A Smart Development Environment for Infrastructure as Code

Jesús Gorroñogoitia[1], Zoe Vasileiou[2], Emilio Imperiali[3], Indika Kumara[4], Dragan Radolović[5], Georgios Meditskos[2]

[1] ATOS, Spain
[2] Information Technologies Institute, Centre for Research and Technology Hellas, Greece
[3] Politecnico di Milano, Italy
[4] Jheronimus Academy of Data Science, The Netherlands
[5] XLAB Research, Slovenia
jesus.gorronogoitia@atos.net, zvasilei@iti.gr, emilio.imperiali@mail.polimi.it, i.p.k.weerasingha.dewage@tue.nl, dragan.radolovic@xlab.si, gmeditsk@iti.gr

Abstract. Cloud computing is a mature paradigm that has evolved to accommodate ever-increasing complex applications such as in the AI and HPC domain. If applications are complex, infrastructure can be even more, spanning over hybrid architectures. As such, producing a less error-prone deployment while offering high performance requires application and infrastructure awareness, and also deep knowledge of the IaC languages. In this paper, we present the SODALITE IDE, a suite that assists the users in the authoring of application deployment topology and infrastructure models for IaC. With focus on quality and performance, the IDE enables the faster and simpler development of IaC by offering features such as in-sync multiple model viewpoints , smart context-aware content assistance and semantic validation, powered by a Knowledge Base.

Keywords: Infrastructure as Code, TOSCA, Ansible, IDE, Semantics

1 Introduction

In recent years, the global market has seen a tremendous rise in utility computing serving as the backend for practically any new technology, methodology or advancement from healthcare to aerospace. SODALITE addresses complex tasks of configuration, deployment and operation of complex applications. The development of these tasks implies knowledge of multiple IaC scripting languages and being able to manage the whole development process of IaC. Given those intricacies, the simplification and abstraction of those DevOps processes is uppermost. To this end, SODALITE provides tools for a simpler and faster development of IaC and deployment and execution of heterogeneous applications in HPC, Cloud & SW defined computing environments, with particular focus on quality, performance, and manageability. Following this vision, SODALITE offers smart modeling capabilities to help non-expert DevOps teams in defining *Abstract Application Deployment Models* (AADMs) . The main novelty of

SODALITE regarding the support to the definition of AADMs is its ability to simplify this task by offering semantic-based guidance to the end users.

The purpose of this demo paper is to describe the SODALITE Integrated Development Environment (SODALITE IDE) highlighting its ability to create trustworthy AADMs. The paper is structured as follows: Section 2 presents the state of the art in the creation of AADMs; Section 3 presents the main characteristics of the SODALITE IDE; Section 4 provides a description of the preliminary usability experiments; Finally, Section 5 draws the conclusion and delineate the plan for future work.

2 State of the art

Several modeling languages have been proposed for supporting the specification of complex application topologies and their deployment into infrastructures including Cloud [6] [3] [2] [4] and HPC [7] [8]. The Topology and Orchestration Specification for Cloud Applications (TOSCA [6]) is a fast emerging standard in the Cloud realm, but lacks visual notation, leading to the appearance of authoring tools with their own non-standardized visual notation. For addressing this limitation, researchers have designed their proposals to standardize the visual notation for TOSCA.

Winery [5] is a Web TOSCA visual editor, that can be also included, as a plugin. It separates modeling concerns to support not only resource experts on the specification of TOSCA types, but also application owners on the definition of their application topologies. CloudCAMP DSML [2], a web-based editor, supports the generation of IaC deployment models from users' abstract business-oriented requirements for the creation of application component topologies by utilizing TOSCA node templates and relationships. Alien4Cloud [1], a Web-based editor, enables application owners to design their deployment topology, as an orchestration of components instantiated from types retrieved from a common TOSCA types Catalogue.

The main novelty of the SODALITE IDE is to support the complete specification of both application deployment topologies, and of the resources the application requires on the target infrastructure as model instances of the SODALITE DSL. This DSL has been designed as an abstraction that leverages TOSCA to facilitate the export of AADM topologies as TOSCA blueprints into the SODALITE IaC environment. Moreover, the SODALITE IDE includes support for the creation of Ansible Models (AMs) explicitly associated with the resource node types defined in a Resource Model (RM) and application node types defined in AADMs, thus fully covering design and operation of the node types.

