

RoQME: Dealing with Non-Functional Properties through Global Robot QoS Metrics

C. Vicente-Chicote¹, J. Berrocal¹, J. García-Alonso¹, J. Hernández¹, A. J. Bandera²,
J. Martínez², A. Romero-Garcés², R. Font³ and J.F. Inglés-Romero³

¹ Universidad de Extremadura, Quercus Software Engineering Group, EPCC, Spain

² Universidad de Málaga, E.T.S.I. de Telecomunicación, Spain

³ Biometric Vox S.L., Spain

cristinav@unex.es

Abstract. Non-functional properties are an essential part of any software solution. There is a lot of literature on what non-functional properties are but, unfortunately, there is also a lot of disagreement and different points of view on how to deal with them. Non-functional properties, such as safety or dependability, become particularly relevant in the context of robotics. In the EU H2020 RobMoSys Project, non-functional properties are treated as first-class citizens and considered key added-value services. In this vein, the RoQME Integrated Technical Project, funded by RobMoSys, aims at contributing a model-driven tool-chain for dealing with system-level non-functional properties, enabling the specification of global Robot Quality of Service (QoS) metrics. The estimation of these metrics at runtime, in terms of the contextual information available, can then be used for different purposes, such as robot behavior adaptation or benchmarking.

Keywords: Robotics, Non-Functional Properties, QoS metrics, MDE.

1 Context and intended goals

Component-Based Software Development (CBSD) aims at promoting software reuse for significantly reducing development time and cost. Existing solutions are encapsulated in well-defined components with clear (required and provided) interfaces that enable their connection to and interoperation with other components. Building systems out of components requires taking into account both functional and non-functional properties. Non-functional properties define “how” a system performs rather than “what” it does [1]. Examples of non-functional properties include timing, dependability, safety or resource consumption, among others. Despite the importance of non-functional properties, there are just a few component models explicitly supporting their specification and management throughout the development process. In most cases, this support is limited and, unlike the well-established solution of embodying functional properties into interfaces, no consensus has emerged on how to handle non-functional properties both at a component and at a system level [1].

RobMoSys: Composable Models and Software for Robotics [2] is a 4-year Project (2017-2020), funded by the EU H2020 Research and Innovation Program under grant agreement No. 732410. The vision of RobMoSys is to create better models, as the basis for better tools and better software, which then allow building better robotic systems. RobMoSys aims at creating an open, sustainable, agile and multi-domain European robotics software ecosystem. RobMoSys seeks to enable the composition of robotics applications with managed, assured, and maintained system-level properties using Model-Driven Engineering (MDE) techniques from a CBSD perspective. To achieve its goals, RobMoSys establishes structures that enable the management of the interfaces between different robotics-related domains, different roles in the ecosystem, and different levels of abstraction. RobMoSys financially supports, through a cascade funding scheme scheduled in two open calls, third party contributions as means to achieve its own objectives. The RoQME Project has been one of the six selected Integrated Technical Project (ITP) to be funded in the context of the first RobMoSys open call (out of the thirty four proposals submitted).

The main intended goal of RoQME is to provide robotics software engineers with a model-driven tool-chain allowing them to: (1) specify relevant system-level non-functional properties in terms of the (internal and external) contextual information available at runtime; and (2) generate RobMoSys-compliant components, ready to provide other components with QoS metrics defined on the non-functional properties, previously specified. Fig. 1[a] illustrates the kind of QoS metrics RoQME plans to target.

RoQME will run for one year, starting March 2018. Achieving substantial results in such a short period of time requires building on previous results. In this vein, the RoQME partners contribute solid background both in robotics and in model-driven software engineering. Apart from the fruitful relationship existing among the RoQME partners, which already resulted in some preliminary results [3][4], it is worth mentioning our close collaboration with some of the RobMoSys partners in line with the current goals of the Project [5-8], guaranties the alignment between RoQME and RobMoSys.

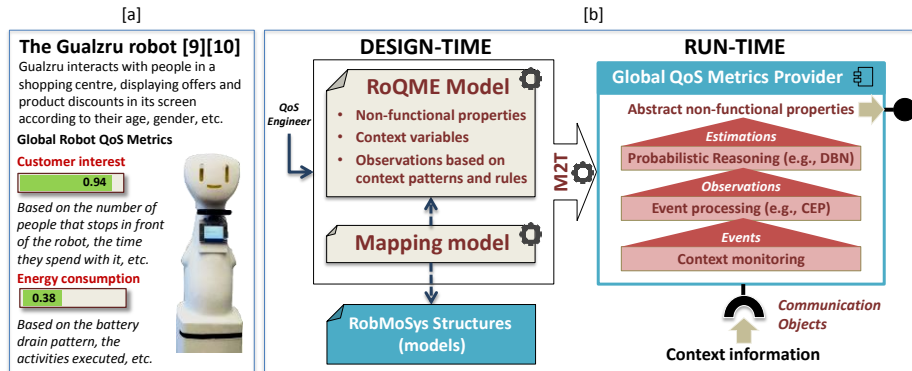


Fig. 1. [a] Example QoS metrics RoQME plans to target. [b] RoQME framework. *Left:* Tool-chain supporting the *QoS Engineer* role at design time. *Right:* Generated RobMoSys-compliant component ready to be used as a QoS metrics provider.

2 Main expected contributions

RoQME intends to support the role of *QoS Engineers*, providing them with a specific *QoS View* that allows them to model system-level non-functional properties according to the *RoQME meta-model*. This new role, view and meta-model complement and interrelate with those already defined in RobMoSys through the so called *RoQME-to-RobMoSys mapping meta-model*. This mapping aims at promoting good design principles, such as high cohesion and loose coupling among the different RobMoSys views, providing a non-intrusive way of extending the RobMoSys meta-model, i.e., modifying the RoQME meta-model would only imply adapting the mapping but not the RobMoSys meta-model, and *vice versa*.

