Rule-Based Selection of Languages for Modeling Cyber-Physical Systems

Veronica Opranescu[a] and Anca Daniela Ionita[b]
National University of Science and Technology Politehnica Bucharest, Bucharest, 060042, Romania

Keywords: Cyber-Physical Systems, Modeling Languages, Recommendation Systems, Rule-Based Systems.

Abstract: In the context of Cyber-Physical Systems (CPS), the choice of suitable modeling languages plays an important role in effectively addressing the varied interests of the system stakeholders. This paper proposes a rule-based recommendation system to suggest appropriate modeling languages that optimally cover all identified viewpoints required for a set of stakeholders. The recommendation engine employs Drools as business rule management system, to highlight the connection between stakeholders' viewpoints and the kind of models supported by available modeling languages. For assessing this method, a case study was performed with a realistic example from the domain of industrial automation and a selection from three modeling languages.

1 INTRODUCTION

The design of advanced cyber-physical systems (CPS) must cover a combination of viewpoints, as each stakeholder possesses distinctive requests and understandings. This complexity requires a meticulous approach to modeling language selection to capture all the relevant viewpoints and promote continuous collaboration at all levels of development of the same system (Cederbladh et al., 2024).

CPS are defined as systems where physical processes are integrated with computational activities for the purposes of real time monitoring, control and automatic operation. Applications for CPS can be seen in transportation, healthcare, and manufacturing industries (Hamzah et al., 2023). Effective detection, data processing and actuation are major requirements to successfully interface with physical processes in safety critical environments (Aslam et al., 2025).

Understanding how digital control contributes to physical behaviors, and how physical performance affects digital control, highlights a reciprocal influence for systems control (Stary, 2021), implying a CPS design characterized by resilience, security and efficiency, capable to easily adapt to different environments while ensuring dependable operations within multi-faceted real-world scenarios.

Modeling within CPS involves creating an abstract representation of both physical processes and computational elements (Barišić et al., 2022) . This process is executed in order to evaluate, model, and verify the behavior of the system. Modeling is an important step in understanding the complexity of interactions in CPS since models allow engineers to determine how the changes to one element of the system may propagate through the system (Daun and Tenberge, 2023). Modeling also provides stakeholders with the ability to represent and test different use cases and understand where problems may exist, as well as identify ways to optimize systems, before putting them into production (Taha et al., 2021). Overall, modeling methods enable effective design decision making that leads to safer, more reliable, and efficient cyber-physical system applications in any industry.

The study of scientific literature has proved that there is a large variety of languages used for CPS analysis and modeling (Broman et al., 2012), therefore the question about finding a set of criteria for making the right choice for modeling a specific application. We base this work on a study of three languages from the same technological space: UML (Unified Modeling Language) (Ordinez et al., 2020), SysML (Systems Modeling Language) (Parant et al.,

[a] https://orcid.org/0009-0005-7886-6888
[b] https://orcid.org/0000-0002-8966-6196

Opranescu, V. and Ionita, A. D.
Rule-Based Selection of Languages for Modeling Cyber-Physical Systems.
DOI: 10.5220/0013710200004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 2: KEOD and KMIS*, pages 120-127
ISBN: 978-989-758-769-6; ISSN: 2184-3228

2023), and MARTE (Modeling and Analysis of Real-Time and Embedded Systems) (Mallet et al., 2017).

Such criteria for identifying the views covered by their various types of diagrams were presented in (Opranescu and Ionita, 2025), which represents the starting point of the current research.

To address this challenge, in the first section, the paper introduces a framework based on a rule-based recommendation system designed specifically for the CPS domain, which automates the decision-making process by providing a set of defined rules for stakeholder - viewpoint - modeling language relationships. The system operates within the CPS context, having predefined rules on how stakeholders, their viewpoints (e.g., performance, safety, security), and the modeling languages best suited to represent those viewpoints are mapped. This section also presents the mapping rules, discusses their limitations, and explains the strategies adopted to mitigate them.

