2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS*W)

Learning Approach for Smart Self-Adaptive Cyber-Physical Systems

Ana Petrovska
Department of Informatics
Technical University of Munich
Munich, Germany
ana.petrovska@tum.de

Supervisor: Prof. Alexander Pretschner
Department of Informatics
Technical University of Munich
Munich, Germany
alexander.pretschner@tum.de

***Abstract*—Modern Cyber-Physical Systems (CPSs) need to be able to operate efficiently and reliably within continually changing, uncertain, and unanticipated environments. Namely, these systems should be capable of learning, automatically reconfiguring themselves, and be able to cooperate and collaborate with other CPSs. In a nutshell, exhibit human-like, smart capabilities in an autonomic manner. However, engineering such systems is all but trivial, primarily because we need to develop systems at design-time that are capable of autonomously coping with the uncertainty and change at runtime. Therefore, not only the importance of self-adaptivity as a system's feature increases, but it becomes a fundamental approach for the systems to continue meeting their functional specifications, fulfilling their business objectives while preserving the performance—despite all the runtime changes that the system may encounter. To tackle these challenges, this paper proposes the initial research vision and agenda with the envisioned contributions towards an approach for self-adaptation of cooperative, smart CPSs through shared knowledge and learning.**

***Index Terms*—Self-Adaptive Cyber-Physical Systems, Context, Change and Uncertainty, Awareness, Learning, Performance**

I. Motivation

In dynamic systems such as cyber-physical systems (CPSs), the change is the only constant, and business continuity requires this change to be handled at runtime. The changes may stem internally from the system (self-changes), for example, sensor uncertainties; or they may also be external, for instance, caused by changing context or environment where the system is operating (context-changes). In this work, the system's self-* attributes, particularly self-adaptation is explored, which aims to support the engineering of systems at design-time that will have the ability to autonomously and independently modify themselves to successfully cope with the *change* at runtime without external human intervention.

On a conceptual level, a self-adaptive system (SAS) consists of (i) a *managed element* that gains the ability to exhibit self-adaptation; and (ii) *adaptation logic* (AL), which is the "brain" of the SAS and the unit that gives the ability to the managed-element to self-adapt. Across the literature, the AL has been referred with different terms such as managing system [1], autonomic manager [2], etc. The AL is often realized accordingly to the MAPE-K (**M**onitor, **A**nalyse, **P**lan, **E**xecute) [2] closed feedback loop with shared **K**nowledge among all the elements of the loop. For simplicity in this paper, we will put the focus on the knowledge component of MAPE-K only.

The SAS operates in and interacts with the context, which is a relevant part of the environment for a specific system, and influences the system's behavior and state but cannot be influenced by the developers of the system. This means that the potential behavior of the context—consequently the input that the system receives at runtime—can not be fully predicted during the development of the system. Therefore the system needs to have mechanisms to cope with the change in the system itself and in the context during system operation. In other words, the scope of this work is to explore to what extent we can develop software that can handle conditions that were not fully anticipated at the time when the software was developed [1].

The majority of the previous works: architecture-based [3], [4], model-based [5]–[8], reflection-based [9], service-oriented [10], [11], requirements-based [12], [13], formal methods based [14]–[16], and even learning-based [17] focused on providing approaches where the AL is predetermined and consists of "hard-coded" knowledge. This knowledge usually presents an abstraction of relevant aspects of the CPS(s)—or the managed element(s), the context and the system's adaptation goals. It is created at the *design time* and *does not* improve during the operation time of the system. However, having predetermined AL cannot provide adequate adaptation when the CPSs and the context are dynamic and changing unpredictably during runtime. Therefore, these changes should accordingly reflect on the AL. The AL should mimic human-like activities like learning and storing knowledge, which would allow making smarter decisions based on the previously encountered situations, and the currently perceived state of the systems and the context.

II. Research Objectives

Problem. The knowledge that is encoded in the AL of a SAS at design-time, cannot fully anticipate the behavior of the managed element (or the CPS) and the context, in which the SASs will be operating during runtime.

Solution. An approach that will deal with both *uncertainties that come from sensors* and *changing contexts* of multiple CPSs by developing more complex AL.

978-1-7281-2406-3/19/$31.00 ©2019 IEEE
DOI 10.1109/FAS-W.2019.00061

Contribution. Collaborative AL that realizes adaptation strategies based on higher-order or global-system-level context models, built through aggregating knowledge of what was perceived from the distributed context monitoring from all the involved SAS. Additionally, the AL continuously stores and considers all the previously encountered contextual situations. Therefore, the initially encoded knowledge in the AL at design-time enhances and improves during runtime.

