Balancing CPS Non-Functional Properties through Annotated Pattern-Based Mechanisms

Tiziano Lombardi
DISIM
University of L'Aquila
L'Aquila (AQ), Italy
tiziano.lombardi@univaq.it

***Abstract*—Cyber-physical systems often require the satisfaction of multiple, potentially conflicting, non-functional properties, such as cybersecurity, performance, and energy efficiency. This paper presents research objectives to provide a framework for managing tradeoffs among NFPs in CPSs. The main focus is on multi-formalism, pattern-based modeling to capture modeling patterns and quality-related semantics, enabling quality models' automated generation. As a first result, the process pipeline was defined to systematically obtain quality models from input CPS models. Future work will formalize quality-based pattern annotations and aims to introduce support for ML-driven runtime adaptation.**

***Index Terms*—Cyber-Physical Systems, Non-Functional Properties, Pattern-Based Modeling, Quality Model Generation**

I. Introduction

Cyber-Physical Systems (CPSs) are complex systems that deeply interplay computation and physical processes. CPSs are ubiquitous in critical sectors such as healthcare, manufacturing, and transportation, and their correct functioning often relies on simultaneously achieving multiple non-functional properties (NFPs). Among these, cybersecurity plays a critical role that must be ensured without significantly compromising other quality attributes such as performance, energy efficiency, or reliability. These adversarial concerns are often hard to balance from the early design phases, and existing tools and techniques typically focus on isolated quality attributes, lacking integrated views that highlight their interplay. The presented model-driven approach aims to tackle this gap enabling the formalization, prediction, analysis, and automated balancing of tradeoffs among interleaved NFPs in CPSs.

II. Research objectives

The overall project is structured around three key objectives, aiming to provide both theoretical foundations and practical tool support definition for CPS quality analysis.

A. Multi-Formalism Modeling of Quality Attributes

The aim is to define a multi-formalism, pattern-based modeling framework where system structure and behavior are captured through reusable modeling blocks enriched with quality semantics. These modeling constructs will be formalized using a specific DSL that can be transliterated into a combination of Epsilon Pattern Language (EPL) resources. Each quality-related modeling pattern can be defined to be applied both to domain meta-models to cover families of CPS instances, and usual system model definitions for particular-case analysis. Moreover, they will include abstract templates using Epsilon Generation Language (EGL) that can generate analyzable formal models (e.g., Queueing Networks, Stochastic Petri Nets, Stochastic Automata Networks).

The modeling environment is expected to support collaborative development and integration with machine learning backends. The key insight is to abstract and reuse quality patterns the most while maintaining analyzability and traceability across formalisms.

B. Tradeoff Analysis Through Quality Model Synthesis

The proposed modeling framework includes model transformation pipelines that convert quality-annotated CPS models or meta-models into multiple analyzable models using well-known formalisms. The presented approach aims to allow the synthesis of multi-objective quality models that can be evaluated individually or in combination to explore tradeoff spaces. This process can be iterated: once a model is evaluated, feedback mechanisms suggest design changes, including refactorings to improve quality or address security weaknesses without violating other constraints.

To support this vision, CPS patterns encapsulate quality semantics, providing guidance for their selection and interchange. Formalized annotations help partially automate this process and act as bridges between heterogeneous modeling notations.

C. ML-based Quality Prediction and Runtime Monitoring

This long-term objective aims to support runtime adaptation integrating machine learning (ML) techniques for predictive analysis and anomaly detection. ML models are trained on historical data and simulation results to detect deviations from expected behavior and recommend mitigations. This supports the creation of adaptive, self-aware systems that can maintain optimal quality tradeoffs even under dynamic conditions.

Embedding quality-aware ML into the CPS development loop moves tradeoff handling from manual to data-driven, context-aware optimization. This is particularly relevant for cybersecurity, where system threats evolve and traditional static protections may not be sufficient.

III. Methodology and Preliminary Results

The initial focus is on the first research objective — the formalization of a multi-formalism, quality-annotated pattern-based modeling approach for automatically deriving quality models from CPS models, as shown in Fig. 1. A twofold methodology was adopted that combines top-down abstraction with bottom-up empirical grounding.

