Evaluation Framework for a Self-Adaptive Cyber-Physical System on an Example of a Multi-Robot System

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: 15.09.2020

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

Context

Modern systems such as CPS are composed of many interacting and interconnected components while inheriting all the complexities of modern large-scale distributed systems. Additionally, their close connection to the physical world means that they are exposed to high uncertainty during their operation. Namely, modern CPSs need to be able to operate efficiently and reliably within a continually changing, uncertain, and unanticipated environment (execution context). Furthermore, they often cooperate and collaborate with other CPSs to provide a shared functionality. A common approach to deal with run-time changes and uncertainties is to make the CPSs self-adaptive.

In the community of (self-)adaptive systems there is no broadly accepted reference system, although there exist a few exemplars varying from quite generic such as artifacts, to rather specific model problems. The exemplars have been derived and focus on different domains, primarily software systems [1, 2, 3, 4], cloud systems [5], databases [6], and IoT [7, 8]. With an exception of exemplars for UAVs [9, 10], previously there has not been proposed any other exemplar from the domain of CPSs, concretely from the domain of multi-agent autonomous mobile robots.

This thesis focuses on a previously created reference problem on Self-Adaptive Cyber-Physical Systems (SACPS) from the robotics domain, and the previous implementation of the model problem in two ROS-based in-house built simulators: 1) realistic simulation with Gazebo, and 2) custom simulation in which are omitted the uncertainties from the AMCL and the move-base modules.

The reference problem is comprised of several cleaning robots operating in the same context, e.g. a room. Each robot is able to autonomously move to a destination while avoiding 1) static obstacles and 2) dynamic obstacles (e.g., other robots, humans) along its way. New dirt tasks continuously appear in the room, with unknown spatial and temporal patterns, and the robots need to discover them in a distributed manner.

Goal

The different simulators generate setups differently. Concretely, different file formats are used as input: 1) regarding the maps of the rooms; and 2) regarding the task (dirt) generation and distribution in the room. Furthermore, evaluating behavior of any system, especially self-adaptive systems that include processes like analyzing, planning and learning is complex in different ways. Namely, the evaluation metrics are usually application specific and use-case dependent, and very rarely they are formed based on consensus of the respective scientific community, if such consensus exist. For our newly created reference problem—described above—we have not established a consensus regarding the evaluation metrics yet.

The main goal of this master thesis is building an evaluation framework and standardization of metrics for the evaluation of the self-adaptive cyber-physical systems on the previously given example of a multi-robot system. We structured the contribution of the master thesis in two phases:

Phase 1: (Automated) Setup generation for both simulators

In this phase we are interested in standardizing the setup generation so it can be used by both simulators interchangeably. Additionally, we are interested in examining which room layouts make sense, including the positioning and amount of tasks and hot-spots.

Phase 2: Evaluation metrics and standardization for the evaluation

In this phase we are interested in setting up evaluation metrics for the reference problem. It is of a crucial importance to notice that time is not universal in different simulators, and the time-dependent performance measures need to take that into consideration.

Working Plan

1. Phase 1: Setup generation for both simulators

(a) Investigate in the previous built systems what map/environment input formats are supported by both of the simulators
(b) “Standardize” meaningful room/map layouts
(c) Investigate how the task generation/spawning is being handled by both of the simulators: frequency-wise and spatial-wise.
(d) “Standardize” patterns/frequencies of dirt spawning
(e) Investigate which outputs already exist in both simulators and could be taken for possible performance metrics
(f) (optional, if the setup generation is automated) Investigate different room/space/map generation from game theory
(g) (optional, if the setup generation is automated) Procedural content generation

2. Phase 2: Evaluation metrics and standardization for the evaluation
   (a) Collect all the evaluation metrics from previous theses
   (b) Research on related work on standardizing metrics
      i. In self-adaptive systems domain
      ii. In robotics domain
      iii. Other relevant literature
   (c) Finalizing the evaluation metrics for our use case, based on the two steps above
3. Normalization of the metrics considering the map layout, positioning and amount of tasks and hot-spots, frequency of spawning tasks amount of robots
4. Realizing the evaluation framework, and accordingly altering the existing simulated systems, based on the results from Phase 1 and Phase 2
5. Data collection and test automation

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

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] S.-W. Cheng and B. Schmerl, *Znn.com*, (accessed November 3, 2019).

[2] M. U. I. Danny Weyns, Radu Calinescu and Y. Ruan., *Tele Assistance System (TAS)*, (accessed November 3, 2019).

[3] C. P. Sanny Schmid, Ilias Gerostathopoulos and T. Bures., *Model Problem (CrowdNav) and Framework (RTX) for Self-Adaptation Based on Big Data Analytics*, (accessed November 3, 2019).

[4] B. S. Gabriel A. Moreno and D. Garlan., *SWIM: An Exemplar for Evaluation and Comparison of Self-Adaptation Approaches for Web Applications*, (accessed November 3, 2019).

[5] M. L. Cornel Barna, Hamoun Ghanbari and M. Shtern., *Hogna: A Platform for Self-Adaptive Applications in Cloud Environments*, (accessed November 3, 2019).

[6] A. R. D. V. L. Wito Delnat, Eddy Truyen and W. Joosen, *K8-Scalar: a workbench to compare autoscalers for container-orchestrated database clusters*, (accessed November 3, 2019).

[7] P. B. D. W. M. Usman Iftikhar, Gowri Sankar Ramachandran and D. Hughes, *DeltaIoT: A Real World Exemplar for Self-Adaptive Internet of Things*, (accessed November 3, 2019).

[8] M. Provoost and D. Weyns., *DingNet: A Self-Adaptive Internet-of-Things Exemplar*, (accessed November 3, 2019).

[9] G. Moreno, C. Kinneer, A. Pandey, and D. Garlan, “Dartsim: an exemplar for evaluation and comparison of self-adaptation approaches for smart cyber-physical systems,” in *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 181–187, IEEE, 2019.

[10] P. H. Maia, L. Vieira, M. Chagas, Y. Yu, A. Zisman, and B. Nuseibeh, “Dragonfly: a tool for simulating self-adaptive drone behaviours,” in *2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pp. 107–113, IEEE, 2019.

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