An application from the robotics domain will be used to evaluate this approach and all the concepts developed throughout this thesis. The initial evaluations will be done on a simulated multi-robot system, and real hardware will be used for the final evaluation. Our robotics system consists of two robots, each considered as a separate SAS. We have divided the system's goals into two groups: mission goals and self-adaptation goals. The mission goal is related to the functional requirements of the system: both robots keep a room clean, in which new dirt is continuously and perpetually appearing at random locations. The self-adaptation goals are two-fold and are associated with the non-functional requirements. First, optimization of the overall system performance by minimizing the time needed for the room to be cleaned and to be kept clean, despite the changing context originating from having two robots deployed in the same room and the random appearance of dirt (new tasks for the robots). Our initial evaluations show $\sim 10\%$ decrease in the system performance, *only* by having two robots deployed in the room instead of one. The second self-adaptation goal is increased fault tolerance by preventing robots from failure and deadlocks, despite the systems' sensors' uncertainties. The goals in our system are fixed; therefore, we don't have a goal-driven self-adaptation.

III. Challenges and Methodology

To accomplish the objectives described in the previous section, we need to overcome two major challenges.

Distributed context monitoring and knowledge aggregation. In our work, the overall proposed system is decentralized, i.e., no central unit has a complete overview or does global monitoring of the context. Consequently, the robots do not know on which places in the room the new dirt appears. Therefore, there are two possibilities: (1) No knowledge aggregation—each robot is cleaning a dedicated space in the room. In this case, both robots behave as independent systems. Consequently, not knowing what is happening with the other robot and its local context brings inefficiency to the overall performance. With no aggregation of the knowledge, only a local minimum is possible, and we need a global minima of the time required for the room to be cleaned/kept clean. (2) Collaboratively aggregating knowledge from both robots, as two different sources, through distributed monitoring—building more complex, collaborative AL which will allow dealing with situations when specific places in the room get dirtier than others. However, aggregating knowledge in the AL adds an additional layer of complexity. The partial observation of the context persists even after cumulating the knowledge, due to the sensors range limitations of the robots. And the other more significant problem are the cases where there are conflicts in the perceptions or the monitoring, concretely when both robots perceive different information in the same location, due to sensors uncertainties.

Collaborative tasks assignment. After the locations of the newly appearing dirt are identified, the second challenge is how to assign these locations to the robots? The robots should traverse the shortest path possible to the task destination (the dirt location) and at the same time traverse paths with a higher probability of new dirt appearing—so that dirt can also be cleaned along the way. A potential solution can be a manipulation of locations to be visited by each robot considering the current state of the aggregated context model and the stored previously encountered contextual situations. The latter is a grid probability map of the dirt appearances at particular locations/cells in the room, which is updated every time a new dirt is detected. This solution will potentially lead to a better and smarter path planning.

To address both of the open challenges, we want to propose a data-driven approach and investigate different AI solutions for intelligent decision making and learning, which will bring us from the monitored data on the local level by each managed element (or CPS) to global-level, aggregated AL. We plan to conduct experiments using the simulated robotic system to learn the context and optimize the performance for different contextual situations.

Although there have been a few proposed approaches in the literature of self-adaptive systems that have utilized different ways of learning [18]–[20], none of these approaches relies on higher-orders of general, aggregated context models, and collaborative AL that involves cooperative strategies between multiple SASs and considers stored knowledge based on the past experiences. To show the applicability and the generalizability of approach we will evaluate in on a real system and potentially on an application in another domain.

IV. Research Plan and Future Work

My doctoral research is halfway through its four-year period. Hence the visions and some of the contributions for the fulfillment of the research objectives are still to be considered partial. The research plan is shown on the table below.

Goals	Description	Deadline	Status
G1	Finding the thesis topic		
G1.1	Systematic mapping study on sensors, uncertainties, CPSs	12/2017	completed
G1.2	Uncertainties classification	07/2018	completed
G1.3	Systematic mapping study on self-adaptive CPSs utilizing different (machine) learning techniques	07/2018	completed
G2	Theoretical part		
G2.1	Identify approaches for engineering SASs	07/2018	completed
G2.2	Propose conceptual model of SASs	09/2018	completed
G2.3	Design approach for smart self-adaptive CPSs	10/2018	completed

G3	Practical part		
G3.1	Investigate and design a reference problem	10/2018	completed
G3.2	Design and implement a ROS-based robotics system testbed for evaluation based on the previously identified reference problem using simulation	12/2018	completed
G3.3	Decide on what real hardware could be used for evaluation, assemble the TurtleBots 3 and deploy the previously implemented software system	05/2019	ongoing
G3.4	Implement aggregating knowledge from two sources	10/2019	
G3.5	Implement queue manipulation approaches	10/2019	
G3.6	Evaluate using simulation	10/2019	
G3.7	Evaluate on the real hardware	02/2020	
G4	Thesis writing	02/2021	

References

[1] Danny Weyns. Software engineering of self-adaptive systems: an organised tour and future challenges. *Chapter in Handbook of Software Engineering*, 2017.

