2025 Forum on Specification & Design Languages (FDL) | 979-8-3315-9379-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/FDL68117.2025.11165407

Digital Twin and Digital Thread for System Security and Performance applied to an Electrical Vehicle Charging Use Case

Special Session Paper

Hagen Heermann, Johannes Koch, Christoph Grimm
Cyber-Physical Systems Chair
University of Kaiserslautern-Landau
Kaiserslautern, Germany
heermann|johannes.koch|cgrimm@rptu.de

Daniela Genius
LIP6 Laboratory
Sorbonne Université
Paris, France
daniela.genius@lip6.fr

Ludovic Apvrille
LTCI, Télécom Paris
Institut Polytechnique de Paris
Sophia-Antipolis, France
ludovic.apvrille@telecom-paris.fr

Ahlem Mifdaoui
ISAE-Supaero
University of Toulouse
Toulouse, France
ahlem.mifdaoui@isae-supaero.fr

Klaus Schneider
Embedded Systems Chair
University of Kaiserslautern-Landau
Kaiserslautern, Germany
klaus.schneider@rptu.de

***Abstract*—System security requires a solid foundation in both development and operation. During development, performance trade-offs result in security infrastructures that are more or less effective, but usually imperfect. Hence, during operation, runtime monitoring and anomaly detection continuously check for security issues.**

In this paper, we show how development and operation can be linked. We demonstrate how information and data from development and operation can be aggregated in a digital twin and/or digital thread which is used as the basis for runtime monitoring and anomaly detection. In particular, we address the trade-off between system security and performance in a concrete smart grid system.

***Index Terms*—digital twin, digital thread, cyber-physical systems**

I. Introduction

System security is a critical concern at every stage of a system's lifecycle, from development to operation. During development, engineering trade-offs often prioritize performance, functionality, or cost over achieving an ideal security infrastructure. As a result, many systems enter operation with vulnerabilities or imperfect protections. Addressing these gaps necessitates robust mechanisms during operation, including runtime monitoring and anomaly detection, to detect and mitigate emerging security threats effectively. Ideally, these mechanisms should be foreseen throughout the design process, from the earliest phases onward.

Cyber-physical systems (CPS) integrate computational and physical processes, where embedded systems monitor and control physical components in real time. However, bridging the gap between development and operation poses significant challenges. Security vulnerabilities often result from incomplete or siloed information transfer between these phases. Furthermore, addressing novel security issues requires navigating trade-offs between correctness, efficiency and security. Over-prioritizing one aspect can negatively affect others, for example, enforcing overly strict security measures may degrade system performance.

To address this issue, we propose to leverage *digital twin and digital thread technologies*. Digital twins are virtual representations of physical systems that enable real-time monitoring, simulation, and analysis. Digital threads build on this concept by providing a connected data flow that spans the entire system lifecycle, linking data from development, production, and operational phases. Together, these technologies can provide an integrated foundation for runtime monitoring and anomaly detection, especially with respect to system security.

In this paper, we explore the application of digital twin and digital thread technologies to enhance security in cyber-physical systems. Our contribution includes the following novelties:

1) *Security/Performance-Focused Application of Digital Twin and Digital Thread:* We present a novel framework that prioritizes system security while guaranteeing performance, linking development-phase insights with operational monitoring for runtime anomaly detection. This focused use of these technologies goes beyond their traditional roles in maintenance and performance optimization.
2) *Application to an Electrical Vehicle Charging Use Case:* We demonstrate the framework in the context of a smart grid and Electrical Vehicles (EV) charging application, a critical infrastructure with unique security challenges

979-8-3315-9379-7/25/$31.00 ©2025 IEEE

such as distributed control, real-time constraints, and 5G inter-connectivity. This tailored application provides a practical example of addressing such challenges.

By integrating data from development artifacts and operational performance, we show how to identify potential anomalies and improve the security posture of such systems. This work offers a novel perspective on addressing security challenges in critical cyber-physical systems under strict performance requirements and provides a blueprint for future applications in other domains.

II. State of the Art

The concepts of digital twins and digital threads have been increasingly recognized for enhancing the security and resilience of cyber-physical systems, particularly in smart grid applications. These technologies offer dynamic, real-time representations of physical systems, enabling advanced monitoring, predictive maintenance, and anomaly detection. However, their seamless integration into security-critical domains when taking into account strict performance requirements, as in smart grids, remains a research challenge.

