Poster: Systematic Top-down Design of Cyber-physical Models with Integrated Validation and Formal Verification

Christoph Luckeneder
TU Wien, ICT
Vienna, Austria
christoph.luckeneder@tuwien.ac.at

Hermann Kaindl
TU Wien, ICT
Vienna, Austria
hermann.kaindl@tuwien.ac.at

ABSTRACT

The complexity of designing and verifying large-scale systems requires abstract models. Consistently and systematically deriving a more concrete model from an abstract model with regard to verification of its behavior against certain properties is an open problem. We propose a new workflow for systematic top-down design of models for a Cyber-physical System (CPS). It builds on a theory of systematic abstraction and refinement techniques in the context of *verification through model checking*. In addition, this workflow includes *validation* in the sense that a refined model is checked for its fit with reality. Our proposed workflow is new with respect to its systematic determination of model changes on different levels of abstraction based on the V&V results *and* the formal property *over-approximation* of an abstract model (as compared to the corresponding concrete model).

CCS CONCEPTS

• **Software and its engineering** → *Software design engineering*; *Formal software verification*;

KEYWORDS

Top-down design, behavioral models, CEGAR

ACM Reference Format:
Christoph Luckeneder and Hermann Kaindl. 2018. Poster: Systematic Top-down Design of Cyber-physical Models with Integrated Validation and Formal Verification. In *ICSE '18 Companion: 40th International Conference on Software Engineering Companion, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3183440.3194967

1 INTRODUCTION

Formal verification through model checking (see, e.g., [1]) intrinsically faces combinatorial explosion. This complexity was addressed in seminal work on Counterexample-guided Abstraction Refinement (CEGAR) [3], for instance, based on a theory of abstraction techniques [4]. In the bottom-up approach of CEGAR, these techniques provide a systematic way to (even automatically) generate an abstract model from a concrete one. The generated abstract model can then be used to verify the behavior of the given concrete model through model checking.

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3194967

Complexity is also a reason for *designing* in a top-down fashion, by starting with an abstract model and refining it. Verification in the course of such a design approach often means checking lower-level artefacts against higher-level ones. How is it about behavioral verification against properties through model checking in such a context? How to derive lower-level models systematically, so that properties are preserved? A naive approach (reminiscent of "naive abstraction" according to [3]) does not preserve such a property.

This paper proposes a new workflow for systematic top-down design of cyber-physical models with integrated validation and formal verification, whose results guide model changes on different levels of abstraction. While in CEGAR always abstract models are derived from concrete ones, our approach additionally defines mappings from abstract (qualitative) to concrete (quantitative) models. In contrast to CEGAR, our approach works top-down for designing a concrete model by starting from an abstract model.

The creation of this workflow happened first through inductive generalization from a case study starting with the qualitative *Adaptive Cruise Control* (ACC) model from [5], an abstract qualitative model of a cyber-physical system (CPS). From this workflow, a more elaborate one additionally using refinement techniques from [3] was derived, and applied for redoing the case study.

2 PROPOSED WORKFLOW

Now let us propose our new workflow for systematic top-down model design, as illustrated in Figure 1. It uses theory on over-approximation and refinement techniques from [3, 4] in a different context and in a new way.

First, an initial abstract model has to be provided (or used, when already given like the abstract qualitative model of a CPS proposed in [5]). This activity requires design skills and domain knowledge as well as a vision. Still, it should be easier to create an abstract model first than a concrete one, without having to take all the details into account upfront. Model-checking the abstract model may already at this stage reveal property violations in the form of counterexamples, and it is more efficient than model-checking a more concrete model. These counterexamples can be used as a source of information for modifying the abstract model with respect to the violated properties. Hence, the counterexamples have to be analyzed and the model modified, until model checking does not find any counterexample.

For a verified abstract model, a mapping to a concrete model is to be determined. The resulting concrete model should be immediately validated whether it is realistic or not. If the validation is not successful, the reasons must be analyzed and the mapping modified accordingly, until a concrete model is considered realistic.

