

# Smart Home Model Verification with AnimUML (Poster)

Frédéric Jouault<sup>1</sup>, Ciprian Teodorov<sup>2</sup> and Matthias Brun<sup>1</sup>

<sup>1</sup>ESEO, Angers, France

<sup>2</sup>Lab-STICC, ENSTA Bretagne Brest, France

## Keywords

UML, model verification, smart home,

Model verification techniques, such as model checking, generally require relatively advanced expertise. They are therefore typically used in applications where their usefulness is especially appreciated, if not necessary, and can offset their costs. Critical system design has, for instance, been one of the main consumers of such techniques. They could however bring benefits to many other domains. Development times can be shortened by the drastically reduced number of mistakes in verified design models. Moreover, they can help reduce the number of bugs remaining in shipped products. Lowering barriers to entry for the application of these techniques should therefore have a significant impact. In this work, we show how model checking can be applied to UML models in the smart home context. The models were created with AnimUML, which makes them markedly easier to create than with traditional tools. Furthermore, this tool's direct model analysis support at the UML level makes it relatively simple to verify properties. It was able to detect several corner case issues, which would have been much harder to detect, and especially diagnose, with testing only.


Besides being time consuming, testing reaches its fundamental limits, checking *what the system should not do*. Besides, in the home automation context, the problem is even more complex due to the distributed nature of the problem [1]. The situation can certainly be improved by using formal verification approaches, like model-checking. These techniques, naturally geared towards distributed systems, allow the verification of properties expressing *what the system should do*, which naturally completes the correctness specification of a system. During the last decade, tremendous progress was achieved on this axis [2, 3], however most of the proposed approaches and tools require a high-degree of sophistication from the home automation designer. Moreover, the marketing target of home automation solutions, like Google Smart Home<sup>1</sup>, is wide and targets non-expert users. Nevertheless, the modeling community started a push towards lively verification environments [4], which enables seamless user interaction during the design and debugging process. The AnimUML environment [5, 6, 7] pushes the frontiers of this approach by allowing not only early debugging of high-level specifications, but also


---

STAF 2022 Workshop MESS'22: International workshop on MDE for Smart IoT Systems, July 5, 2022, Nantes, France

✉ frederic.jouault@eseo.fr (F. Jouault); ciprian.teodorov@ensta-bretagne.fr (C. Teodorov); matthias.brun@eseo.fr (M. Brun)

ORCID 0000-0002-2395-9623 (F. Jouault); 0000-0002-0722-5857 (C. Teodorov)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

<sup>1</sup><https://developers.google.com/assistant/smarthome/overview>

formal verification of partial (under development) specifications. This work presents a case study<sup>2</sup> where a LYWSD03MMC<sup>3</sup> temperature and humidity sensor is integrated with the Google Smart Home automation platform using local fulfillment<sup>4</sup>. We leverage the capabilities offered by AnimUML both to better understand the overall architecture, through lively user-model interactions, and to allow for the verification of non-trivial properties. The two main limitations of the approach are the following. 1) Although the tooling significantly helps, it is still necessary to learn to use AnimUML. 2) In addition to modeling the app, the designer must also model the behavior of the Smart Home API, and of the device. Regarding the first issue, we plan to keep improving the tool to make it easier to learn and use. As for the second issue, ideally manufacturers should provide such models. Moreover, we already provide a reusable Google Smart Home API model as part of our case study.

## References

- [1] A. Demeure, S. Caffiau, E. Elias, C. Roux, Building and Using Home Automation Systems: A Field Study, in: ISEUD 2015, Madrid, Spain, 2015. doi:10.1007/978-3-319-18425-8\_9.
- [2] A. Souri, M. Norouzi, A state-of-the-art survey on formal verification of the internet of things applications, *Journal of Service Science Research* 11 (2019) 47–67. doi:10.1007/s12927-019-0003-8.
- [3] C.-J. M. Liang, L. Bu, Z. Li, J. Zhang, S. Han, B. F. Karlsson, D. Zhang, F. Zhao, Systematically debugging iot control system correctness for building automation, in: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, BuildSys '16, Association for Computing Machinery, 2016, p. 133–142. doi:10.1145/2993422.2993426.
- [4] D. Ingalls, The lively kernel: Just for fun, let's take javascript seriously, in: Proceedings of the 2008 Symposium on Dynamic Languages, DLS '08, Association for Computing Machinery, 2008. doi:10.1145/1408681.1408690.
- [5] F. Jouault, V. Besnard, T. L. Calvar, C. Teodorov, M. Brun, J. Delatour, Designing, animating, and verifying partial uml models, in: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS '20, Association for Computing Machinery, 2020, p. 211–217. doi:10.1145/3365438.3410967.
- [6] F. Jouault, V. Seville, V. Besnard, T. L. Calvar, C. Teodorov, M. Brun, J. Delatour, Animuml as a uml modeling and verification teaching tool, in: 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 2021, pp. 615–619. doi:10.1109/MODELS-C53483.2021.00094.
- [7] M. Pasquier, F. Jouault, M. Brun, J. Pérochon, Evaluating tool support for embedded operating system security: An experience feedback, in: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Association for Computing Machinery, 2020. doi:10.1145/3417990.3420048.

---

<sup>2</sup>Case study material is available on GitHub at <https://github.com/fjouault/SmartHomeCaseStudy>.

<sup>3</sup>[https://esphome.io/components/sensor/xiaomi\\_ble.html#lywsd03mmc](https://esphome.io/components/sensor/xiaomi_ble.html#lywsd03mmc)

<sup>4</sup><https://developers.google.com/assistant/smarthome/concepts/local>