It is sometimes difficult to separate the notions of digital twins digital thread. Digital threads contain a data monitoring component, but more recently, it is included in the definition of "digital twin" [1]).

A comprehensive survey by Zheng [2] explores the role of Digital Twins in bolstering cybersecurity within smart grids. The study emphasizes their potential to simulate and analyze cyber-physical interactions, thereby facilitating the identification and mitigation of vulnerabilities. However, the authors note that practical implementation remains challenging due to the complexity of integrating Digital Twins into existing infrastructures.

In [3], a framework leveraging the FIWARE platform to enhance cybersecurity monitoring in smart power and distribution grids has been proposed. This framework aligns with the Common Information Model (CIM) and integrates data spaces and DTs to improve interoperability and real-time data analysis. The authors demonstrate the framework's effectiveness through its application to a real-world power distribution grid in Kropa, Slovenia.

Teixeira et al. [4] propose an attack-graph-based solution for quantitative security analysis of CPS, and a set of model transformations rules to bridge the gap between SysML and the underlying security tool MulVAL.

SysMLSec [5] addresses the problem with a two-level approach based on SysML formal modeling, supported by the public domain software TTool [6]. One level targets high-level system architecture and behavior, while the other addresses modeling fine-grained hardware and low-level security. Vulnerabilities can be identified using the ProVerif [7] formal verification and countermeasures proposed. W-Sec [8] approach from the same group proceeds in four stages.

In the realm of anomaly detection, Raman and Mathur [9] introduce a hybrid physics-based and data-driven framework tailored for industrial control systems. This approach combines physical models with data analytics to detect anomalies, offering a robust solution for systems where extensive historical data may be lacking. While not specific to smart grids, the methodology provides valuable insights applicable to similar CPS environments.

Furthermore, Heermann and Grimm [10] present the concept of H-classifiers, aiming to bridge the gap between anomaly detection and runtime verification. This approach leverages formal models to simulate expected system behaviors, facilitating the detection of deviations indicative of potential security threats. Such methodologies are particularly beneficial in scenarios where data-driven approaches are constrained by limited historical data.

Koch [11] extends this discussion by emphasizing the importance of *knowledge modeling* in the context of power grids. Their work leverages SysML-based modeling approaches to enhance system-level understanding and traceability in CPS, with a focus on smart grids. The study underscores how structured modeling techniques can act as a foundational element for integrating security and resilience into CPS design and operation.

Existing works in the area of smart grid discuss the open issues of communication technologies like 5G when focusing on one specific requirements like performance, safety or security. However, the early design and verification of networks in this domain when taking into account security and performance is a key issue to correctly design the whole system. The most relevant compositional approach in this specific area is the Network Calculus [12], that derives maximum bounds on delays and backlogs in complex communication networks. Recent existing work analyzes the effect of safety on performance performance for time-sensitive networks [13]. However, to the best of our knowledge, there is no existing work to compute accurate performance bounds when taking into account security issues at both network and system levels.

Despite these advancements, challenges persist in the seamless integration of digital twins and digital threads into smart grid infrastructures. Issues such as data interoperability, real-time processing capabilities, and the development of standardized frameworks require further research and development. Addressing these challenges is crucial for realizing the full potential of digital twin and digital thread technologies in enhancing the security, performance and resilience of smart grids.

III. Systems Engineering Phase

The overview of the proposed tool chain is illustrated in Fig. 1. In this section, we focus on the first step, the systems engineering phase.

The development usually starts with the definition of requirements, and a specification. First analysis of use cases and requirement in this phase often lead to the first high-level architectural decisions and state machines. Nowadays, in particular model-based systems engineering (MBSE) using modeling languages like SysMLv2 [14] are used.

