

Chapter 8

Toward Impact Generation and Future Research



Joao Pita Costa, Elisabetta Di Nitto, and Indika Kumara

Abstract This book has presented the main results developed by the SODALITE project, some examples of models an end-user can develop for the purpose of exploiting the SODALITE deployment and operation features, and the three specific case studies that have been used as first experimental playground for the approach. In this chapter we wrap up by first providing an overview of how the SODALITE results are being packaged in ready-to-use tools and then discussing about future research challenges.

8.1 Impact Generation: The SODALITE Stack

In order to determine the scope of the SODALITE technology from the perspective of the usefulness of its inventions, i.e., its innovation, we organized the SODALITE results as a stack of five innovation layers that highlight the multidimensional capability of this software toolset (see Fig. 8.1 for illustration and [1] for the formal definition of software stack that we follow). All layers shown in the figure can either be used in combination with each other or they can be adopted in isolation and exploited in any TOSCA-based context.

J. Pita Costa
XLAB, Ljubljana, Slovenia
e-mail: joao.pitacosta@xlab.si

E. Di Nitto (✉)
Politecnico di Milano, Milan, Italy
e-mail: elisabetta.dinitto@polimi.it

I. Kumara
Jheronimus Academy of Data Science, Tilburg University, Tilburg, The Netherlands
e-mail: i.p.k.weerasinghadewage@tilburguniversity.edu

© The Author(s) 2022
E. Di Nitto et al. (eds.), *Deployment and Operation of Complex Software in Heterogeneous Execution Environments*, PoliMI SpringerBriefs,
https://doi.org/10.1007/978-3-031-04961-3_8

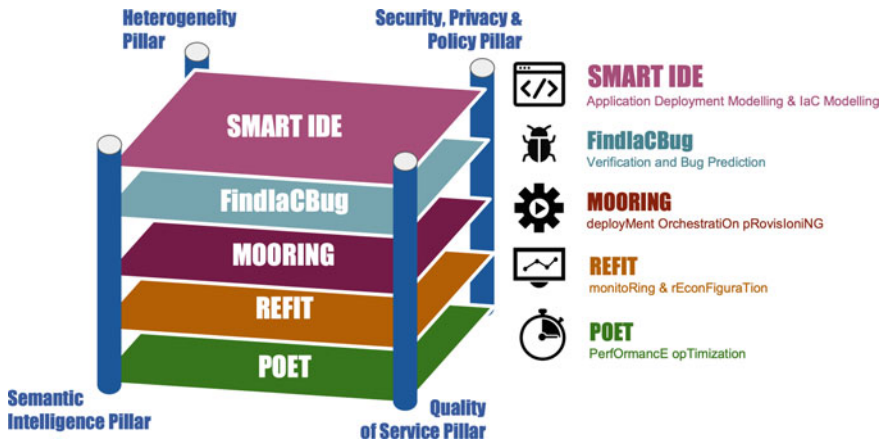


Fig. 8.1 Layered presentation of the SODALITE stack

The first innovation layer includes all elements that constitute the SMART IDE (see Chap. 3) and allow end users to define Abstract Application Deployment Models and to transform them into executable IaC.

The second innovation layer, FindIaCBug offers the verification and bug prediction features (see Chap. 4, Sects. 4.2 and 4.3).

The third and fourth innovation layers, MOORING and REFIT, focus, respectively, on orchestration of deployment (see Sect. 5.2) and monitoring and reconfiguration (see the rest of Chap. 5).

Finally, the last innovation layer includes POET, which encompasses the performance optimization mechanisms available at design time (Chap. 4, Sect. 4.4) and at runtime (Chap. 5, Sect. 5.6).

All above layers are sustained and supported by four pillars that constitute the main structure of the whole SODALITE solution. These concern the usage of semantic and AI technologies (the semantic intelligence pillar), the attention to Quality of Service, the focus on enabling the usage of heterogeneous infrastructures and, finally, the emphasis on security and privacy. For space reason, this last point has not been described in detail in this book. On the one side, it concerns the adoption of state of the art approaches to access and secret management, security vulnerability scanning, and code inspection. On the other side, it includes the attention to the quality and security issues in IaC scripts generated from the deployment models created by the users through the Smart IDE (see Chap. 4).

8.2 Future Research

SODALITE represents one of the first attempts to tackle the problem of simplifying the work of those who need to automate the deployment and operation of complex software. In particular, SODALITE has tried to address the problem from multiple perspectives, from the introduction of smart editing features to the usage of ontologies and machine learning to infer possible bugs as well as possible code completion. Problems such as the automated generation of execution containers and the identification of the optimal ones given a specific execution environment have been addressed as well, together with the automated generation of models for the discovered resources and the automated identification of possible refactoring options.

The solutions we have developed show that all above problems can be addressed. There are though a number of challenges that are still to be addressed. Among the others, we highlight the following ones.

- *Single modelling approach and compatibility with multiple IaC languages*: even if SODALITE offers modelling support to the definition of application deployment models, still the user does not face a single modelling paradigm, but he/she should be able to handle the usage of multiple Domain Specific Languages that are heterogeneous in terms of their characteristics. A single paradigm for specifying application deployment would certainly help the users and facilitate their work. If the result of the modelling phase could be translated into multiple languages, the approach could be adopted by users of different orchestration frameworks (e.g., TOSCA and Terraform).
- *Deployment and operation frameworks for the cloud continuum*: SODALITE has considered HPC, Cloud and edge resources. A natural continuation of the project would be to manage heterogeneous resources as a seamless continuum where components could be dynamically deployed on the most suitable resource types and could be moved in the continuum if the situation in terms of resource consumption and application-level requirements evolves.

Both aforementioned areas are becoming the focus of new projects and initiatives (see, for instance, the H2020 PIACERE project¹). We do hope that the findings, experience, and concrete results we have achieved in SODALITE will be beneficial for them and, possibly, reused and extended.

¹<https://www.piacere-project.eu/>.

Reference

1. Hack JJ, Heinonen N, Papka ME (2019) Software stack in a snapshot. *Comput Sci Eng* 198(21):114–116

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

