, T. and I. Stenius (2018): 'State-of-practice survey of model-based systems engineering'.Systems Engineering, vol. 22, no. 2, pp. 134–145.

[83] Mazeika, D.; Morkevicius, A.; Aleksandraviciene, A. MBSE driven approach for defining problem domain. In Proceedings of the 11th System of Systems Engineering Conference (SoSE), Kongsberg, Norway, 12–16 June 2016; pp. 1–6.

[84] Roudier, Y.; Apvrille, L. SysML-Sec: A model driven approach for designing safe and secure systems. In Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD), Angers, France, 9–11 February 2015; pp. 655–664.

[85] Roudier, Y.; Apvrille, L. SysML-Sec: A model-driven environment for Developing Secure Embedded Systems. In Proceedings of the 8th Conference on the Security of Network architecture and Information Systems (SARSSI'2013), Mont de Marsan, France, 16–18 September 2013.