Framework for Inter-Model Analysis of Cyber-Physical Systems

Ivan Ruchkin, Dionisio De Niz, Sagar Chaki, David Garlan. Carnegie Mellon University, Pittsburgh, PA, USA.

Research Problem

CPS engineering combines diverse modeling methods to capture various aspects of the system, relying upon:

- Structurally and semantically diverse system *models*.
- *Analyses* — reasoning operations using models.

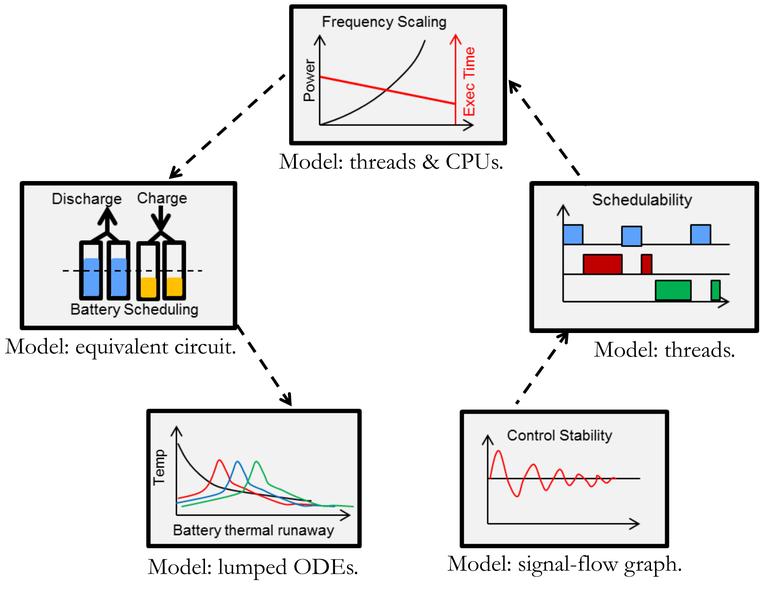
Improper combination of analyses from different models may lead to errors and, potentially, system failures.

Hence the research questions:

- How to detect inconsistencies between models?
- How to compose analyses correctly?

Example

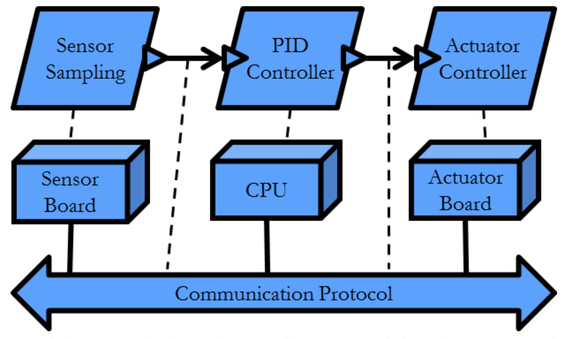
How can frequency scaling model determine the optimal CPU frequency, if it affects thread scheduling and battery scheduling, potentially invalidating those models?



Views for Models

First, we create a common representation of each model's information that may concern other model. We use *architectural views* — sets of components and connectors — to capture each model in architecture description languages: Acme and AADL.

In some cases, such as a thread timing model, an architectural view is created from scratch. In others, e.g., hybrid programs, we employ annotations to capture the model components, connectors, and their properties.



An architectural view for software and hardware models.

Analysis Contracts

Each analysis is assigned a *contract* — a set of inputs, outputs, assumptions, and guarantees. Inputs and outputs are specified in terms of the view analysis. Assumptions and guarantees are specified in FOL and LTL to verify correct analysis application.

For example, frequency scaling contract assumes DMS:

$I = \{threads, CPUs, CPUBind, Dline\}$, $O = \{CPUFreq\}$,

$A = \{ \forall t_1, t_2: threads \mid t_1 \neq t_2 \wedge CPUBind(t_1) = CPUBind(t_2): \mathbf{G}(CanPrmpt(t_1, t_2) \Rightarrow Dline(t_1) \leq Dline(t_2)) \}$, $G = \{\}$.

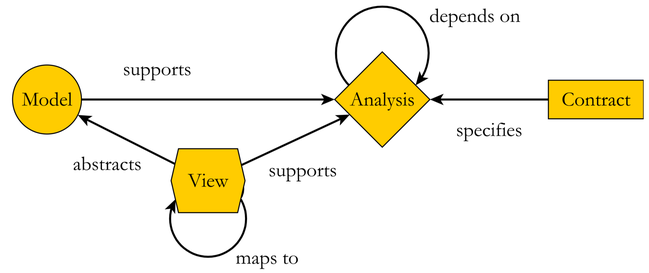
Inter-Model Analysis Framework

The framework allows engineers to use views and analysis contracts to:

- Verify model consistency through views.
- Verify correct analysis composition.

The first goal is achieved by specifying architectural constraints on views, appropriate to the context, and checking their satisfaction.

To achieve the second goal, the framework determines a sound ordering of analysis execution based on inputs and outputs. During the execution, the framework matches assumptions and guarantees with verification models to determine their satisfaction. Currently, the framework supports SMT solving of first order formulas and Spin verification of temporal contracts for thread scheduler and battery.



The metamodel of the framework.

References

- I. Ruchkin, D. De Niz, S. Chaki, and D. Garlan. *Contract-Based Integration of Cyber-Physical Analyses.* Appears in EMSOFT 2014.
- A. Rajhans, A. Bhave, I. Ruchkin, B. Krogh, D. Garlan, A. Platzer, and B. Schmerl. *Supporting Heterogeneity in Cyber-Physical Systems Architectures.* Appears in IEEE Transactions on Automatic Control.

Carnegie Mellon University — institute for SOFTWARE RESEARCH — Software Engineering Institute