[2] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, (1):41–50, 2003.

[3] David Garlan, S-W Cheng, A-C Huang, Bradley Schmerl, and Peter Steenkiste. Rainbow: Architecture-based self-adaptation with reusable infrastructure. *Computer*, 37(10):46–54, 2004.

[4] Naeem Esfahani, Ehsan Kouroshfar, and Sam Malek. Taming uncertainty in self-adaptive software. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, pages 234–244. ACM, 2011.

[5] Jacqueline Floch, Svein Hallsteinsen, Erlend Stav, Frank Eliassen, Ketil Lund, and Eli Gjorven. Using architecture models for runtime adaptability. *IEEE software*, 23(2):62–70, 2006.

[6] Luciano Baresi, Liliana Pasquale, and Paola Spoletini. Fuzzy goals for requirements-driven adaptation. In *Requirements Engineering Conference (RE), 2010 18th IEEE International*, pages 125–134. IEEE, 2010.

[7] Mathieu Acher, Philippe Collet, Franck Fleurey, Philippe Lahire, Sabine Moisan, and Jean-Paul Rigault. Modeling context and dynamic adaptations with feature models. In *4th International Workshop Models@ run. time at Models 2009 (MRT'09)*, page 10, 2009.

[8] Douglas Eskins and William H Sanders. The multiple-asymmetric-utility system model: A framework for modeling cyber-human systems. In *Quantitative Evaluation of Systems (QEST), 2011 Eighth International Conference on*, pages 233–242. IEEE, 2011.

[9] Éric Tanter, Jacques Noyé, Denis Caromel, and Pierre Cointe. Partial behavioral reflection: Spatial and temporal selection of reification. *ACM SIGPLAN Notices*, 38(11):27–46, 2003.

[10] Radu Calinescu, Lars Grunske, Marta Kwiatkowska, Raffaela Mirandola, and Giordano Tamburrelli. Dynamic qos management and optimization in service-based systems. *IEEE Transactions on Software Engineering*, 37(3):387–409, 2011.

[11] Valeria Cardellini, Emiliano Casalicchio, Vincenzo Grassi, Stefano Iannucci, Francesco Lo Presti, and Raffaela Mirandola. Moses: A framework for qos driven runtime adaptation of service-oriented systems. *IEEE Transactions on Software Engineering*, 38(5):1138–1159, 2012.

[12] Andres J Ramirez, Betty HC Cheng, Nelly Bencomo, and Pete Sawyer. Relaxing claims: Coping with uncertainty while evaluating assumptions at run time. In *International Conference on Model Driven Engineering Languages and Systems*, pages 53–69. Springer, 2012.

[13] Jon Whittle, Pete Sawyer, Nelly Bencomo, Betty HC Cheng, and Jean-Michel Bruel. Relax: Incorporating uncertainty into the specification of self-adaptive systems. In *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International*, pages 79–88. IEEE, 2009.

[14] Antonio Filieri, Carlo Ghezzi, and Giordano Tamburrelli. Run-time efficient probabilistic model checking. In *Proceedings of the 33rd international conference on software engineering*, pages 341–350. ACM, 2011.

[15] Paolo Arcaini, Elvinia Riccobene, and Patrizia Scandurra. Modeling and analyzing mape-k feedback loops for self-adaptation. In *Proceedings of the 10th international symposium on software engineering for adaptive and self-managing systems*, pages 13–23. IEEE Press, 2015.

[16] Zuohua Ding, Yuan Zhou, and Mengchu Zhou. Modeling self-adaptive software systems by fuzzy rules and petri nets. *IEEE Transactions on Fuzzy Systems*, 26(2):967–984, 2018.

[17] Christopher Simpkins, Sooraj Bhat, Charles Isbell Jr, and Michael Mateas. Towards adaptive programming: integrating reinforcement learning into a programming language. *ACM Sigplan Notices*, 43(10):603–614, 2008.

[18] Ahmed Elkhodary, Naeem Esfahani, and Sam Malek. Fusion: a framework for engineering self-tuning self-adaptive software systems. In *Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering*, pages 7–16. ACM, 2010.

[19] Alessia Knauss, Daniela Damian, Xavier Franch, Angela Rook, Hausi A Müller, and Alex Thomo. Acon: A learning-based approach to deal with uncertainty in contextual requirements at runtime. *Information and software technology*, 70:85–99, 2016.

[20] Ilias Gerostathopoulos, Dominik Skoda, Frantisek Plasil, Tomas Bures, and Alessia Knauss. Architectural homeostasis in self-adaptive software-intensive cyber-physical systems. In *European Conference on Software Architecture*, pages 113–128. Springer, 2016.