Towards Security Architecture Modeling for Modular Automation Systems

Volodymyr Trykoz*, Björn Leander*†, Saad Mubeen*, Mohammad Ashjaei*
†ABB Process Automation, Process Control Platform, Västerås, Sweden,
bjorn.leander@se.abb.com
*Mälardalen University, Västerås, Sweden, firstname.lastname@mdu.se

***Abstract*—Software and system architecture modeling is a well-established technique for addressing design-time challenges and enabling validation early in the system and software development process. However, software and system modeling has not been as widely used in automation systems as it has been in the automotive industry. In addition to validating functional properties through modeling, security properties and features can be evaluated in the modeling phase. This paper provides a brief overview of existing modeling approaches for future automation systems, known as modular automation, that support security modeling and analysis. Modular automation refers to a new trend in automation in which various modules are developed in advance and integrated to build an automation system. Moreover, we study the gaps and missing components of the existing modeling approaches via a use case demonstration. Our study shows that the current state of automation and control system modeling has several limitations that require urgent attention.**

I. Introduction

Industrial automation and control systems supervise a wide range of industrial systems, including power generation and distribution, process industries, and production assemblies. The safe and secure operation of these systems is of utmost importance as their malfunction can have significant consequences on both economy and efficiency in services that they provide. The conventional automation architectures commonly rely on hierarchical controllers that are responsible for centralized management and control of a set of input and output signals. However, the hierarchical architecture is constrained by various limitations, including inadequate flexibility, high latency due to multiple layers, and difficulties in retrofitting different components in automation [1]. Consequently, these limitations impede the automation system's ability to adapt to evolving production needs and innovative technological solutions [2].

Modular automation [3], [4] is an emerging design strategy that provides a set of desired characteristics for automation systems, such as high flexibility, easy scale up/scale down and fast innovation cycles. Modular automation overcomes the limitations posed by conventional automation systems. However, modular automation, despite the benefits in flexibility, imposes a set of challenges. One of the challenges is the dependability aspects of the system, in particular, security issues that arise due to modularity and openness of the system. This paper focuses on the security aspects of such systems.

To design and pre-evaluate software and system architectures in industries, a system architecture modeling approach is commonly used. Although it is mostly common in the automotive domain [5], the automation industry has also inherited this approach to validate various requirements in the design phase [6]. However, current approaches for modeling and analyzing security properties of automation systems focus on conventional system architectures, such as the top-down and controller-centric ones. To the best of our knowledge, no studies have evaluated these approaches in the context of modular automation, where a network-centric and service-oriented approach is more common.

Paper contributions: This paper aims at evaluating existing approaches for security modeling and analysis in modular automation systems. The concrete contributions are as follows:

- We provide a detailed description of security aspects that can arise in modular automation systems that may not exist in conventional automation systems.
- We select a promising existing modeling approach based on a previous literature study to evaluate a modular automation use case example. Via the experiment, we show the potential and feasibility of the approach as well as the limitations and tool gaps. We discuss further developments that are required to enhance the selected approach to support modeling of security aspects in modular automation systems.

II. Background and Related Work

This section will introduce a relevant theoretical framework as well as establish existing works in the area.

A. Modular automation

Modular automation is an emerging approach for structuring industrial automation systems which emphasizes decentralized and service-oriented architectures [3]. These modular architectures rely on self-contained, pre-configured modules that expose standardized interface information via Module Type Packages (MTPs). Using the information provided in MTPs, external actors can interact with a service endpoint and request services from the module.

Modules are typically specialized, meaning each can only perform a small subset of the total work required for product manufacturing. An *orchestrator* is a special component responsible for coordinating different modules, making sure they execute a pre-defined production workflow called a *recipe*. The orchestrator reads the recipe and requests specific services from various modules so that the workflow is correctly executed.

A typical modular setup implies a central network to which all module endpoints and the orchestrator are connected. Additional components, such as engineering or operator workstations, may also be present on the network, allowing human

operators to monitor and control the production process. Security devices, such as a server that authorizes communication between components to prevent unauthorized access, might be included as well. Each module typically contains an internal network of components that is supposed to be isolated from the central network. Interaction between the internal and central networks occurs via the service endpoints, e.g., using the Open Process Communication Unified Architecture (OPC UA) [7]. Moreover, OPC UA can be used to provide authorization services in systems implementing a zero-trust policy [8].

B. State of the art

In a previous study from 2025 we identified 25 publications that were focused on modeling security aspects of automation systems published between 2015 and 2024 [9]. Among the 25 publications, 23 unique approaches were identified with different focus areas and research contributions. The literature review looked at how various approaches perform modeling of automation system architectures and how they conduct security analysis. The publications focused on traditional automation systems rather than modular automation as the concept is recent.

