Minitrack Introduction: Self-Adaptive Systems: Technologies, Domains, Principles, and Practices

Rose Gamble
University of Tulsa
gamble@utulsa.edu

Matt Hale
University of Nebraska - Omaha
mlhale@unomaha.edu

Abstract

This minitrack focuses on defining, creating, implementing, and evaluating self-adaptive systems for different domains.

1. Introduction

Self-adaptive systems research has been steadily emerging as a force behind the development of autonomous systems [1-3]. These systems may support one or more adaptive operations, such as self-healing, self-repairing, self-maintaining, self-coordinating, self-referencing, and self-securing, and can be applied to a wide variety of systems. These systems can include services, wearables, and IoT. The design and implementation of such systems can be extremely challenging [1]. As such, this field has a special interest in new software development methods, techniques, and frameworks that can facilitate more efficient processes to implement self-adaptive systems [1]. In addition, a significant challenge moving forward is to produce accurate models of self-adaptive systems that can enhance new or existing software development methods or provide improvements in decision making, performance, sustainability, security, and usability of new or existing self-adaptive systems [3].

The second offering of the minitrack accepted three papers. In “Active Loop Programming for Adaptive Systems”, Laundauer and Bellman demonstrate a new software development method called *active loop programming*, inspired by biological principles of organization, that can be used to design self-organizing adaptive systems. In “Architectural Principles for Autonomous Microservices”, Mikkelsen et al. identify a set of design principles that can be applied to create an asynchronous and agnostic microservice architecture. Additionally, they provide a toolkit that can be used to quickly implement such architectures. Lastly, in “The JDownloader Immune System for Continuous Deployment”, Rechenmacher, Riehle, and Weber present a new fitness model that can evaluate a continuously deployed software application called *JDownloader*.

2. References

[1] Software Engineering for Self-Adaptive Systems I, Editors: Cheng, B.H.C., de Lemos, R., Inverardi, P., Magee, J., Springer, 2009.

[2] Software Engineering for Self-Adaptive Systems II, Editors: de Lemos, R., Giese, H., Müller, H., Shaw, M. Springer, 2013.

[3] Software Engineering for Self-Adaptive Systems III, Editors: de Lemos, R., Garlan, D., Ghezzi, C., Giese, H., Springer, 2017.

URI: https://hdl.handle.net/10125/64543
978-0-9981331-3-3