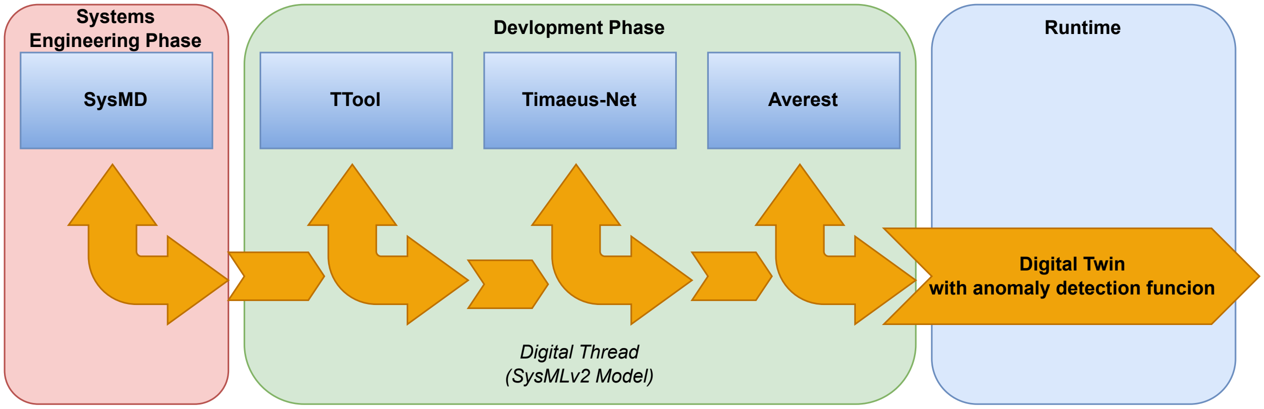

Fig. 1. Overview of the proposed tool chain: interaction with the digital thread and the generation of the digital twin.

SysMD [15] combines SysMLv2 with a constraint propagation solver and symbolic methods. This permits to model uncertainties by ranges, and to propagate them through different kind of structure and behavior. As a result, one gets, for system level quantities that occur in the MBSE models, symbolically encoded ranges and sequences of ranges of feasible values in space and time that are consistent. They can be used during operation as an indicator for anomalies like security breaches.

TABLE I
INFORMATION FROM MBSE FOR MONITORING.

Data	Use cases during operation
Constraints	Monitors for values
Use cases	Automata for runtime-verification
Tolerances	Deviations that monitors/verifiation shall accept

IV. Development Phase

In this section, we describe the different tools to enable the second step of the proposed chain, depicted in Fig. 1.

A. TTool

TTool [6][1] is a free and open-source toolkit. Based on UML/SysML diagrams, it integrates simulation and formal verification at the push of a button. It also includes methods dedicated to the design of embedded systems.

In TTool, as depicted in Figure 2, Model-Based Engineering of (digital) embedded systems can be performed at different abstraction levels, grouped into two subsets: *partitioning* (high level) and *software design* (low level). Specific SysML views and diagrams have been defined for each abstraction level. The partitioning level features two sublevels.

1) The purely *functional level* describes both the structure of functions and their behavior. Complexity of operations can be abstracted with complexity operators, assuming a logical time.
2) The *(system-level) mapping level* gives a physical time to al operations, including complexity operations. However, the highly abstracted hardware components of our approach make the physical time still imprecise: verification results—which might be used as a partitioning decision—are expected to be confirmed during the next abstraction levels.

At software design level, tasks are further detailed and then deployed on more concrete hardware components. Thus, software deployment intends to experiment the interaction of software with all other components (digital and analog). The software design level also includes two sublevels.

1) At the software component design level, high-level timing constraints of software components can be evaluated by interactive simulation or formal verification, yet without hardware.
2) The *deployment level* allows designers to deploy software components onto hardware elements (CPU, memory, etc.) and then to generate a virtual prototype, or to simply generate the software code and deploy it on a physical platform.

For digital platforms, TTool can generate a SystemC-based cycle-precise virtual prototype from the lowest abstraction level, using the SoCLib library [16]. For mixed digital/analog platforms, TTool can generate a virtual prototype for SystemC/SystemC AMS co-simulation [17].

One of the strengths of TTool is to offer property verification from most diagrams [6], relying either on intensive simulation or on formal verification. This concerns safety, security and performance properties. Typical safety properties are the absence of deadlock situations, the reachability of model elements, and more generally any property that can be expressed in CTL. Security properties are: confidentiality, integrity, authenticity. Performance properties that can be directly captured are currently the latency (min, max and average) between two events, CPU and bus load, and others.