Among various approaches, we focused on AMLsec [10] which is an extension of AutomationML (AML)[1] that adds several security-related features. It covers a wide range of system components, from low-level devices such as sensors and actuators to higher-level management systems. Moreover, the MTPs are based on AML, hence using AML as the primary modeling language for describing the system would be beneficial.

Another important aspect for choosing the modeling approach was the availability of tools. The majority of the studies identified in [9] did not have any tooling support available. Even if a solution was developed as a part of the work, it was often not available to the public. That left us with only a handful of options to choose from. AMLsec [10] had a publicly available tool with up-to-date instructions. It is open source and can be downloaded from GitHub[2]. Describing the details of the literature review and methodologies are out of the scope of this paper, while interested readers are referred to [9] for further information.

There have been other papers that focused on surveying or evaluating existing approaches for automation system security modeling. Geisman and Bodden [11] provides a systematic literature review aimed at identifying and reviewing modeling approaches for security aspects of cyber-physical systems. The study identified seven relevant approaches and described their strengths and weaknesses.

Similarly, Mashkoor *et al.* [12] is looking at model-driven approaches for combined safety and security of software systems. They found that while there is support for modeling security aspects of software systems in general, there is a lack of tools and methodologies.

Hosseini *et al.* [13] conducted an experimental evaluation of an existing modeling approach called SysML-sec using a Cobot system architecture as a toy example. SysML-sec is a dialect of SysML with an emphasis on system architecture security. In that same work Hosseini *et al.* [13] designed the Cobot system following the approach based on a reference architecture. They found that the approach could be used for designing Industry 4.0 systems with some drawbacks, such as limited capability for risk analysis, model checker inefficiency when operating on large models, and absence of compliance to a specific standard. Moreover, the work used a SysML-based environment, whereas AMLsec [10] is based on AutomationML in combination with OWL [14].

None of the previous works considers the specifics of modeling modular architectures, which is the main objective of our work.

III. Security Modeling Requirements in Modular Automation

Modular automation systems inherit many security requirements from the broader domain of automation systems, but they also introduce domain-specific challenges. The traditional concept of demilitarized zones is becoming less relevant, giving way to more individualized, identity-centric security models [15]. Given their inherently flexible design, there are compelling arguments for adopting zero-trust security mechanisms instead of the zone-based security policies [8].

Common for zero-trust systems [16] is reliance on access control mechanisms that manage access to system components and resources. These mechanisms incorporate entities for policy decisions and policy enforcement. The role of the policy components is to determine whether a system entity should be trusted, and what resources it will be allowed to access. With the zone-based approaches, there is implicit trust between components that are assigned to the same zone. By contrast, in a zero-trust environment, each actor in the system is considered individually, on a case-by-case basis. The actors are not placed into zones with implicit trust and they do not trust each other by default. In order to make the necessary interactions happen, trust has to be actively established between the components. Necessary permissions should also be given to the actors on the network. The difficulty lies in determining the appropriate level of trust and permissions for each of the actors on the network.

Achieving this requires careful planning and a clear understanding of the roles different components play on the network. Moreover, it is important to know how various components are expected to operate and how they are connected to other actors in the system. These considerations can complicate the system design, especially when designing a modular system where users or devices on the network might change rapidly. Modeling the zero-trust systems and associated actors can reduce the complexity by allowing the users to work with high-level abstractions when designing their system. This can simplify reasoning about the system and it will also leave less room for errors during the implementation phase.

Modeling these aspects can be approached in two complementary ways:

- **Device authorization component:** at least one dedicated component should be present on the network, that is supposed to be responsible for managing onboarding and

[1] https://www.automationml.org/

[2] https://github.com/sbaresearch/amlsec

offboarding of devices. In addition, this component is responsible for granting permissions according to defined access policies and establishing trust between the components.

- **Permission annotations on connections or components:** when one component needs to interact with another, the connection between them should be explicitly annotated as authorized. During the modeling phase, this enables clear tracking of which components are permitted to communicate. Before the analysis phase, an authorization policy is defined. During analysis, any deviation between the modeled permissions and the defined policy can be identified and reported as a violation.

IV. Modular Automation Use Case Evaluation

Based on the literature review presented in Section II, the AMLsec approach is selected for evaluation due to its foundation in AutomationML and its open-source availability. The research community has proposed leveraging AutomationML to validate the transfer of information via MTPs as well as for modeling and simulating automation systems [17]. Consequently, using it as a modeling language can make it easier to convert between models and actual implementations for modular systems. At the same time, other approaches also exist for modeling similar system architectures. One such example is IEC 61499, which uses function blocks to represent components in a distributed system [18]. It is important to point out that IEC 61499 focuses on functional modeling, emphasizing functionality and interactions between components rather than the components themselves. Modular automation systems are typically composed of complex objects, such as subsystems comprising many devices, which impose different requirements on modeling. The modeling approach must support large-scale, system-of-systems views that encompass both individual devices and groups of devices. While other approaches for modeling large systems exist, such as SysML, the connection between AutomationML and modular automation makes AMLsec our modeling language of choice.

