Quo vadis, self-* software?[¶]

Veronika Szabóová[a], Csaba Szabó[b]

Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
[a]Veronika.Szaboova@tuke.sk, [b]Csaba.Szabo@tuke.sk

Abstract

The research on self-* software began with Laddaga's DARPA report [1]: "Self-adaptive software evaluates its own behavior and changes behavior when the evaluation indicates that it is not accomplishing what the software is intended to do." As time went, there was significant growth in the field of self-adaptive systems, many self-* properties were defined to allow better focused research. Guinea presented in [2] in 2013 a very important process characteristics as well: "Its life-cycle cannot be stopped after its development and initial setup." Which means that the challenge of self-* systems is to "live and change." Changes alter the software itself but could reconfigure its operating environment as well. Or implement the adaption using both ways. We focus on internal changes. Regarding to these self-modifications, Salehie and Tahvildari wrote in [3]: "there is still a lack of powerful languages and frameworks that could help realize adaptation processes." Our goal is to address this problem in this article. We focus on implementation of self-modification and related issues of self-testing of the changes introduced.

Keywords: self-adaptive, self-testing, self-* properties

MSC: 68N30; 68N20

References

[1] LADDAGA, R., Self Adaptive Software SOL BAA 98-12, *DARPA/ITO* (1998).

[2] GUINEA, S., Software Architecture for Adaptive Systems, *Self-Adaptive Software Systems PhD Course with Carlo Ghezzi at Politecnico di Milano* (2013).

[3] SALEHIE, M., TAHVILDARI, L., Self-Adaptive Software: Landscape and Research Challenges, *ACM Trans. Autonom. Adapt. Syst.* Vol. 4 No. 2 Article 14 (2009), 42 p.