[1] https://ttool.telecom-paris.fr

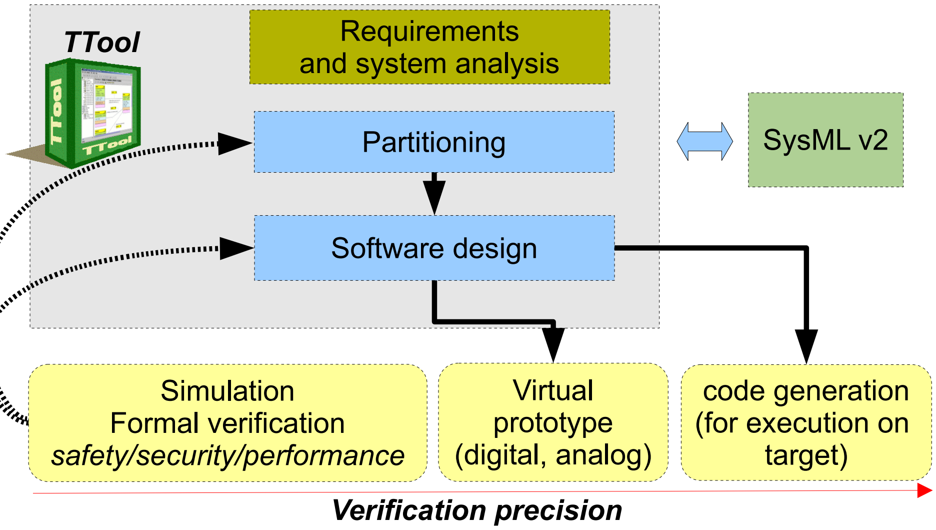
TTool
Requirements and system analysis
Partitioning
Software design
SysML v2
Simulation
Formal verification
safety/security/performance
Virtual prototype (digital, analog)
code generation (for execution on target)
Verification precision

Fig. 2. Overview of the architecture of TTool

More generally, observers can be added to models to express more complex safety, security or performance properties.

TTool integrates the SysMLSec methodology described in [5]. In this methodology, attacker models and security countermeasures are described and then verified using ProVerif [7]. We are currently working on an extension of the deployment level to integrate a precise simulation of security countermeasures into the virtual prototype [18].

Last but not least, TTool can export views in SysML v2 textual format, and can reimport from this format: this makes it possible to exchange models with third party tools.

B. Timaeus-Net

Timaeus-Net [19] is an extended version of WoPANets tool [20] for analyzing and verifying the worst-case performance of real-time networks, particularly in CPS domain. It is a decision-support tool based on Network Calculus [12], a formal and scalable timing analysis approach, destined to analyze the worst-case performance of large-scale real-time communication networks, such in smart grid for which real-time constraints need to be guaranteed. It measures key metrics such as end-to-end delays and backlogs to evaluate the impact of system changes, including the introduction of security mechanisms. Additionally, Timaeus-Net ensures that time and overflow constraints are met, helping to validate the reliability and efficiency of end-to-end architectures, such as those used in charging systems.

However, the use of security mechanisms can increase computation time and communication latencies due to the overheads. Therefore, appropriate proof of transmission time determinism at system level as well as network level should be ensured. Hence, we need first to define analytical models for selected security mechanisms and then to derive end-to-end delay bounds analyses of the global communication architecture. This quantitative analysis will enable the selection of the most suitable security mechanisms at the network level offering the best trade-offs between both aspects, i.e., security and performance. The interdependency between both network and system levels will be carefully analyzed through a feedback loop between TTool and Timaeus-Net.

C. Averest

Averest[2] is a framework for the model-based design of reactive systems that supports the modeling, specification, simulation, compilation, formal verification, and synthesis of hardware and software for reactive embedded systems [21]. It contains compilers for synchronous languages, a simulator for the latter, support for formal verification with temporal and other logics, and various transformations for the hardware and software synthesis of reactive embedded systems which also covers pure hardware circuits and pure software systems.

The focus of the compilation and verification algorithms in Averest is on the use of symbolic algorithms that avoid the enumeration of control-flow states since the even the number of control-flow states may grow exponentially with the size of the programs. The translation from synchronous programs into symbolically represented transition systems has been formally verified against the structural operational semantics of synchronous programs.

Averest is the result of a long-term and still ongoing research effort. The system is available as a .NET NuGet[3] that can run on any platform.

V. Runtime

In this section, we describe the last step of the proposed chain, depicted in Fig. 1, enabling the generation of the digital twin. A specific overview of what happens during the runtime is presented in Fig. 3.

