A Framework for Engineering Self-Adaptive Cyber-Physical Systems

Master's Thesis

Supervisor: Prof. Dr. Alexander Pretschner
Advisor: Ana Petrovska
Email: {alexander.pretschner, ana.petrovska}@tum.de
Phone: +49 (89) 289 - 17830

Starting date: 15.02.2021

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17830
https://www4.in.tum.de

Context

In order to design cyber-physical systems (CPSs) capable of performing a variety of tasks involving constant interactions with the real world, we have to take into account the uncertainties they are subjected to. The changing nature of the physical world means that they have to reliably collect data, extract information from it and ideally make the best possible decision. From the set of changing environmental (contextual) and internal circumstances arises the need for self-adaptivity in order for the CPSs to accomplish operational robustness and preserve and improve certain quality objectives.

Across the literature, multiple frameworks and models have been proposed for a descriptive representation and classification of software and systems architectures. The most prominent ones are: the 4+1 architectural view model [1], the SPES XT modeling framework [2], and Zachman's framework for information systems architectures [3, 4]. On a general level, the architecture of any system—including self-adaptive systems—can be viewed from three levels of abstraction: the conceptual, logical, and physical levels. As we move towards the lower levels of abstraction, the granularity of the system's architecture is increased.

At the *conceptual level*, self-adaptivity can be modeled using the MAPE-K (Monitor, Analysis, Plan and Execute, with Knowledge) [5] feedback loop. However, the conceptual MAPE-K model is not informative nor descriptive when it comes to engineering self-adaptive systems, which even lack commonly accepted definitions in the literature [6].

Within the frame of our prior efforts to define(self-)adaptivity, we have also proposed a *logical architecture* [7, 8] for engineering self-adaptive CPS, as well as a model problem, from a target class of systems, from the multi-agent robotics domain. Additionally, based on the model problem we have built two ROS-based simulated systems. The first system uses a realistic Gazebo simulator, whereas, for the second system, we have implemented a custom simulator [9] in which we omitted the uncertainties from the AMCL and the move-base modules that are by default non-deterministic in the realistic simulator. For completeness, the ROS-graphs from both systems represent their *physical architecture.*

The existing implementation of the custom simulator system cannot be used as a more general framework for engineering self-adaptive CPSs. Therefore, based on the current physical implementation of the system and guided by the logical architecture, we want to propose a framework for engineering self-adaptive CPSs that generalizes and enables more flexible and efficient extensions for building and experimenting with self-adaptive CPSs.

Goal

The main goal of this thesis is to propose a framework for engineering self-adaptive CPS, which generalizes and uses the current implementation of the system as a basis, in which the design decisions will be steered by the previously proposed logical architecture. The framework we aim for can be seen as an intermediate step between the logical architecture and the physical implementation, allowing for reconciliation between the two. While being functionally adequate, the current implementation does not extend from the logical architecture. Furthermore, the framework should respect the modular structure of the current ROS implementation while 1) addressing some the already identified problems with the current implementation of the system and 2) improving code quality, readability, and maintainability. Concretely, the current implementation is not modular, although it uses the modularity of ROS. For instance, to add new functional elements to the system, eg., a new planning algorithm, changes in multiple locations in the source code become compulsory because of code and file dependencies. This significantly increases the time costs for adding new functionality.

Working plan

1. Read and understand previous literature on self-adaptive systems, defining self-adaptive systems and the proposed logical architecture for engineering self-adaptive CPS

2. Research and compare previously proposed frameworks in the literature for engineering-self adaptive systems
3. Understand the implementation of the custom simulator-based system
4. Conceptual development of an applicable framework
    - understand how frameworks for software systems are developed in general
    - map the current system's implementation to the logical architecture and identify inconsistencies and gaps
    - finalize the initial concept of the framework that generalizes the implemented system and is consistent with the logical architecture
5. Realization of the framework
    - address the already identified problems with the current implementation of the robotics system, for example, 1) dynamic, run-time addition of more than two robots in the room, 2) fixing the hard-coded issues with the planning policy, and 3) adding multiple planning algorithms
    - re-structure and generalize certain modules, or introduce new modules in the current implementation to align it with the logical architecture
    - elicit, specify and document the interfaces between the different modules
6. Evaluate the framework by the means of
    - extensibility, ie., adding new functional elements
    - comparison with the other existing frameworks and architectures for engineering self-adaptive systems
7. Write the final thesis report

Deliverables

- Source code of the implementation
- Technical report with comprehensive documentation of the implementation, i.e. design decision, architecture description, API description and usage instructions. Usually as part of the GitLab documentation
- Final thesis report written in conformance with TUM guidelines

Pre-requisite

- Good Python and C/C++ skills
- Ideally, previous knowledge and experience with ROS

References

[1] P. B. Kruchten, "The 4+ 1 view model of architecture," *IEEE software*, vol. 12, no. 6, pp. 42–50, 1995.

[2] K. Pohl, M. Broy, H. Daembkes, and H. Hönninger, "Advanced model-based engineering of embedded systems," in *Advanced Model-Based Engineering of Embedded Systems*, pp. 3–9, Springer, 2016.

[3] J. A. Zachman, "A framework for information systems architecture," *IBM systems journal*, vol. 26, no. 3, pp. 276–292, 1987.

[4] J. A. Zachman, "The zachman framework for enterprise architecture," *Primer for Enterprise Engineering and Manufacturing.[si]: Zachman International*, 2003.

[5] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[6] D. Weyns, "Software engineering of self-adaptive systems," in *Handbook of Software Engineering*, pp. 399–443, Springer, 2019.

[7] A. Petrovska, "Logical Architecture for Engineering (Smart) Self-Adaptive Cyber-Physical Systems," *International Conference on Software Architecture (ICSA)*, 2021. under review.

[8] T. Beffart, "Logical Architecture for Engineering Smart Self-Adaptive Cyber-Physical Systems," Guided Research in Informatics, Chair of Software and System Engineering, Department of Informatics, Technical University Munich, 2020.

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17830
https://www4.in.tum.de

[9] J. Weick, “Realization of Adaptive System Transitions for Smart Se;f-Adaptive Cyber-Physical Systems using a Modular Approach with Learned Parameters on an Example of Multi-Robot Collaboration,” Master Thesis in Informatics, Chair of Software and System Engineering, Department of Informatics, Technical University Munich, 2020.

Fakultät für Informatik
Lehrstuhl 4
Software & Systems Engineering
Prof. Dr. Alexander Pretschner

Boltzmannstraße 3
85748 Garching bei München

Tel: +49 (89) 289 - 17830
https://www4.in.tum.de