To establish a baseline for the tool's capabilities, it is initially applied to model a modular architecture use case without any extensions. Below, we present the use case itself, the modeling process, and the preliminary results from the baseline evaluation.

A. Use case modeling

Fig. 1 shows the use case example of an ice cream factory process automation for the evaluation. The example consists of an orchestrator, an engineering and operator workplaces, six Programmable Logic Controllers (PLCs) performing the roles of service endpoints, and a switch that connects the components.

The architecture is modeled with AutomationML using the AMLsec extension [10], which allows modeling two connection types: physical and logical. For this specific use case, the physical connections are modeled as Ethernet cables. The POL Switch acted as the central communication hub where all cables are plugged in. Controllers, the orchestrator, and the workplaces are assumed to have at least one Ethernet socket. In terms of logical connections, OPC UA is used as a primary way of communication between the components. At the same time, deliberate flaws are added to the system design to test the approach under investigation. The orchestrator has a direct logical connection to each of the controllers using the OPC UA protocol. The engineering workplace has an HTTP connection with the orchestrator, which is a deliberate miss, and designed to test the security analysis tool.

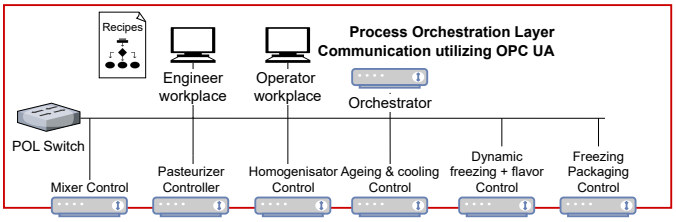

Fig. 1: Architecture of the ice cream factory used as a toy example in this study

B. Modeling expressiveness

It is possible to model the architecture at a high level. The modeling framework allows for separation between logical and physical connections, which is a common consideration in modular architectures [4], though it is not exclusive to this domain. It is also possible to model OPC UA communication between different components by adding logical endpoints to each of the components and connecting them via the logical network. However, there are several limitations in the framework which make it difficult to model the use case architecture. The modeling framework has no concept of an orchestrator. Therefore, we represented it as a PLC in this model, which can assume the role of an orchestrator in such systems. Defining such a dedicated component can provide further analysis possibilities. Moreover, it is not possible to separate the system into different modules. The only possibility is to use zones representing different modules. The concepts of modules, service endpoints, and an orchestrator would provide meaningful additions to the current modeling approach, allowing the users to better express, depict, and annotate modular architectures.

TABLE I: Support for concepts relevant to modular automation architectures

Concept	Support	Notes
Logical connections	Y	
Module	N	
MTP	N	
OPC UA	Y	Present as OPC UA endpoints
Orchestrator	N	Modeled indirectly via the PLC concept
Service endpoint	N	Modeled indirectly via the PLC concept
Physical connections	Y	
PLC	Y	
Recipe	N	
Switch	Y	

C. Use case analysis

The tool reported two vulnerabilities in the architecture which are unused logical endpoints and temporary connections within the same zone. The unused logical endpoints are deliberately added to the operator workstation to trigger the analysis engine. The tool did not detect the use of the insecure HTTP protocol that is employed for communication

between the operator workstation and the orchestrator. The rule regarding temporary connections within the same zone states that devices that may be temporarily connected to the network, such as the operator, should be placed in separate zones. This rule comes from the IEC 62443-3-2 standard, which is applied when dividing a system into zones that are connected via conduits [10]. While this approach works well for conventional automation systems, it has limitations when applied to modular automation systems [19]. In this case, the system is modeled as a single zone, which led to a violation related to temporary connections. None of the other IEC 62443-based rules are triggered in the analysis.

In addition to the above, the analysis engine reports vulnerabilities from Common Vulnerabilities and Exposures (CVE) entries when concrete devices are modeled instead of abstract components. Currently, the device list needs to be updated manually to ensure that it contains the most recent models. The tool consults the SEPSES Cybersecurity KG endpoint to check for available CVE entries [10]. This requires manually adding models for specific physical assets which might not always be known during design time. For the purposes of this work, generic components are used instead that cannot be used for CVE checks.