[2] https://www.averest.org
[3] https://www.nuget.org/

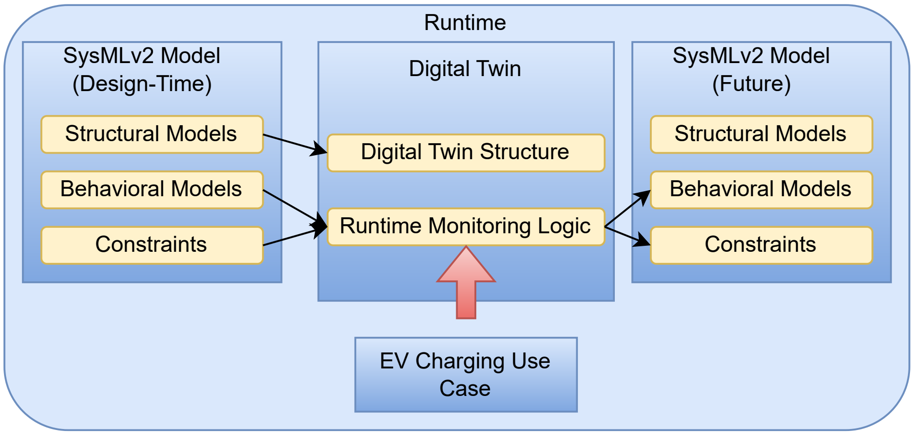

Fig. 3. Overview of the runtime anomaly detection framework based on a model-driven digital twin. The SysMLv2 model at design time provides structural, behavioral, and constraint-based specifications, which are deployed into the digital twin. At runtime, the digital twin observes system behavior—demonstrated here with an EV charging use case—and verifies it against the expected behavior derived from the design-time model. Detected deviations are used to refine behavioral models and constraints, feeding back into the system model for future iterations.

A. Digital Twin Generation

The digital thread, established through a SysMLv2 model that captures all information from the development process, serves as the foundation for generating a digital twin [22]. A digital twin (DT) is a comprehensive digital representation of a physical system or object, continuously synchronized with real-world data from its physical counterpart. Beyond mirroring system behavior, digital twins are increasingly leveraged as active components in system management, analysis, and optimization.

These digital twins are typically used to add functionalities that enhance safety, such as runtime verification, or to provide added-value features like predictive maintenance. In this context, the DT serves as a deployment platform for integrating additional security functionality, thereby enhancing the system's resilience. Among these functionalities, anomaly detection plays a pivotal role in safeguarding cyber-physical systems like smart grids.

Our approach embeds formally verified components within the DT, enabling it to function as both a monitoring and decision-support system during runtime. This tight integration allows the digital twin to detect deviations from expected behavior and initiate countermeasures or alerts before failures propagate. By aligning the DT closely with the SysMLv2-based development artifacts, the generated twin reflects the system's structure and behavior with high fidelity, ensuring consistency between design-time models and their runtime manifestations.

This formal consistency forms the basis for trust in the digital twin's outputs, particularly when applied in safety-critical domains. As the system operates, the digital twin acts not only as a passive observer but also as a knowledge-rich, context-aware runtime component capable of supporting diagnostic and prognostic decisions.

B. Runtime Functionality

Anomaly detection approaches are widely utilized for security purposes across various domains, including fraud detection and similar applications. In cyber-physical systems such as a smart grid, potential attacks often target the physical system itself, making it crucial to detect anomalies in system behavior effectively. Recent advancements, such as those presented in [9], extend anomaly detection to include the physical behavior of the system. However, a common limitation of such approaches is their heavy reliance on data. For novel systems, sufficient data may not yet be available, posing a significant challenge for purely data-driven methods.

To address this limitation, we propose leveraging model-driven approaches that directly utilize behavioral models derived from SysMLv2, as demonstrated in [10]. Rather than depending solely on historical data, our method uses formal models to simulate and evaluate expected system behaviors. This allows for effective anomaly detection even when operational data is sparse, by comparing live system behavior against the model's predictions.

In the context of smart grids, this capability not only enhances security but also ensures resilience against emerging cyber-physical threats, making it a critical component of digital twin frameworks. The runtime engine embedded within the digital twin carries out these verification checks in real time, flagging deviations from intended behavior as soon as they occur. This formal runtime verification capability complements traditional data-driven anomaly detection methods, thereby bridging the gap between design-time specifications and operational monitoring.

Furthermore, the insights gained from runtime anomaly detection can be systematically reintegrated into the development process by refining system models and updating behavioral constraints in SysMLv2. This feedback loop ensures that the system evolves to mitigate detected vulnerabilities and continually enhances its security posture, effectively establishing a living system architecture where runtime observations consistently inform design improvements.