In the second section, the implemented algorithm that encompasses the rule-based part is described, translating the predefined mapping rules into an operational logic that can be executed to support automated and consistent recommendations within the system.

The last section contains a presentation of a case study for analysis and modeling in the industrial automation domain. The recommendation system is tested against a real-world CPS, evaluating its practicality and usefulness. The construction the recommendation system's architecture and the implementation of the rules have undergone rigorous validation procedures based on an actual scenario discussed in the previously mentioned case study.

The use of a real-life case study enables assessing the system's performance within empirical conditions, verifying the recommendations in practice and not only in theory.

2 RULE-BASE SELECTION OF MODELING LANGUAGES

In the design and development of CPS, a clear alignment between stakeholders and their areas of concern is of prime importance (Shahin et al., 2019). Different roles like system architects or hardware engineers, would naturally focus on different dimensions or aspects of the system that are otherwise referred to as a concern or viewpoint (ISO/IEC/IEEE, 2011). A system architect focuses on the different configurations in a system, which are partially defined by the control mechanisms and data flows, whereas a hardware engineer deals with the exact timing schedule of physical events, including associated constraints.

In an effort to minimize the time it takes individuals to manually establish properly connected stakeholders (Azzouzi et al., 2022) to their proposed concerns, a rules-based approach was selected, employing automated connection to what they would typically be responsible for. The defined rules act as the knowledge base and will be clear enough to help the system look up the proper stakeholder name, and connected viewpoints for that name to process.

This study uses the structure and classifications presented in (Opranescu and Ionita, 2025), thereby assuming them as an evidenced basis for continued research and methodological refinement.

Once the stakeholders have been mapped onto their individual perspectives, consideration is made to determine the most suitable languages to address these concerns. A list of languages will be suggested, offering complete coverage, and additional languages may be suggested to allow for greater flexibility or specificity. Moreover, a summary describing the relevance of each language to specific stakeholders will be given, thereby assisting teams in putting the recommendations into context for improved practical usage.

By using this strategy we aim to cover stakeholders' needs, and to allow project teams to focus on design and implementation and avoid the gap between stakeholders' concerns and specific technical requirements.

2.1 Mapping Rules

CPS projects generally comprise a broad group of stakeholders, such as system architects, network architects, system engineers, software developers, hardware engineers, business/data analysts, and testers. Every one of these stakeholders is accountable for certain system-related issues that are communicated through various viewpoints (Rahatulain and Onori, 2018).

These include structural, behavioral, and non-functional concerns such as data flow, control flow, state transitions, scheduling, timing constraints, security, etc., amounting to a total of 16 views identified in the present framework, based on our previous results from (Opranescu and Ionita, 2025) in Table 7 from which a subset is hereby presented in Table 1.

With the intention of ensuring these concerns are properly addressed, adequate modeling languages

must be selected to cover identified viewpoints. In this approach, the modeling languages to consider are UML, SysML, and MARTE. UML is most used for general-purpose systems and software modeling. SysML, as an extension of UML, handles additional specific features, addressing systems engineering' model requirements, behavior, structure, and parametric. MARTE, on the other hand, is focused on embedded and real-time systems and is essential when dealing with scheduling and timing constraints in CPS.

Table 1: Mapping examples of stakeholders' viewpoints to modeling languages.

Viewpoint	Stakeholder			Modeling Language		
	System Architect	Data Analyst	Tester ...	UML	SysML	MARTE
Data Flow	✓	✓		✓	✓	
Internal Structure	✓			✓	✓	
Timing	✓	✓		✓		✓
Functional Req	✓	✓	✓	✓	✓	
....	...	...	...	...	...	...
Resources	✓		✓			✓

Instead of using a logic that encompasses every stakeholder and their respective viewpoints and related modeling languages, in the above method, the system is based on a rule-based system supported by Drools (Proctor, 2012). Drools is a business rules management system that allows for the domain knowledge to be expressed in a declarative manner. The proposed method introduces a measure of flexibility into the process as an entity (Huang et al., 2020).

