ReqV: A Tool for Requirements Formal Consistency Checking

Massimo Narizzano[1], Luca Pulina[2], Armando Tacchella[1], and Simone Vuotto[1,2]

[1] DIBRIS, University of Genoa, Viale Causa 13, 16145 Genoa, Italy
massimo.narizzano@unige.it, armando.tacchella@unige.it
[2] Chemistry and Pharmacy Dept., University of Sassari, Via Muroni 23A, Sassari, Italy
lpulina@uniss.it, svuotto@uniss.it

Summary

In the context of safety- and security-critical Cyber-Physical Systems (CPSs), checking the sanity of functional requirements is an important, yet challenging task. It is largely recognized that a flaw in the requirements specification can lead to delays, additional expenses and, possibly, the failure of the project. Nonetheless, due to the intrinsic difficulty of dealing with natural language sentences, requirements are often checked manually, an error-prone and time-consuming activity. Given the increasing demand and complexity of CPSs, and the need to reduce time-to-market and costs, practical solutions to enable automated verification of requirements are in order. Formal methods provides a viable solution, but they often requires overburdening formalization and a high degree of expertise. As a trade-off between formalization and usability, a recurrent solution in the literature is the use of Property Specification Patterns (PSPs), English-like structured natural sentences that provides a direct mapping to one or more logics.

In this paper, we present REQV, a tool developed in the context of the H2020 EU CERBERO [1] project[3] that leverage on PSPs to tackle the problem stated before. In [2], we extended PSPs by considering Boolean as well as atomic numerical assertions of the form $x =^* c$, where X is a variable of the system, $c \in \mathbb{R}$ is a constant real number and the operator $=^* \in \{<, <=, =, >=, >\}$ have the usual interpretation. Furthemore, we presented an encoding to reduce the *inner consistency* of extended PSPs, *i.e.*, logical errors in the specification that prevents any possible system to satisfy all requirements, to the Linear Temporal Logic (LTL) [3] satisfiability problem. We also extended the previous work with a new algorithm to find a minimum set of conflicting requirements in case of inconsistency, and we collected all these functionalities in a Java library called SpecPro[4].

REQV exploits these capabilities to provide an easy-to-use interface for the verification of requirements. Its goal is to enable users with no background knowledge of formal methods and logical languages to write requirements in PSP forms and check their consistency. REQV also aims at minimizing the setup process for the user, and therefore it is developed as a web application that can easily be accessed with a browser.

REQV is available for download at `https://gitlab.sagelab.it/sage/ReqV` and it can be tested at `https://reqv.sagelab.it`. A video tutorial is also available at `http://www.cluster-prossimo.it/docs/ReqV_video.mp4`.

Acknowledgnemnts The research of Luca Pulina and Simone Vuotto has been funded by the EU Commission's H2020 Programme under grant agreement N.732105 (CERBERO). The research of Luca Pulina has been also partially funded by the Sardinian Regional Project PROSSIMO (POR FESR 2014/20-ASSE I) and the FitOptiVis (ID: 783162) project.

References

1. Masin, M., Palumbo, F., Myrhaug, H., de Oliveira Filho, J., Pastena, M., Pelcat, M., Raffo, L., Regazzoni, F., Sanchez, A., Toffetti, A., et al.: Cross-layer design of reconfigurable cyber-physical systems. In: Proceedings of the Conference on Design, Automation & Test in Europe. pp. 740–745. European Design and Automation Association (2017)
2. Narizzano, M., Pulina, L., Tacchella, A., Vuotto, S.: Consistency of property specification patterns with boolean and constrained numerical signals. In: NASA Formal Methods: 10th International Symposium, NFM 2018, Newport News, VA, USA, April 17-19, 2018, Proceedings. vol. 10811, pp. 383–398. Springer (2018)
3. Pnueli, A., Manna, Z.: The temporal logic of reactive and concurrent systems. Springer 16, 12 (1992)

[3] `http://www.cerbero-h2020.eu/`
[4] `https://gitlab.sagelab.it/sage/SpecPro`