C. Performance Evaluation on the Virtual Prototype

Security measures such as encryption weigh on runtime performance. From TTool, virtual prototypes can be generated, representing both a cycle-bit accurate model of the digital part of the platform and a SystemC-AMS based simulation of the analog part. A virtual prototype does not yet exploit data obtained from real-world scenarios, which is a shortcoming as well as a benefit: in our methodology, evaluation on a virtual prototype intervenes in one of the early design phases, and does not depend on the fact that data has already been obtained and cast in an exploitable format.

Causality between the two models of computation, which are based on discrete and continuous time, respectively, is guaranteed [17]. The library of hardware models on which the digital part of the virtual prototype is based, SoCLib, proposes tools enabling performance evaluation on a cycle and bit-accurate level, such as a logger and throughput analysis, and a memory checker and gdb server to control correctness of execution. Existing counters can be activated in the hardware models written in SystemC, and models can be enriched with new counters. For the analog part, such tools will yet have to be developed. SystemC-AMS features functionalities to generate textual trace files, which tend to be space-consuming (vcd tracing only pertains to the digital part). One of our aims will be to restrict the number of parameters monitored to what is strictly necessary in the application context, and to define a more compact trace format.

Even though performance is evaluated for the virtual prototype in place of the real system, it still gives useful information that can be fed back to the higher abstraction levels as described in [23]. These evaluations can already be done on the software design level described as the second phase in section IV.

VI. Smart Grid Use Case

In the evolving landscape of smart grids [11], the integration of Electric Vehicles (EVs) plays a pivotal role in shaping sustainable energy systems. As EV adoption grows, managing their charging demands efficiently becomes critical to maintaining grid stability and optimizing resource allocation. This requires a seamless coordination mechanism between EVs, Charging Stations (CSs), and energy providers. Leveraging 5G communication technology, a robust framework can be established to enable real-time information exchange, efficient load balancing, and dynamic decision-making.

An illustrative example of such a use case is presented in Fig.4. The 5G connectivity facilitates the interaction between EVs requesting charging services and a centralized Global Aggregator (GA). An EV sends a charging request containing key information such as location, current State of Charge (SoC), and desired final SoC. The GA, based on real-time data from nearby CSs, responds with optimized charging details, including availability, expected charging time, and cost. This system ensures that the charging process is not only efficient but also scalable, as the 5G network can handle numerous simultaneous requests from EVs.

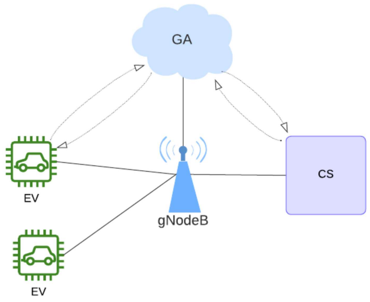

Fig. 4. EV Charging with Smart Grid and 5G connectivity use case

Within the broader smart grid context, this use case highlights the importance of real-time coordination to balance energy demand and supply. The GA acts as a centralized controller, dynamically managing charging sessions based on incoming EV requests and the status of charging infrastructure. To address key requirements such as low latency, high throughput, and robust reliability, 5G technologies—like cryptographic methods for secure communication, beamforming, and adaptive resource allocation—are employed. This ensures secure data exchange, protection of sensitive information (e.g., EV location and payment details), and uninterrupted charging services.

In the evolving landscape of smart grids, the integration of advanced tool chains and frameworks plays a vital role in enhancing system efficiency, security, and scalability. Several specialized tools contribute to this effort by enabling robust modeling, verification, and real-time monitoring within the grid infrastructure. At the heart of these integrations is the digital thread, represented by the SysMLv2 model, which serves as the binding element, connecting various tools and processes seamlessly.

The SysMD tool chain is instrumental in the early development and modeling of uncertainties within the system. It enables the consistent verification of internal consistency, allowing for an initial ”rough sketch” of the smart grid. This ensures that even in the early stages, the foundational framework of the grid is both logical and coherent.

The TTool integration extends the digital thread by accepting behavioral models derived from SysMLv2 as input. These models are further refined and verified, producing validated behavioral representations alongside updates to security and performance metrics on the virtual prototype at design time as described in section V-B. This process ensures that the system is optimized for both functional integrity and operational reliability.