The relationships between stakeholders, perspectives, and modeling languages are expressed by adjustable rules that are simple to change or extend. For this reason, when new stakeholders or modeling languages are added, the system needs to make minimal adjustments to its internal logic.

The Drools rules framework presents a formal technique of knowledge representation (Grimm, 2009), which encodes professional insights related to stakeholder roles and viewpoints as well as their respective modeling languages. Through its declarative nature the system manages complex relationships to generate consistent recommendations within the CPS framework.

The selection of the Drools-Java technology stack is based on the findings of a previous study (Opranescu et al., 2020), which analyzed a range of technologies relevant to ontology-based recommendation system development. That paper provided a comparative overview and practical insights for researchers and practitioners, with a particular focus on the automatic integration of rule engines into CPS. Based on that analysis, Drools was identified as a suitable choice due to its declarative rule syntax, compatibility with semantic technologies, and native integration with Java applications, particularly effective for supporting automated rule integration.

This approach proposes that Drools rules defined in .drl files should serve as the core mechanism for domain-specific knowledge encapsulation which enables stakeholder view alignment and view-to-modeling language mapping. By enabling declarative logic specification through the Drools platform users can increase readability and simplify maintenance efforts.

Every single rule contains a specific, focused piece of logic, totaling 7 rules to map each stakeholder to corresponding viewpoints and 16 rules that map each viewpoint to covered modeling languages. For instance, one guideline might stipulate that an individual in the role of a "System Architect" will generally be concerned with viewpoints such as "Data", "Control Flow", "Internal Structure", "State Flow", "Interaction", "Communication" etc. (see Figure 1).

Another rule states that if a stakeholder is interested in the "Control Flow" viewpoint, both UML and SysML are the appropriate modeling languages to fulfill that concern (see Figure 2).

```
rule "Define System Architect Stakeholder"
  when
     not Stakeholder(name == "System Architect")
  then
     insert(new Stakeholder("System Architect",
     Arrays.asList("Data", "Control Flow", "Internal Structure",
     "State Flow", "Interaction", "Communication")));
end
```

Figure 1: Rule that maps Stakeholder to Viewpoints.

```
rule "Map Control Flow Viewpoint to UML and SysML"
  when
    $stakeholder : Stakeholder(viewpoints contains "Control Flow")
  then
    HashMap<String, List<String>> explanation = new HashMap<>();
    explanation.put("Control Flow", Arrays.asList("UML", "SysML"));
    recommendations.add(new Recommendation($stakeholder.getName(),
        Arrays.asList("UML", "SysML"), explanation, new HashSet<>()));
end
```

Figure 2: Rule mapping Viewpoint to Modeling Language.

2.2 Limitations Analysis

Utilizing Drools rules for stakeholder–viewpoint–language mapping is an adaptable option characterized by high transparency, but it also has certain limitations and concerns that require further research. Despite the modularity and flexibility of the knowledge-mapping specification methods available in rule-based systems, such as Drools, they are subject to specific restrictions that can impact the scalability and adaptability of the recommendation.

One such limitation is *the expressiveness of static rules* (Kaczor et al., 2011). All rules need to be manually and consistently articulated, and all changes made to the domain model (e.g., adding a new kind of stakeholder class, an extra viewpoint, or using a different modeling language) still require human intervention in most of the of rules. The ongoing cost of maintaining such an architecture (often difficult to quantify) combined with the impediments that arise as system components differ over time, frequently outweighs its benefits. This is particularly true when operational costs are considered within the broader context of system sustainability and long-term effectiveness.

Another restriction focuses on the *conflict resolution and rule ordering* (Jung et al., 2012). If there are multiple rules that pertains to the same stakeholder or perspective, the reasoning mechanism must decide between the rules as to which one holds higher priority. Pushing execution order control to the level of Drools, with salience (rule priority) and agenda groups, can result in the rule base becoming large, hard to understand and debug.

Also, the proposed method provides a *binary classification* feature*,* indicating whether a modeling language addresses a particular viewpoint (Hille Pascal Van et al., 2012). As such, this method is insensitive to the subtle nature involved in modeling needs. . In practice, a language may provide partial support of a viewpoint, or several languages must be combined to achieve complete representation—situations difficult to articulate by a set of declarative rules.