D. Discussions and observations

As mentioned, modular automation systems rely on zero-trust policies rather than dividing the architecture into fixed zones. Because of this, the current emphasis on zoning could be omitted to better align with zero-trust principles and improve the modeling experience. An alternative to the zone-based approach is to introduce the concept of modules into the modeling approach and tool. This allows to separate assets in the system in a way that is meaningful to modular design.

Currently, there is no concept of an authorization service or authorized connections in the model. This means that it is not possible to represent the authorization security features, which can be important in many use cases. Emulation through the already existing components, such as using PLC components instead of modeling the orchestrator, is not feasible due to different security priorities of the modeling approach. There is currently no capability to define authorization policies that specify what kind of access should be granted to different components or modules.

V. Conclusion and ongoing work

In this paper, we identified that AMLsec modeling approach is a promising modeling approach for security properties of modular automation systems. To evaluate the feasibility of the selected modeling approach, we conducted a use case study modeling an ice cream factory's automation process. According to the use case evaluation, the current approach still has limited expressiveness for modular automation systems in both system modeling and security analysis. It includes some of the common components found in modular architectures such as a PLC and a network switch. It also allows modeling of physical and logical connections separately, which helps to represent how devices are connected and how they communicate. However, the approach and its tool lacks important modular automation concepts such as the orchestrator, MTP representation, and recipes. The security focus is on zoning instead of zero-trust, and there is no concept of authorization or support for authorization policies.

Ongoing work aims at adding the components and concepts that are related to modular automation systems. In particular, we are adding components such as an orchestrator component, an authorization server along with authorization policies, and adding possibility of splitting components into modules. Additionally, a more complex use case architecture is envisioned for the evaluation of the modeling approach in the future work.

Acknowledgments

The work is supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the FLEXATION project; and by the Swedish Knowledge Foundation (KKS) via MARC and SEINE projects. We would like to thank all our industrial partners, particularly ABB and Westermo.

References

[1] P. Dimitrov and M. Alexandrova, "From automation pyramid to industry 4.0: Transitioning process and practical applications," in *2024 International Conference Automatics and Informatics*, 2024.

[2] C. Lucizano *et al.*, "Revisiting the automation pyramid for the industry 4.0," in *2023 15th IEEE International Conference on Industry Applications*, 2023.

[3] ZVEI—German Electrical and Electronic Manufacturers' Association, "Process industrie 4.0: The age of modular production," White Paper, August 2019.

[4] B. Leander, T. Marković *et al.*, "Enhanced simulation environment to support research in modular manufacturing systems," in *49th Annual Conference of the IEEE Industrial Electronics Society*, 2023.

[5] L. Lo Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent advances and trends in on-board embedded and networked automotive systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, 2019.

[6] D. Ardagna *et al.*, "Rethinking the use of models in software architecture," in *Quality of Software Architectures*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5281.

[7] "IEC 62541 OPC unified architecture," IEC, Geneva, CH, Standard, 2016.

[8] I. Radonjić *et al.*, "An authorization service supporting dynamic access control in manufacturing systems," in *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, 2023, pp. 01–06.

[9] V. Trykoz, "Security Modeling for Automation Systems Architecture: A Systematic Literature Review," Master's thesis, Mälardalen University, School of Innovation, Design and Engineering, 2025.

[10] M. Eckhart *et al.*, "Automated security risk identification using AutomationML-based engineering data," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1655–1672, 2022.

[11] J. Geismann and E. Bodden, "A systematic literature review of model-driven security engineering for cyber–physical systems," *Journal of Systems and Software*, vol. 169, no. 110697, 2020.

[12] A. Mashkoor *et al.*, "Model-driven engineering of safety and security software systems: A systematic mapping study and future research directions," *Journal of software: evolution and process*, vol. 35, no. 7, 2023.

[13] A. M. Hosseini *et al.*, "Formal verification of safety and security properties in industry 4.0 applications," in *2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation*, 2023.

[14] "Owl web ontology language guide," W3C, Wakefield, US, Standard, 2005-2013.

[15] C. Zanasi *et al.*, "A zero trust approach for the cybersecurity of industrial control systems," in *2022 IEEE 21st International Symposium on Network Computing and Applications*, vol. 21, 2022.

[16] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST, Gaithersburg, MD, Tech. Rep., 2020.

[17] S. Sarkar, K. Stark, and M. Hoernicke, "Design of a validator for module type packages," in *48th Annual Conference of the IEEE Industrial Electronics Society*, 2022.

[18] "IEC 61499 function blocks - part 1: Architecture," IEC, Geneva, CH, Standard, 2005-2013.

[19] B. Leander *et al.*, "Applicability of the IEC 62443 standard in Industry 4.0/IIoT," in *14th International Conference on Availability, Reliability and Security (ARES)*. ACM, 2019.