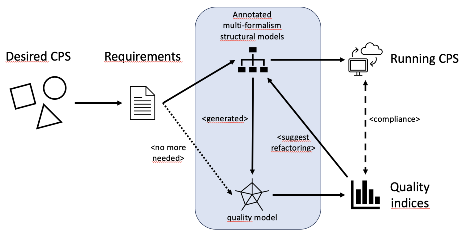

Fig. 1. Basic idea underlying the research project

A. Top-Down Exploration of Structural Modeling Patterns

Starting from a top-down perspective, existing literature and engineering practices was analyzed across both CPS and broader software-intensive systems to identify recurring architectural modeling patterns. This exploration was intentionally not limited to CPS-specific domains, allowing us to draw from a wider pattern taxonomy and adapt relevant structures to CPS modeling. This effort led to the identification of a catalog of generic modeling patterns, selecting a preliminary subset:

- ***Layered Pattern***: representing hierarchical system decompositions (e.g., sensing, control, actuation layers).
- ***Component-Based Pattern***: capturing modular system organizations with well-defined interfaces.
- ***Pipe-and-Filter, Event-Driven, and Publish-Subscribe***: useful for communication-intensive CPS applications.

These high-level abstractions can serve as structural blueprints for modeling CPS architectures and become the first candidates for formalization into reusable modeling blocks enriched with quality semantics.

B. Bottom-Up Derivation from CPS Case Studies

To complement the top-down identification of generic modeling patterns, a bottom-up investigation was conducted using concrete case studies and recent literature. This empirical analysis aimed to surface domain-specific architectural patterns that are observable in practical CPS deployments, and suitable for formalization and reuse within the presented pattern-based modeling framework. Two relevant studies were selected because capturing different yet complementary perspectives on CPS architecture modeling under uncertainty and dynamic behavior.

1) Adaptive Architectures in Dynamic Physical Environments: This first study [1] examines the architectural design of a robotic hospital logistics system where physical space is subject to dynamic changes — for instance, temporary unserviceable hallways or intermittently locked doors. This paper compares different communication patterns in terms of their ability to maintain system performance (e.g., response time) under varying configurations of the robots' coordination, namely: i) ***Centralized***; ii) ***Room-centric***; iii) ***Fully Decentralized***. This work not only demonstrates the need for architecture-level variability and adaptability in real-world CPS but also reinforces the importance of associating quantitative quality metrics with architectural structures.

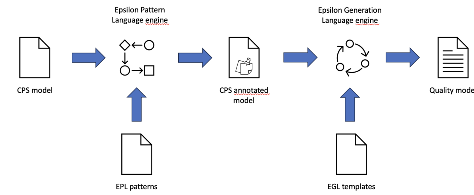

Fig. 2. Process pipeline to catch modeling patterns and generating quality models

2) Self-Adaptive CPS and Collective Intelligence Patterns: This second study [2] provides a survey-based synthesis of recurring architectural patterns used in the engineering of self-adaptive CPS and Cyber-Physical Production Systems (CPPS). These systems must operate under persistent uncertainty, scale, and heterogeneity, and often require runtime adaptation strategies spanning multiple layers of the architecture. The analysis led to the identification of a first set of three multi-layered patterns: ***SYNTHESIZE- UTILIZE***, ***SYNTHESIZE-COMMAND***, and ***COLLECT-ORGANIZE***.

C. Process Pipeline: From CPS Models to Quality Models

From previous analysis, a transformation pipeline was defined as a first concrete result, that enables the systematic progression from a high-level CPS model to formal quality models. The pipeline consists of three main phases, as shown in Fig. 2: i) **CPS Modeling** using native multi-formalism modeling approach to catch structural and behavioral aspects; ii) **CPS Model Annotation** automatically enriches the base model with quality properties annotations; iii) **Quality Model Generation** let the model (once annotated) undergo an automated transformation.

IV. Conclusion and Future Work

The presented process pipeline definition together with a small initial set of modeling patterns concur to reach the first research objective. Future work passes immediately through the formalization of quality properties annotations onto defined patterns in order to afford the second research objective regarding the automatic generation of quality models.

References

[1] R. Pinciroli and C. Trubiani, "Model-based performance analysis for architecting cyber-physical dynamic spaces," in *2021 IEEE 18th International Conference on Software Architecture (ICSA)*. IEEE, 2021, pp. 104–114.

[2] A. Musil, J. Musil, D. Weyns, T. Bures, H. Muccini, and M. Sharaf, "Patterns for self-adaptation in cyber-physical systems," *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, pp. 331–368, 2017.