Scalability is a particular concern as the expansion of stakeholders, perspectives and languages will lead to a nonlinear increment on the number of rules (Luo et al., 2021). This growth can have a negative impact on performance or further complicate the ruleset into which to evaluate and maintain.

These limitations suggest that although Drools rules will enable the expression of linear transformations easily, it is very likely that they would have to be complemented with more flexible or data-driven approaches (e.g., learning algorithms, ontology-based reasoning - enhances Drools by supplying inferred facts for richer reasoning, or several layers of metadata - provide a mechanism for more nuanced and adaptive recommendations), to deal with the sophisticated environments which characterize the development of cyber-physical systems.

2.3 Mitigating Rule-Based Mapping Limitations

To achieve greater scalability, the rule-based system is cautiously designed with an emphasis on modularity. Every individual rule is clearly defined to perform a specific mapping operation (e.g., mapping a stakeholder to viewpoints or mapping a viewpoint to one or more modeling languages). The proposed architecture facilitates the ease of managing, maintaining, and integrating rules, even as systems undergo requirement transformations. Adding extra stakeholders, viewpoints, or modeling languages is done through the creation of new rules, hence securing the underlying logic.

Using Java programming, advanced post-processing is addressed by employing rules through the Drools engine. After the rules are executed, and processed, and the mappings are loaded, the Java code supports the algorithm by resolving any inconsistencies, ordering modeling languages based on coverage, and presenting the recommendations. These additional steps provide a simple and compact way of handling rules, while the Java part of the algorithm manages internal logic, including ordering and sorting.

The transparency principle is a critical advantage that is linked with this design. In .drl files used in the Drools framework, stakeholders' concerns and modeling languages are well defined. These are designed to be readable and editable by domain experts, irrespective of programming knowledge. The system thus achieves traceability and flexibility, hence improving peer-review mechanisms and allowing recommendation rules to evolve incrementally.

3 RECOMMENDATION SYSTEM

The highlight of the language modeling recommendation algorithm is its integration with a general rule-based knowledge representation system by utilizing Drools. Integration makes it easy to separate decision logic from domain knowledge so

that the system is flexible enough to handle the changing modeling demands for CPS design.

3.1 Algorithm

The recommendation flow starts by accepting as input a set of stakeholders and the respective viewpoints that are of interest to each stakeholder. Upon receiving the input, the process moves through a number of steps.

One of the most important features of the recommendation framework is that it acknowledges the collective nature of all stakeholders working together on the same CPS. Instead of making individual language recommendations for each stakeholder, the algorithm calculates a shared set of modeling languages that collectively represent the opinions of all parties involved. This approach maintains communication and modeling consistency within the team and thereby guarantees the smooth integration of contributions from all disciplines. The engineering process is made more efficient and less prone to misinterpretation or redundancy by creating a common linguistic foundation.

As previously mentioned, the system uses a rule based Drools engine to find the specific viewpoints for each stakeholder. The input dataset provided by the stakeholders determine what viewpoints they are concerned with. The rule engine takes this information and attempts to map stakeholders with their appropriate viewpoints.

Then, the algorithm correlates those viewpoints to the appropriate modeling languages, like UML, SysML, and MARTE. This mapping is also performed using Drools rules which specify, for each viewpoint, its corresponding modeling language/s. If a viewpoint is not covered by any of the previously selected languages, the algorithm will add the newly identified modeling language to recommended languages list that provide full coverage for the initial viewpoints.

After mapping the viewpoints and modeling languages, the algorithm analyses and compares the language with the number of covered viewpoints. This analysis is carried out in the Java logic part of the system which sorts the languages based on the number of viewpoints they cover, starting with the most used. The selection is done first for the primary languages considering the overall coverage of viewpoints.