Timaeus-Net enhances the digital thread by providing detailed performance analysis for smart grids to consolidate the selection of security mechanisms. Using SysMLv2 models as input, it evaluates critical metrics such as end-to-end delays and backlogs, verifies time and overflow constraints, and highlights the impact of modifications, such as adding security mechanisms. The tool suggests optimized configurations and annotates critical paths or bottlenecks, feeding validated performance insights back into the SysMLv2 model. This iterative feedback enriches the model, fostering continuous improvement.

The digital twin integration serves as a real-time bridge between the smart grid and its operational environment. By collecting critical data such as voltages and other variables, the digital twin continuously supervises the grid's behavior. Integrated anomaly detection mechanisms identify deviations from expected patterns, classifying them as either natural occurrences, modeling errors, or potential attacks. This system provides actionable feedback during runtime, allowing for parameter adjustments and supporting a dynamic development cycle. More importantly, it ensures that operational data is seamlessly reintegrated into subsequent development phases, fostering continuous refinement and resilience.

Together, these tool chains and integrations, unified by the SysMLv2 digital thread, create a comprehensive ecosystem for managing and evolving smart grid infrastructures. They complement real-time coordination mechanisms, such as those used for managing Electric Vehicle (EV) charging demands through 5G connectivity, by ensuring the underlying grid remains secure, efficient, and adaptable to future challenges.

VII. Conclusion

This paper introduced a novel framework for enhancing the security and performance of smart grid systems through the integration of digital twin and digital thread technologies within a SysMLv2-driven digital thread. The proposed approach builds on a robust toolchain—including SysMD, TTool, Timaeus-Net, and Averest—that tightly integrates with the SysMLv2 model to support system development, runtime monitoring, and iterative improvement processes.

SysMD facilitates the early modeling of uncertainties and the propagation of constraints, providing a foundational layer for consistent and logical system architectures. TTool extends this foundation by enabling behavioral modeling, formal verification, and simulation at various abstraction levels, ensuring that safety, security, and performance properties are validated throughout the development lifecycle. Timaeus-Net complements these capabilities by performing detailed runtime performance analyses, such as evaluating end-to-end delays and verifying time constraints, and by feeding critical insights back into the SysMLv2 model for further refinement. Averest enhances the toolchain by offering symbolic modeling, simulation, and verification of reactive systems, enabling the development of precise and efficient embedded systems while avoiding state explosion problems. Together, these tools ensure a seamless workflow from high-level modeling to runtime integration.

The integration of these tools into a unified SysMLv2 digital thread creates a dynamic link between development and runtime operations. At runtime, the digital twin leverages real-time data to enhance anomaly detection and security functionality. This integration enables the classification of anomalies as natural deviations, modeling errors, or potential security threats and ensures that runtime insights are systematically reintegrated into development processes.

The smart grid use case demonstrated the practical application of this framework, particularly in managing distributed control and real-time constraints within a highly interconnected infrastructure. By aligning development and runtime phases through the digital thread, the framework enhances the system's resilience against emerging threats while ensuring operational efficiency.

Future work will focus on scaling this framework to more complex cyber-physical systems, refining the integration of runtime feedback into the SysMLv2 model, and extending its application to other critical infrastructures. This iterative and tightly integrated approach underscores the potential of digital twin and digital thread technologies to advance system security and resilience.

Acknowledgment

The authors would like to thank ANR and DFG for funding under number ANR No. S24JRAR003, DFG project MoVe4SPS No. 530118585.

References

[1] F. Tao, H. Zhang, and C. Zhang, "Advancements and challenges of digital twins in industry," *Nature Computational Science*, vol. 4, no. 3, pp. 169–177, 2024.

[2] T. Zheng, M. Liu, D. Puthal, P. Yi, Y. Wu, and X. He, "Smart grid: Cyber attacks, critical defense approaches, and digital twin," *arXiv preprint arXiv:2205.11783*, 2022.

[3] Y. He, M. P. Papazoglou, and J. Yang, "Building cyber-resilient smart grids with digital twins and data spaces," *Applied Sciences*, vol. 13, no. 24, p. 13060, 2023.

[4] H. Teixeira De Castro, A. Hussain, G. Blanc, J. El Hachem, D. Blouin, J. Leneutre, and P. Papadimitratos, "A model-based approach for assessing the security of cyber-physical systems," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, 2024, pp. 1–10.