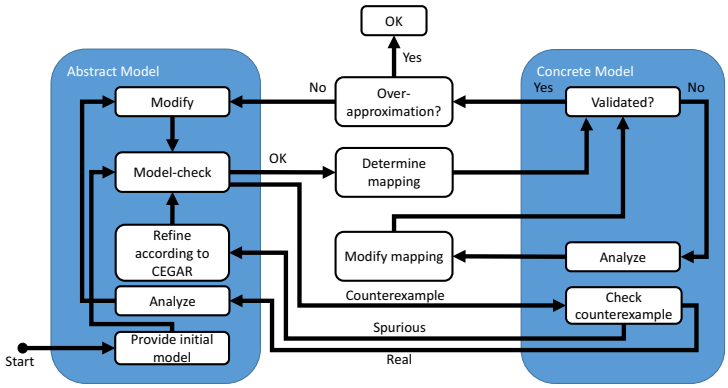

Figure 1: Proposed workflow for systematic top-down model design

For answering the question whether the concrete model fulfils the properties that the abstract model has already been verified against successfully, the abstract model needs to be an over-approximation of the concrete model. This must be checked, and such a check can be intricate when done manually. Even if this holds with respect to the originally mapped concrete model, the mapping may actually have changed due to changes of the concrete model regarding its validation. Hence, the over-approximation check must be performed bottom-up and separately. Especially after changes of the concrete model, this check is likely to fail.

In this case, the abstract model needs to be modified accordingly and model-checked again. If a counterexample is found, it is checked whether it is *spurious* (according to [3]). This is the case, if it is not a counterexample for the concrete model, i.e., it is an artefact of the abstraction. Hence, the abstract model is to be *refined*, and this can be done systematically according to [3, 4.4]. If the counterexample is real then the abstract model needs to be analyzed and modified accordingly, as well as model-checked again, etc.

Once the check for over-approximation is successful, the workflow can stop with the result that the concrete model is verified successfully against the same properties that hold for the abstract model.

3 RELATED WORK

Since model checking intrinsically faces combinatorial explosion, abstraction techniques were studied, e.g., in [3, 4], to reduce the state-space. These techniques provide a systematic way to generate an abstract model from a concrete one with one-sided error. In contrast to our approach, this work considers the concrete model as given and fixed. Hence, these techniques start from a given concrete model, while our workflow starts from an abstract model either given or defined in due course. Still, these abstraction techniques can be used in the course of enacting our workflow.

Clarke et al. [2] extended CEGAR for verification of *hybrid systems*. While we derived a quantitative model from a qualitative one in the course of our case study, it is still a finite system since we restrained it, e.g., to a finite number of distinct speed values. Hence, we did not have to use these extended techniques for hybrid systems. Still, applying them should be possible in the context of our workflow.

Software and CPS design in practice integrates verification as well, of course, both informally and formally. However, we are not aware of any previous approach that would systematically determine changes of models on different levels of abstraction based on verification results like our proposed workflow.

4 CONCLUSION

This proposed workflow provides a systematic way for top-down design of CPS models. Although it starts from the top, it has to iteratively work on different levels of abstraction, while keeping the models formally consistent. Like in CEGAR, the proof through model checking only needs to be done on the abstract level (with less complexity than the concrete model). In contrast to CEGAR, the purpose is not verifying an existing concrete model but designing one, including such formal verification.

In a case study using ACC (which we cannot include here due to lack of space), we built on an abstract qualitative model of a cyber-physical system proposed in [5]. Using our new workflow proposed here, we systematically derived a consistent quantitative model. This case study provided anecdotal evidence for its feasibility.

ACKNOWLEDGMENT

The FeatureOpt project (No. 849928), is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the program "ICT of the Future" between June 2015 and May 2018. More information can be found at https://iktderzukunft.at/en/.

REFERENCES

[1] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking.* MIT Press, Cambridge, MA, USA. I–XVII, 1–975 pages.
[2] Edmund Clarke, Ansgar Fehnker, Zhi Han, Bruce Krogh, Olaf Stursberg, and Michael Theobald. 2003. Verification of hybrid systems based on counterexample-guided abstraction refinement. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 192–207.
[3] Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. 2003. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM (JACM)* 50, 5 (2003), 752–794.
[4] Edmund M. Clarke, Orna Grumberg, and David E. Long. 1994. Model Checking and Abstraction. *ACM Trans. Program. Lang. Syst.* 16, 5 (Sept. 1994), 1512–1542. https://doi.org/10.1145/186025.186051
[5] Michael Rathmair, Christoph Luckeneder, and Hermann Kaindl. 2016. Minimalist Qualitative Models for Model Checking Cyber-physical Feature Coordination. In *Proceedings of the 23rd Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, USA, Article 1, 8 pages.