Lastly, the algorithm then generates a list of recommended modeling languages, necessary to capture the view of all stakeholders involved. In the situation where more than one language is needed to capture the view of a specific stakeholder, the system will explicitly state that the mentioned languages are meant to be used concurrently, giving a clear and comprehensible indication of the language(s) needed for full representation.

The integration of Drools rules within the Java application is a straightforward process, which contributes to the system's adaptability and maintainability. This approach facilitates changes implementation of mapping rules and recommendations without impact on the logic part of the algorithm. Consequently, the proposed solution supports rolling adjustments while decreasing the impact to existing features of the system.

3.2 Outputs

The results generated by the modeling language recommendation system are intended to offer users who are engaged in the design of CPS precise and helpful advice. Taking as input a collection of stakeholders and corresponding views, the system generates two significant categories of output: a general proposal of modeling languages that cover all viewpoints, and individualized recommendations specific to each stakeholder.

Basically, the system determines the specific modeling languages (e.g., UML, SysML, MARTE) needed to address the total set of viewpoints accumulated from all users as specified by the stakeholders. The languages are listed and extracted with information on the number and percentage of viewpoints that they address. This then provides users with a concise overview of the significance and applicability of each language in the present framework. In cases where one language is not enough, the system clearly announces that a multilingual approach is unavoidable, pointing to the need for interdisciplinary collaboration.

Besides the overall global guidelines, the system also generates personalized recommendations for specific stakeholders. For every one of the stakeholders identified, it lists the applicable modeling languages and indicates the viewpoints supported by each language. This assists in making sure that every team member not only knows which tools to utilize but also why specific languages are needed, based on their role and responsibilities.

This dual reporting, both individual and team wise, enables organizations to achieve modeling practice harmonization between functions, leading to consistency, better communication, and reduced integration complexity in cyber physical systems design projects.

The recommendation system generates a structured, multi-layered output aimed at ensuring that the views of all stakeholders are addressed by appropriate modeling languages. The outputs include:

a. Modeling Language Integration for Complete Coverage

The framework could provide one modeling language or define a collection of modeling languages, which collectively guarantee full coverage of all the views described across the different stakeholders. The set is presented as a conjunctive set (i.e., all the modeling languages mentioned need to be used in conjunction with each other) so that no viewpoint is left out.

b. Individual Modeling Language Coverage Summary

For each modeling language in the suggested combination, the system reports:

a. the number of viewpoints that it encompasses
b. the percentage of the entire list of viewpoints that it covers
c. the specific viewpoints covered by that language.

This enables users to see the position of each modeling language in the total coverage.

c. Stakeholder-Centered Language Assignment

To provide precision in how each stakeholder's interests are supported and what modeling tools they need to use. For each stakeholder, the system identifies:

- the modeling language(s) that must be used to respond to their respective viewpoints
- the list of the viewpoints addressed by each recommended modeling language.

Together, these outputs provide a comprehensive, traceable recommendation framework that translates modeling language choice to stakeholder needs and viewpoint requirements.

4 CASE STUDY

In order to assess and analyze the performance of the proposed modeling language recommendation system, an industrial automation case study has been chosen, specifically a Cyber-Physical Production System (CPPS) as described in (Weyer et al., 2016). This case study illustrates a mature and highly modular smart factory setting, in which several stakeholders with varying perspectives collaborate in the modeling, simulation, and operation of manufacturing systems.

Using the proposed recommendation model within the industrial automation domain, the purpose is to present its potential to effectively recommend relevant modeling languages that can fully cover the large variety of stakeholder concerns as well as system perspectives required in such environments. The real-life environment offers a good test case for testing the appropriateness and comprehensiveness of the proposed modeling methods. As presented in Table 2, a representative subset of stakeholders and their associated concerns has been identified in the studied test case, although these do not encompass all stakeholders and concerns involved.

Table 2: Case study data input for recommendation system.

