CO-OPTIMIZATION OF CYBER-PHYSICAL SYSTEMS: LEVERAGING DOMAIN KNOWLEDGE IN MULTI-DOMAIN SYSTEM DEVELOPMENT PROCESSES

Yon Vanommeslaeghe

Cosys-Lab (FTI), University of Antwerp
AnSyMo/Cosys, Flanders Make vzw
Groenenborgerlaan 171, 2020 Antwerpen, BELGIUM
yon.vanommeslaeghe@uantwerpen.be

ABSTRACT

The term cyber-physical system (CPS) is generally used to describe a system that integrates software (cyber) and physical components, allowing this software to monitor and interact with the physical world through sensors and actuators. Due to the nature of these CPSs, i.e., the combination of the *cyber* and the *physical*, their design and development inherently involves engineers from multiple engineering domains. However, the complexity of CPSs is ever-increasing, driven by the demand for ever better performing, safer, more feature-rich, and more intelligent systems. In this regard, many of the advances in recent years have been achieved through software by using increasingly complex control and monitoring algorithms. However, as these systems become more complex, so does their design and development. Not only do engineers need to take into account more and more design parameters and objectives, and their interdependencies within their own domain, but the different engineering domains involved in the design also become more and more intertwined. While this introduces new challenges throughout the design process, in this dissertation, we focus on three specific stages: (i) the *detailed design* stage, (ii) the *system integration* stage, and (iii) the *system validation* stage. These three stages correspond to three distinct parts of the dissertation, as illustrated in Figure 1. The focus of each part and our contributions are described in detail in the following paragraphs.

First, the increasing number of dependencies within and across different engineering domains complicates the *detailed design* stage, as decisions made in one domain can impose additional restrictions and requirements on another. In such cases, the traditional approach, i.e., the subdivision of the design along the borders between the different engineering domains, becomes insufficient to find an optimal system design. It be-

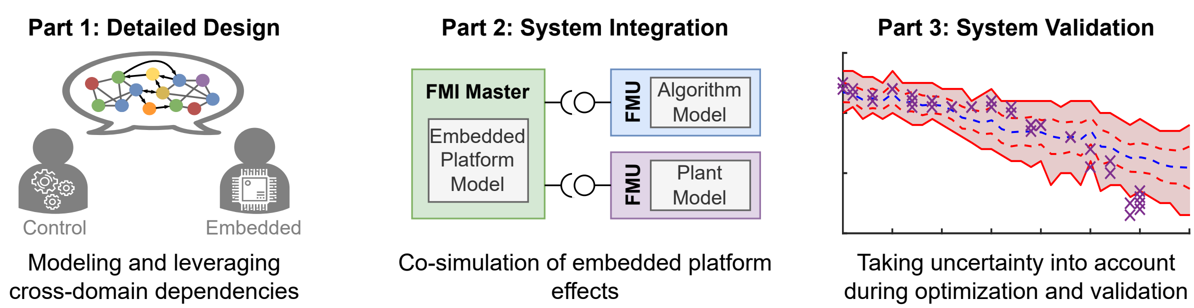

Figure 1: High-level overview of the dissertation.

comes necessary to regard this as a multi-domain co-design and co-optimization problem. However, in such an integrated approach, the design space expands dramatically. In this dissertation, we demonstrate how ontologies can be used to model the dependencies between design parameters and objectives, both within and across different engineering domains. Thus, explicitly capturing the (cross-)domain knowledge of the different engineers. We also present an approach to update this ontology in a structured way as new information, such as sensitivity information, becomes available during the development process. We show how this ontology allows engineers to reason about the design of the overall system and how it allows them to *manually* determine an efficient design space exploration (DSE) strategy. One which leverages these cross-domain dependencies to reduce the design space. We go on to present an approach to *automatically* determine efficient DSE workflows based on, among others, the information contained in the ontology.

Second, regarding the *system integration* stage, we explicitly consider the *embedded deployment* of control or monitoring algorithms. These algorithms are often designed and tested in a simulation environment, often ignoring certain limitations of embedded platforms, such as limited computational resources, limited numerical precision, non-ideal physical interfaces, etc. As such, embedded deployment of these algorithms can impact their functional behavior. This impact generally only becomes visible late in the integration stage, e.g., during hardware-in-the-loop (HiL) testing, and can thus lead to costly re-engineering if problems are found. In this dissertation, we demonstrate how co-simulation can be used to evaluate the impact of embedded platform effects on application behavior earlier in the development process. First, we demonstrate how an explicit model of the embedded platform interface allows the evaluation of its impact and the impact of related design parameters, e.g., ADC resolution. Next, we demonstrate how SimEvents® can be used to evaluate the impact of temporal behavior, i.e., non-negligible execution time, scheduler behavior, etc., on the application's functional behavior. Expanding on this, we present an approach to co-simulate embedded platform models, modeled using the discrete event system specification (DEVS) formalism, and application models, contained in functional mock-up units (FMUs), focusing on multi-core specific aspects, such as shared resources. We show how these techniques can be used to construct a virtual prototype, which can be tested in a virtual HiL or "vHiL" setup.

Finally, a common way of managing the complexity of CPS design and development is to employ model-based systems engineering (MBSE) techniques. Similarly, in the first two parts of the thesis, we extensively use modeling and simulation to solve various problems. However, models are always an abstraction of the real-world systems they represent, which introduces uncertainty at the model level. Similarly, variations in real-world system parameters can introduce additional uncertainty. This complicates *system validation*, as this uncertainty can cause discrepancies in system behavior between simulation and the real world. Indeed, suppose the system does not perform as expected. In that case, it may be challenging to determine if this is due to the uncertainty associated with the used models and parameters, or mistakes or oversights in the design and implementation of the system. In this thesis, we use a *descriptive* process model to describe our efforts to validate the real-world performance of a CPS co-optimized using a model-based design space exploration approach. We demonstrate how uncertainty affects the predicted system behavior, obtaining probability distributions rather than single values for the performance of different design candidates. We then compare the measured real-world performance of multiple design candidates to their predicted performance, demonstrating how the obtained probability distributions allow for a more objective assessment. Lastly, we discuss insights gained and lessons learned. Based on this, we present a possible *prescriptive* process model of the model calibration, design space exploration, and system validation process, which specifically takes into account uncertainty.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Paul De Meulenaere and Joachim Denil for convincing me to start a PhD and for their guidance, ideas, and insights along the way.