RoQME will allow QoS Engineers to model *context variables* (e.g., battery level) and, from them, relevant *context patterns* (e.g., “the battery level drops more than 1% per minute”). The detection of a context pattern will be considered an *observation* associated with a variable in a *belief network*. Belief networks will be used to specify the dynamics of non-functional properties (e.g., power consumption). The degree of fulfillment of these non-functional properties will then be used to estimate the *QoS metrics*, obtained as real values in the range [0, 1].

RoQME aims to be application domain agnostic, providing QoS Engineers with a compact set of modeling tools to express system-level QoS metrics. However, it is being designed to be as flexible as possible, e.g., supporting extension mechanisms that allow QoS Engineers (or other RobMoSys roles, such as Safety Engineers or Performance Designers) to enrich and customize the RoQME modeling capabilities with domain-specific requirements (e.g., related to safety, dependability, etc.).

The *RoQME tool-chain*, delivered as an Eclipse plug-in, will provide both modeling and code generation tools, enabling the creation of RobMoSys-compliant components, readily usable in RobMoSys-based solutions as QoS information providers (see Fig. 1[b]). This information could then be used by other components for different purposes, e.g., robot behavior adaptation or benchmarking.

Internally, the generated components will estimate the value of each QoS metric, specified in the RoQME model, by successively processing the available contextual information, either from internal (e.g., robot sensors) or external (e.g., web services, other robots, etc.) sources. The contextual information received by the component will be sequentially processed by three modules (see Fig. 1[b]): (1) a context monitor that will receive raw contextual data and will produce context events (e.g., changes in the battery level); (2) an event processor that will search for the event patterns specified in the RoQME model and, when found, will produce observations (e.g., battery is draining too fast); and, finally (3) a probabilistic reasoner that will compute a numeric estimation for each metric (i.e., the degree of fulfillment of each non-functional property).

3 Additional remarks

This paper aims at introducing the RoQME Project to the JISBD community and the Spanish Network of Excellence in Model-Driven Software Engineering, funded by the

Spanish Ministry of Economy, Industry and Competitiveness (TIN2016-81836-REDT). Although RoQME is still in its initial stage, we considered it important to share it with the MDE community, since it supports the fact that there are application domains with relevant open challenges to which MDE can (and is expected to) contribute a lot.

It is worth mentioning that all the RoQME partners are pledged to making the knowledge generated in the course of the Project as widely and freely available as possible for subsequent research and development. For this reason, the Project partners are fully committed to open-access and open-source. In this vein, all the Project results will be punctually announced through the Project social networks (Twitter: @RoQME_ITP, LinkedIn: RoQME Group, or ResearchGate: RoQME Project), and made publically available through the RobMoSys website [2].

Acknowledgment

The RoQME Integrated Technical Project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 732410, in the form of financial support to third parties of the RobMoSys Project.

References

1. Sentilles, S.: Managing extra-functional properties in component-based development of embedded systems. PhD Thesis, Mälardalen University (2012). ISBN: 978-91-7485-067-3.
2. RobMoSys website, <http://robmosys.eu>, last accessed 2018/05/04.
3. RobMoSys technical wiki, <http://robmosys.eu/wiki/>, last accessed 2018/05/04.
4. Romero-Garcés, A., et al.: Testing a fully autonomous robotic salesman in real scenarios. IEEE Int'l Conf. on Autonomous Robot Systems and Competitions (ICARSC), 2015. DOI: 10.1109/ICARSC.2015.19
5. Ingles-Romero, J.F., Romero, A., Vicente-Chicote, C., Martínez, J.: A Model-Driven Approach to Enable Adaptive QoS in DDS-Based Middleware. IEEE TETCI, 1(3): 176 - 187 (2017). DOI: 10.1109/TETCI.2017.2669187.
6. Schlegel, C., Lotz, A., Lutz, M., Stampfer, D. Inglés-Romero, J.F., Vicente-Chicote, C.: Model-driven software systems engineering in robotics: covering the complete life-cycle of a robot. *it - Information Technology*, 57(2): 85-98 (2015). DOI: 10.1515/itit-2014-1069.
7. Lotz, A., Inglés-Romero, J.F., Stampfer, D. Lutz, M., Vicente-Chicote, C., Schlegel, C.: Towards a Stepwise Variability Management Process for Complex Systems: A Robotics Perspective. *IJISMD*, 5(3): 55-74 (2014). DOI: 10.4018/ijismd.2014070103.
8. Inglés-Romero, J.F., Lotz, A., Vicente-Chicote, C., Schlegel, C.: Managing Run-Time Variability in Robotics Software by Modeling Functional and Non-functional Behavior. *LNBIP*, 147: 441-455 (2013). DOI: 10.1007/978-3-642-38484-4_31.
9. Inglés-Romero, J.F., Lotz, A., Vicente-Chicote, C., Schlegel, C.: Dealing with Run-Time Variability in Service Robotics: Towards a DSL for Non-Functional Properties. *DSLRob-2012*. arXiv:1303.4296.
10. Fernández, F., et al.: Gualzru's Path to the Advertisement World. *IROS Workshop: Fine-R, Hamburg (Germany)*, pp. 55–65, 2015.
11. Romero, A., Ingles-Romero, J.F., Martínez, J., Vicente-Chicote, C.: Self-adaptive quality-of-service in distributed middleware for robotics. In *Proc. 2nd Int'l Workshop on Recognition and Action for Scene Understanding (REACTS 2013)*.