Stakeholder	Selected Viewpoints
System Architect	Deployment Resources Communication Functional Req Non-Functional Req Software Assembly Structure
System Engineer	Control Flow State Flow Timing Allocation Functional Req
Software Developer	Data Flow Timing Communication Internal Structure Software Assembly Structure
Hardware Engineer	Deployment Resources Allocation Communication Timing
Data Analyst	Data Data Flow Functional Req Communication Interaction

In the domain of industrial automation, modeling and simulation processes are of utmost significance. The increasing complexity of manufacturing environments, combined with the need for shorter product life cycles, requires the application of tools that can accurately mimic behavior, effectiveness, and system flexibility.

The output of recommendation system is the selection of UML in conjunction with both profiles, SysML and MARTE: "All requested viewpoints (16) are fully covered by using: UML and SysML and

MARTE (Coverage: 100.0%)", providing the coverage percentage for each of them (as presented in Table 3).

Also, for each stakeholder, the system recommends the corresponding modeling language that covers its specific viewpoints. For example: "Stakeholder 'Hardware Engineer' should use: MARTE (to cover viewpoint(s): Allocation, Resources, Timing, Communication) and UML (to covers viewpoint(s): Deployment, Timing, Communication) and optional SysML (to cover viewpoint(s): Allocation)".

Table 3: Mandatory recommended modeling languages to achieve complete viewpoints coverage.

Recommended modeling language	Coverage
UML	68.8% covers 11 out of 16 viewpoints
SysML	62.5% covers 10 out of 16 viewpoints
MARTE	43.8% covers 7 out of 16 viewpoints

To ensure complete stakeholder-relevant perspective coverage in the industrial automation CPS field, the system proposes all relevant modeling languages that may address each perspective. In the majority of cases, there is more than one modeling language capable of serving the same perspective (e.g., both UML and MARTE can serve Timing and Communication for the Hardware Engineer). Presenting all relevant options gives freedom in model selection.

Nevertheless, certain perspectives important to individual stakeholders are addressed by only one modeling language. For example, the Non-Functional Requirements perspective important to the System Architect is facilitated by SysML alone. In these cases, the system must incorporate that specific modeling language for full viewpoint representation for all stakeholders.

This method guarantees that every issue for the stakeholders is adequately represented without precluding flexibility when various languages can be used for similar functions.

5 CONCLUSIONS

This study proposes a rule-based solution to the recommendation of modeling languages in CPS development, thereby focusing on one of the biggest challenges in the assurance that the viewpoints of all stakeholders are properly addressed. From a pre-established mapping model and based on a rule-based system implemented with Drools, the proposed recommendation system determines a language or a set of languages that cover end-to-end modeling of the viewpoints of concern to a particular stakeholder group. It aims the removal of modeling gaps and offers a clear, traceable justification for language choice.

Employment of formal specifications improves clearness and reliability, characteristics that are typically absent from the conventional selection processes of modeling languages.

The applicability of the proposed methodology was demonstrated in an industrial automation CPS case study. The system produced an enriched set of outputs, covering language pairs in terms of quantified coverage, per-modeling language contribution, and stakeholder-specific guidance.

Currently, the system analyzes the possibilities offered by three modeling languages (i.e., UML, SysML, and MARTE). Future work is expected to expand its potential to supply additional modeling languages and viewpoint ranking, enhancing its applicability across domains with different and new stakeholder requirements, intricate testing activities using automated generated use cases and a user-friendly interface.

REFERENCES

Aslam, M. et al. (2025). Cyber Security in Cyber Physical Systems. In *AI-Enhanced Solutions for Sustainable Cybersecurity*, pp.149–180.

Azzouzi, E et al. (2022). A Model-Based Engineering Methodology for Stakeholders Coordination of Multienergy Cyber-Physical Systems. In *IEEE Systems Journal*, vol. 16, no. 1, pp. 219–230.

Barišić, A et al. (2022). Multi-paradigm modeling for cyber–physical systems: A systematic mapping review. In *Journal of Systems and Software*, vol. 183, p. 111081.

Broman, D et al. (2012). Viewpoints, formalisms, languages, and tools for cyber-physical systems. *Proceedings of the 6th International Workshop on Multi-Paradigm Modeling*, ACM, New York, NY, USA, pp.49–54.

