Joel Greenyer, Malte Lochau, and Thomas Vogel (Hrsg.): Explainable Software for Cyber-Physical Systems (ES4CPS), GI-Dagstuhl Seminar 19023, January 2019,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 11

Model-based Testing Strategies for Configurable Software with Unbounded Parametric Real-Time Constraints

Lars Luthmann[1], Malte Lochau[1]

Extended Abstract

Non-functional requirements like real-time behaviors are of ever-growing interest in many application domains of *cyber-physical systems (CPS)*. Additionally, we require CPS to be *configurable* such that variability of systems is taken into account. Consequently, existing modeling formalisms and analysis techniques for reasoning about time-critical behaviors have to be adapted to engineering of configurable CPS, too. Featured timed automata (FTA) [Co12] extend *timed automata (TA)* [AD90], a modeling formalism for discrete event/continuous time software systems, with annotation-based variability modeling [CA05] and (symbolic) family-based model-checking [Cl11] of configurable CPS. Here, we present configurable parametric timed automata (CoPTA) [Lu17] to further extend expressiveness of FTA by freely configurable and a-priori unbounded timing intervals of real-time constraints known from parametric timed automata (PTA) [He94]. Hence, CoPTA models impose infinite configuration spaces, making variant-by-variant analysis practically infeasible.

Instead, we present a family-based test-suite generation methodology for CoPTA models ensuring symbolical location coverage for every model configuration. Furthermore, we define a novel coverage criterion, called Minimum/Maximum Delay (M/MD) coverage [Lu18], requiring every location in a CoPTA model to be reached by test cases with minimum/maximum possible durations, for systematically investigating best-case/worst-case execution times. Consequently, we extend our family-based test-suite generation methodology to also achieve M/MD coverage on CoPTA models. Our evaluation results, obtained from applying our CoPTA tool to a collection of systems, reveal efficiency improvements of test-suite generation for configurable CPS, as compared to a variant-by-variant strategy in case of finite configuration spaces. Concerning the goals of the Dagstuhl Seminar on ES4CPS, we consider the following questions.

What is an ES4CPS problem, and/or what is an ES4CPS solution, that we are interested in? Based on our testing methodology for configurable real-time CPS, our techniques

[1] TU Darmstadt, Real-Time Systems Lab, Magdalenenstr. 4, 64289 Darmstadt, Germany, lars.luthmann@es.tu-darmstadt.de, malte.lochau@es.tu-darmstadt.de

may be used to systematically inspect configurable CPS to get a better understanding of the impact of parameters on execution times. For instance, we may explain best-case/worst-case execution times of CPS. Furthermore, one needs to first find a suitable model to express interesting aspects of explainable software to tackle these problems, whereupon a notion of testing may be defined.

What is the ES4CPS-related expertise that we can contribute (to solving this problem)? To solve this problem, we can contribute our above-mentioned expertise in generating test cases (and test suites) for models of explainable software. Especially, we may generalize the above-mentioned concept of M/MD coverage from CoPTA to hybrid automata such that we can analyze, e. g., neural networks.

What external expertise do we need (possibly from the other participants) in order to work on the problem/solution? Before being able to generate test cases, we require expertise in (1) finding (and defining) a suitable hybrid model for (families of) explainable software, and (2) in (black-box) testing of neural networks. Additionally, (3) as models of software are rarely available, they need to be generated which could, for instance, be achieved by applying a machine-learning approach.

References

[AD90] Alur, Rajeev; Dill, David: Automata for modeling real-time systems. In: ICALP'90, volume 443 of LNCS. Springer, pp. 322–335, 1990.

[CA05] Czarnecki, Krzysztof; Antkiewicz, Michał: Mapping Features to Models: A Template Approach Based on Superimposed Variants. In: GPCE'05. Springer, pp. 422–437, 2005.

[Cl11] Classen, Andreas; Heymans, Patrick; Schobbens, Pierre-Yves; Legay, Axel: Symbolic Model Checking of Software Product Lines. In: ICSE'11. ACM, pp. 321–330, 2011.

[Co12] Cordy, Maxime; Schobbens, Pierre-Yves; Heymans, Patrick; Legay, Axel: Behavioural Modelling and Verification of Real-time Software Product Lines. In: SPLC'12. ACM, pp. 66–75, 2012.

[He94] Henzinger, Thomas A.; Nicollin, Xavier; Sifakis, Joseph; Yovine, Sergio: Symbolic Model Checking for Real-Time Systems. Information and Computation, 111(2):193 – 244, 1994.

[Lu17] Luthmann, Lars; Stephan, Andreas; Bürdek, Johannes; Lochau, Malte: Modeling and Testing Product Lines with Unbounded Parametric Real-Time Constraints. In: SPLC'17. ACM, pp. 104–113, 2017.

[Lu18] Luthmann, Lars; Gerecht, Timo; Stephan, Andreas; Bürdek, Johannes; Lochau, Malte: Minimum/Maximum Delay Testing of Product Lines with Unbounded Parametric Real-Time Constraints. 2018. Submitted to Journal of Systems and Software.