[5] L. Apvrille and Y. Roudier, *Model-Driven Engineering and Software Development.* Switzerland: Springer International Publishing, 2016, ch. Designing Safe and Secure Embedded and Cyber-Physical Systems with SysML-Sec, pp. 293–308.

[6] P. de Saqui-Sannes, L. Apvrille, and R. Vingerhoeds, "Checking sysml models against safety and security properties," *Journal of Aerospace Information Systems*, vol. 18, no. 12, pp. 906–918, 2021. [Online]. Available: https://doi.org/10.2514/1.I010950

[7] B. Blanchet *et al.*, "Modeling and verifying security protocols with the applied pi calculus and proverif," *Foundations and Trends® in Privacy and Security*, vol. 1, no. 1-2, pp. 1–135, 2016.

[8] B. Sultan, L. Apvrille, P. Jaillon, and S. Coudert, "W-sec: a model-based formal method for assessing the impacts of security countermeasures," in *International Conference on Model-Driven Engineering and Software Development.* Springer, 2021, pp. 203–229.

[9] M. R. G. Raman and A. P. Mathur, “A hybrid physics-based data-driven framework for anomaly detection in industrial control systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 9, pp. 6003–6014, 2022.

[10] H. Heermann and C. Grimm, “Bridging the gap between anomaly detection and runtime verification: H-classifiers,” in *Proceedings of DATE 2025*, 2025, pp. 1–7.

[11] J. Koch, A. Wansch, and C. Grimm, “Knowledge modeling of power grids with sysmd,” in *2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, 2022, pp. 1–6.

[12] J.-Y. Le Boudec and P. Thiran, *Network Calculus*, ser. Lecture Notes in Computer Science, J.-Y. Le Boudec, P. Thiran, G. Goos, J. Hartmanis, and J. Van Leeuwen, Eds. Berlin, Heidelberg: Springer, 2001, vol. 2050.

[13] L. Thomas, A. Mifdaoui, and J.-Y. Le Boudec, “Worst-case delay bounds in time-sensitive networks with packet replication and elimination,” *IEEE/ACM Transactions on Networking*, vol. 30, no. 6, pp. 2701–2715, 2022.

[14] SysML v2 Submission Team, “Introduction to the SysML v2 Language – Textual Notation.” [Online]. Available: https://drive.google.com/drive/folders/1VMirOt7aQHyG912eQJETXU606WkAdgVw

[15] A. Ratzke, S. Post, J. Koch, and C. Grimm, “Constructive model analysis of sysmlv2 models by constraint propagation,” in *2024 19th Annual System of Systems Engineering Conference (SoSE)*. IEEE, 2024, pp. 239–244.

[16] D. Genius and L. Apvrille, “Virtual yet precise prototyping: An automotive case study,” in *ERTSS’2016*, Toulouse, 2016.

[17] R. Cortés Porto, D. Genius, and L. Apvrille, “Handling causality and schedulability when designing and prototyping cyber-physical systems,” *Software and Systems Modeling*, pp. 1–17, 2021.

[18] M. Rayon-Richter and D. Genius, “Tool support for precise assessment of software security/performance tradeoffs (extended abstract),” in *Conference on Application of Concurrency to System Design (ACSD)*, 2025.

[19] P. Cuenot, T. Leydier, D. Fruchard, M. Barbero, and Q. Bailleul, “Yet another experience on tsn tools interoperability for critical embedded networks,” in *ERTS2024*, Toulouse, France, June 2024, hAL Id: hal-04672432. [Online]. Available: https://hal.archives-ouvertes.fr/hal-04672432

[20] A. Mifdaoui and H. Ayed, “WOPANets: A tool for WOrst case Performance Analysis of embedded Networks,” in *2010 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD)*, Dec. 2010, iSSN: 2378-4873.

[21] K. Schneider and T. Schuele, “Averest: Specification, verification, and implementation of reactive systems,” in *Conference on Application of Concurrency to System Design (ACSD)*. Citeseer, 2005.

[22] H. Heermann, M. Herzog, J. Koch, and C. Grimm, “Generating digital twins from sysmlv2 models,” in *2024 IEEE 22nd International Conference on Industrial Informatics (INDIN)*, 2024, pp. 1–6.

[23] D. Genius, L. W. Li, L. Apvrille, and T. Tanzi, “Multi-level latency evaluation with an mde approach,” in *6th International Conference on Model-Driven Engineering and Software Development*, 2018.