Cederbladh, J et al. (2024). Experiences and challenges from developing cyber-physical systems in industry-academia collaboration. In *Software: Practice and Experience*, vol. 54, no. 6, pp. 1193–1212.

Daun, M. and Tenbergen, B. (2023). Context modeling for cyber-physical systems. In *Journal of Software: Evolution and Process*, vol. 35, no. 7.

Grimm, S. (2009). Knowledge Representation and Ontologies. In *Scientific Data Mining and Knowledge Discovery*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp.111–137.

Hamzah, M et al. (2023). Distributed Control of Cyber Physical System on Various Domains: A Critical Review. In *Systems*, vol. 11, no. 4, p. 208.

Hille Pascal Van et al. (2012). Comparing Drools and ontology reasoning approaches for telecardiology decision support. In *Studies in Health Technology and Informatics.*

Huang, B., He, Z., and Tang, Y. (2020). Application Research of Business Rules Engine Management System Based on Drools. In *Cyber Security Intelligence and Analytic,* pp.238–242.

'ISO/IEC/IEEE Systems and software engineering -- Architecture description'. 2011.

Jung, CY., Sward, KA., and Haug, PJ. (2012). Executing medical logic modules expressed in ArdenML using Drools. In *Journal of the American Medical Informatics Association*, vol. 19, no. 4, pp. 533–536.

Kaczor, K et al. (2011). Visual Design of Drools Rule Bases Using the XTT2 Method. In *Semantic Methods for Knowledge Management and Communication. Studies in Computational Intelligence*, vol 381. Springer, Berlin, pp.57–66.

Luo, X et al. (2021). A scalable rule engine system for trigger-action application in large-scale IoT environment. In *Computer Communications*, vol. 177, pp. 220–229.

Mallet, F., Villar, E., and Herrera, F. (2017). MARTE for CPS and CPSoS. In *Cyber-Physical System Design from an Architecture Analysis Viewpoint*, Springer Singapore, Singapore, pp.81–108.

Opranescu, V. and Ionita, AD. (2025). Review of Cyber-Physical Systems Modeling With UML, SysML, and MARTE. In *IEEE Access*, vol. 13, pp. 47132–47145.

Opranescu, V., Anghel, A. M., Ionita, A. D. (2024). Study of Ontologies and Rule Engine Integration Applied to Sensor Networks. In *2024 28th International Conference on System Theory, Control and Computing (ICSTCC)*, Sinaia, Romania.

Ordinez, L et al. (2020). Using UML for Learning How to Design and Model Cyber-Physical Systems. In *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 15, no. 1, pp. 50–60.

Parant, A et al. (2023). Model-based engineering for designing cyber-physical systems from product specifications. In *Computers in Industry*, vol. 145, p. 103808.

Proctor, M. (2012). Drools: A Rule Engine for Complex Event Processing. In *Applications of Graph Transformations with Industrial Relevance. AGTIVE 2011. Lecture Notes in Computer Science,* vol 7233. Springer, Berlin, Heidelberg.

Rahatulain, A. and Onori, M. (2018). Viewpoints and views for the architecture description of cyber-physical manufacturing systems. In *Procedia CIRP*, vol. 72, pp. 450–455.

Shahin, M et al. (2019). An empirical study of architecting for continuous delivery and deployment. In *Empirical Software Engineering*, vol. 24, no. 3, pp. 1061–1108.

Stary, C. (2021). Digital Twin Generation: Re-Conceptualizing Agent Systems for Behavior-Centered Cyber-Physical System Development. In *Sensors*, vol. 21, no. 4, p. 1096.

Taha, WM., Taha, A-EM., and Thunberg, J. (2021). Cyber-Physical Systems: A Model-Based Approach. In *Springer International Publishing*, Cham.

Weyer, S. et al. (2016). Future Modeling and Simulation of CPS-based Factories: an Example from the Automotive Industry. In *IFAC Workshop on Intelligent Manufacturing Systems (IMS),* vol. 49, no. 31, pp. 97–102.