

SOAR: Self-Organizing ARchitectures

Danny Weyns¹, Sam Malek², Rogério de Lemos³, Jesper Andersson⁴

¹ *DistriNet Labs, Katholieke Universiteit Leuven, Belgium*
danny.weyns@cs.kuleuven.be

² *Department of Computer Science, George Mason University, USA*
smalek@gmu.edu

³ *Department of Informatics Engineering, University of Coimbra, Portugal*
rdelemos@dei.uc.pt

⁴ *Department of Computer Science, Växjö University, Sweden*
jesper.andersson@vxu.se

Abstract—*Self-adaptability has been proposed as an effective approach to automate the complexity associated with the management of modern-day software systems. The study of self-adaptability has taken two lines of research: an architecture-centric focus when developing top-down solutions and an algorithmic/organizational focus when developing bottom-up solutions (also referred to as self-organization). Whereas both lines of research have been successful at alleviating some of the associated challenges of constructing self-adaptive systems, persistent challenges remain, in particular when building complex distributed self-adaptive systems. With the term “Self-Organizing Architectures” (SOAR) we refer to an engineering approach for self-adaptive systems that combines architectural approaches for self-adaptability with principles and techniques from self-organization. The general goal of the SOAR workshop is to provide a middle ground that combines the architectural perspective of self-adaptive systems with the algorithmic perspective of self-organizing systems.*

Keywords: *self-adaptive systems, self-organizing systems*

I. INTRODUCTION

Self-adaptability has been proposed as an effective approach to automate the complexity associated with the management of modern-day software systems. Self-adaptability endows a software system with the capability to adapt itself at runtime to deal with changing operating conditions or user requirements.

Researchers on self-adaptive systems mostly take an architecture-centric focus on developing top-down solutions [4][5]. In this approach, the system is monitored to maintain an explicit (architectural) representation of the system and based on a set of (possibly dynamic) goals, the system's structure or behavior is adapted. Researchers of self-organizing systems mostly take an algorithmic/organizational focus on developing bottom-up solutions [2][5]. In this approach, the system components adapt their local behavior or patterns of interaction to changing conditions and cooperatively realize system adaptation. Self-organizing approaches are often inspired by biological or natural phenomena. With the term “Self-Organizing

Architectures” (SOAR) we refer to an engineering approach for self-adaptive systems that combines architectural approaches for self-adaptability with principles and techniques from self-organization.

Whereas both lines of research have been successful at alleviating some of the associated challenges of constructing self-adaptive systems, persistent challenges remain, in particular for building complex distributed self-adaptive systems [1][3]. Among the hard challenges in the architectural-centric approach are handling uncertainty and providing decentralized scalable solutions. Some of the hard challenges in the self-organizing approach are connecting local interactions with global system behavior, and accommodating a disciplined engineering approach. The awareness grows that for building complex distributed self-adaptive systems, principles from both self-adaptive systems and self-organizing systems have to be combined. For instance, web-scale information systems, intelligent transportation systems, and the power grid are all innately decentralized systems, but control in local sub-systems may be highly centralized. Engineering such complex systems puts forward questions such as: What kind of bottom-up mechanisms can be exploited in order to deal with uncertainty but at the same time provide the required assurances? How to derive and exploit tactics, architectural patterns, and reference architectures to realize robust, scalable, and long-lived solutions?

II. GOALS AND TOPICS

The general goal of the SOAR workshop is to provide a middle ground that combines the architectural perspective of self-adaptive systems with the algorithmic perspective of self-organizing systems. Concretely, the workshop aims to identify the critical challenges and advance the state-of-the-art in:

- *Software architectures* (tactics, patterns, reference architectures, etc.) for complex self-adaptive systems that integrate principles from both self-adaptability and self-organization.

- *Design* (modeling, analysis, synthesis) of self-adaptive systems that exploit principles of self-organization to deal with uncertainty and large scale.
- *Construction* (frameworks, middleware, applications, etc.) of software systems based on self-organizing architectures in practice.

Topics of interest for SOAR include, but are not limited to:

- Architectural patterns and tactics for self-adaptive systems
- Reflective architectures for self-adaptive systems
- Decentralized control in dynamic software architecture
- Multi-agent system architectures for self-adaptation
- Self-representations in decentralized systems
- Dealing with uncertainty in self-adaptive systems
- Control of emergent properties in self-adaptive systems
- Resilience of self-adaptive systems
- Instrumentation for realizing self-adaptation
- (Ultra) large-scale self-adaptive systems
- Self-adaptation and software product lines
- Application of principles from biology, sociology and physics to engineer self-adaptive systems
- Quality of service concerns in self-adaptive systems
- Applications of self-adaptive and self-organizing systems

III. DISSEMINATION

Workshop notes with the accepted papers will be distributed at the workshop.

Post-proceedings with a selection of revised and extended workshop papers, complemented with invited papers from renowned researchers in the field, will be published by Springer as a volume in the Lecture Notes in Computer Science series.

IV. SCIENTIFIC COMMITTEE

The program committee of the workshop consists of experts from both self-adaptive systems and self-organizing systems:

- Nelly Bencomo (Lancaster University, UK)
- Yuriy Brun (University of Southern California, USA)
- David Garlan (Carnegie Mellon University, USA)
- Kurt Geihs (University of Kassel, Germany)
- Holger Giese (Hasso Plattner Institute at the University of Potsdam, Germany)
- Jorge J. Gómez Sanz (Universidad Complutense de Madrid, Spain)
- Tom Holvoet (DistriNet Labs, Katholieke Universiteit Leuven, Belgium)
- Marco Mamei (DISMI, Università' di Modena e Reggio Emilia, Italy)
- Hausi A. Müller (University of Victoria, Canada)

- Flavio Oquendo (Université Européenne de Bretagne, Université de Bretagne-Sud, France)
- H. Van Dyke Parunak (Vector Research Center, division of TTGSI, Ann Arbor, USA)
- Onn Shehory (IBM Haifa Research Lab, Israel)
- Mirko Viroli (Alma Mater Studiorum - Università di Bologna, Italy)

V. WORKSHOP WEBSITE

<http://distrinet.cs.kuleuven.be/events/soar/2009/>

ACKNOWLEDGMENT

We are grateful to the WICSA/ECSA 2009 organizers for supporting the organization of SOAR. We thank the members of the program committee. Last but not least, we thank all the authors for submitting their work to SOAR.

REFERENCES

- [1] J. Andersson, R. de Lemos, S. Malek, and D. Weyns. Reflecting on Self-Adaptive Software Systems. *ICSE 2009 workshop on Software Engineering for Adaptive and Self-Managing Systems*, Vancouver, Canada, May 2009.
- [2] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Inc., New York, NY, USA, ISBN 0-19-513159-2, 1999.
- [3] B. H. C. Cheng, et al. Software Engineering for Self-Adaptive Systems: A Research Roadmap. In *Software Engineering for Self-Adaptive Systems*, Editors B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, Lecture Notes in Computer Science, Springer, 2009.
- [4] J. Kramer and J. Magee. Self-Managed Systems: an Architectural Challenge. *International Conference on Software Engineering*, Minneapolis, Minnesota, May 2007.
- [5] P. Oreizy, N. Medvidovic, and R. Taylor. Architecture-Based Runtime Software Evolution. *International Conference on Software Engineering*, Kyoto, Japan, May 1998
- [6] H. V. D. Parunak, Go to the Ant: Engineering Principles from Natural Agent Systems, *Annals of Operations Research*, vol. 75, pp. 69-101, 1997.