

# Logical Architecture for Engineering Smart Self-Adaptive Cyber-Physical Systems

Guided Research



**Supervisor:** Prof. Dr. Alexander Pretschner

**Advisor:** Ana Petrovska

**Email:** {alexander.pretschner, ana.petrovska}@tum.de

**Phone:** +49 (89) 289 - 17830

**Starting date:** October 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>

## Context

Self-adaptive systems are able to autonomously and independently modify themselves to successfully cope with the changes and the uncertainties, without external human intervention or administration during run time. The changes and the uncertainties might originate internally from the system itself, and externally from the context where the system is operating at run-time.

Until now the self-adaptive systems community does not have a reference system that is widely used and accepted among everyone in the research field. However, in the community there have been a few proposed exemplars for self-adaptive systems, varying from quite generic such as artifacts, to rather specific—model problems. The exemplars additionally vary in the type of the systems that are being considered. Primarily all the previous research has been focusing on self-adaptive *software systems*, for example, server systems, load balancers, etc. Namely, none of the previous exemplars are depicting and modeling a problem from the *CPSs domain*.

## Foundations

A conceptual model of a self-adaptive system is outlined in Figure 1. On L1 in Figure 2, we present an extended conceptual structure of a self-adaptive system, which maps and depicts the transition from design time to run-time of such systems.

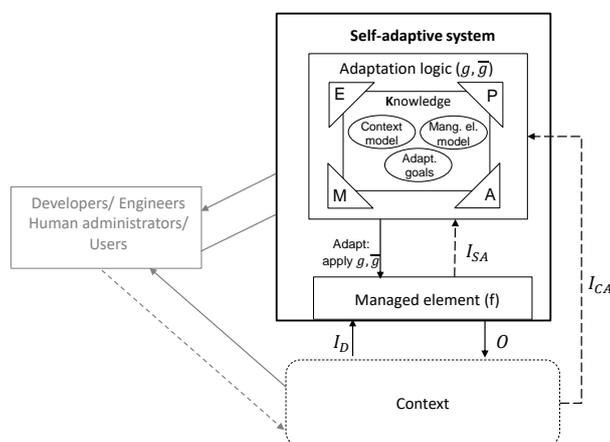


Figure 1: Conceptual model of a self-adaptive system [1].

To prove the notions of self-adaptivity in CPSs, we have created a multi-robot collaborative dirt exploration and allocation use-case from the robotics domain. Initially, as part of a previous master thesis [2] we have built a ROS-based multi-robot system (depicted as the managed element in Figure 1), whose implementation, together with the context and the adaptation logic was later completed during a following lab course. A ROS graph of the technical implementation is shown on L3 in Figure 2.

To bridge the gap between the conceptual model and the technical implementation of a self-adaptive system, in this guided research project we want make a methodological contribution and propose a logical architecture for engineering self-adaptive CPSs (L2 in Figure 2). The final aim is to come up with an architecture that should generalize the process from the existing implementation, and offer a generic way how to specify and implement a self-adaptive CPS, which preferably is not domain specific.

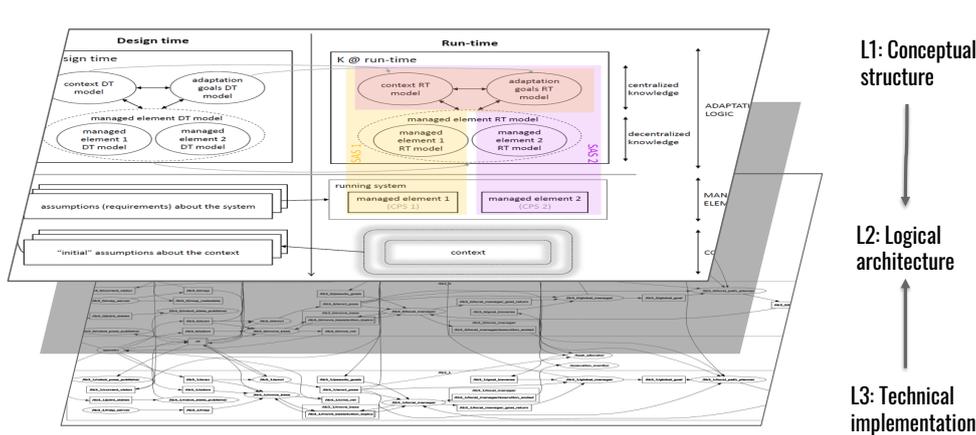


Figure 2: Self-adaptive CPSs - Logical and technical frameworks



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>

## Goal

Define a logical architecture for self-adaptive systems. In the minimal case, we want to demonstrate its viability based on the multi-robot use-case, based on which the architecture would be constructed at first place. In an ideal case, this should provide insights about the strengths and the weaknesses of the reference architecture. If the time constraints allow, we want to investigate the usability of the reference architecture in engineering a different self-adaptive CPSs from another domain. However, similarities between the systems need to be there.

Additionally, another important contribution of this guided research would be coming up with requirements and constraints on what format of input do we need to receive from the stakeholders of the system at design time. Namely, as input we need some model of the context and some specifications about the functional requirements and the quality objectives. Based on that, how to incorporate the requirements and assumptions from design time into the system so that it can base its run-time adaption on them.

## Working Plan

1. Familiarize with the literature on self-adaptive systems in more details, and on previously proposed logical and reference architectures for different systems
2. Based on the conceptual structure and the technical implementation of the multi-robot use case:
  - (a) Discuss and define the characteristics of the logical architecture:
    - i. With respect to scope and level of abstraction
    - ii. How requirements and context assumptions are formulated
3. Discuss how to evaluate the approach, and perform the evaluation
4. Write report & technical description of implementation

## 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.
- Final report written in conformance with TUM guidelines.

## Pre-requisite

- Good analytical skills
- Good Python and C/C++ skills
- Previous knowledge and experience with ROS
- Familiar with the technical implementation of the multi-robot system

## References

- [1] A. Petrovska and A. Pretschner, "Formalizing self-adaptivity: Creating a foundation to understand the systems of the future," *International Conference on Software Engineering*, submitted, 2019.

[2] S. Yousfi, "Design and implementation of ros-based simulated multi-robot systems for data collection," Master's thesis, Technical University of Munich, 12 2018.



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>