3 SODALITE IDE main characteristics

The SODALITE IDE[6], designed and implemented in Eclipse as a set of plugins, offers editors supporting the modeling of a) infrastructure resources in *resource models* (RMs), b) application deployment topologies in AADMs, c) application optimization in *optimization models* (OMs) and d) operation implementations as Ansible playbooks

[6] AADM & OM demo https://www.youtube.com/watch?v=D5vKpoE47rA

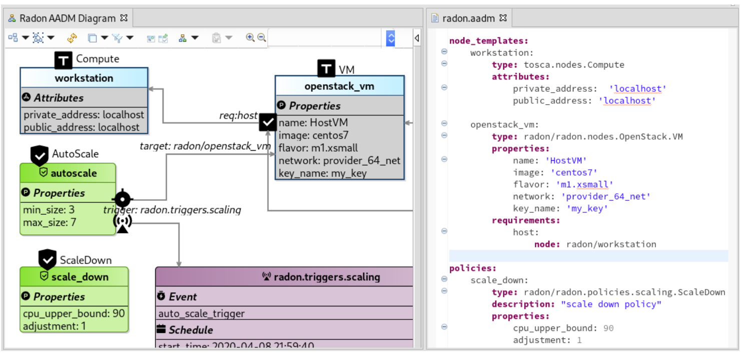

Fig. 1. AADM graphical and textual view representation

in *Ansible models* (AMs). These models constitute instances compliant with the SODALITE DSLs for RMs, AADMs, OMs and AMs, respectively. The RM and AADM DSLs are closely related to the TOSCA specification: RM DSL provides TOSCA related modeling concepts supporting the definition of types for infrastructure resources, and AADM DSL supports the specification of application component instances, as TOSCA templates. This separation of modeling concerns, targeting different modeling roles, simplifies the TOSCA complexity and fosters the reusability of resource types stored in a common, shared SODALITE *knowledge base* (KB).

The current release of the IDE[7] offers textual (XText)editors for all the SODALITE DSLs, and a graphical view representation (Sirius) for the AADM, which is automatically generated (Figure 1). Sirius technology permits the creation of multiple graphical viewpoints of the same DSL. Indeed, the textual representation constitutes another viewpoint representation. These representations are syncronised, making changes in one representation propagate to the others sharing the same underlying model.

The IDE textual editors offer most the features included by XText, remarkably the syntax coloring, and validation, formatting and auto-completion, cross-reference navigation, quick-fix proposals, outline view, and others. Smart context-aware content-assistance and semantic validation is largely supported through the symbolic inference capabilities of the KB (see Figure 2), where infrastructure resources are represented as RDF Knowledge Graphs for further reuse in AADMs. These RDF graphs capture both the structural and semantics relationships of the models, promoting reusability and interoperability using standard ontology languages (OWL2) and reasoning tools. Content-assistance suggests modelers both syntactic (e.g. modeling blocks) and semantic (e.g. reusable resources matching component's requirements) content that fits into the current modeling context. Regarding semantic content, the IDE requests it to the KB, which infers the best matching content from current context (e.g. application component) by searching within its repository. A similar process is performed by the KB, upon IDE request, to conduct the semantic validation of the AADM, resulting in reported issues, such as errors and recommendations, with associated quick fixes, which

[7] https://github.com/SODALITE-EU/ide

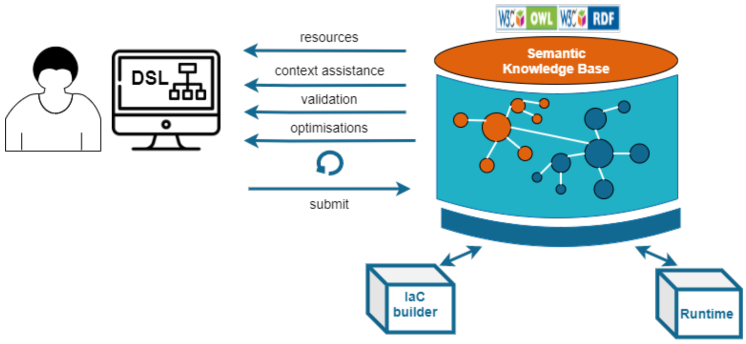

Fig. 2. SODALITE IDE conceptual architecture

are shown to the modeler, in the textual editor, at the point of occurrence. The same KB-powered assistance and validation is supported for RMs and OMs, and it is planned for AMs. AADMs and RMs can be stored into the KB, enabling the reuse of the resource instance and types defined in them. Additionally, the SODALITE IDE provides support to browse the KB, retrieve, modify and delete models stored therein. It also supports other SODALITE processes, notably the deployment of AADMs into target infrastructures, the creation of resource images, and the governance of deployments.

Ansible Models are used to provide the implementation of an operation defined in the RM. The IDE provides support for this integration between TOSCA and Ansible, by allowing to couple an AM with an operation from a RM. In this way, while the user is defining the AM, the IDE suggests the available inputs provided by the TOSCA operation, that can be used in the AM. Furthermore, the AM DSL [8], while being related to the Ansible specification, conceptually groups the Ansible attributes in categories based on their usage, for better organization of the code. The IDE takes care of generating the concrete Ansible playbook from the abstract AM, that can then be executed for the deployment in the target infrastructures.

4 Preliminary experiments

We performed some controlled experiments with two different types of users, namely non-expert users (9 participants) and TOSCA experts (5 participants), with the objective to receive feedback on perceived ease of use, usefulness and intention to use of the DSL and IDE. The mentioned factors are usually determining the user acceptance of a particular technology. All experiment participants have been asked to focus on the development of AADMs for a Machine Mearning (ML) application, which consists of 1) a database that stores training data, 2) a component that trains a ML model 3) a repository that stores trained machine learning models, 4) a component that makes predictions/inferences based on the trained models. The use case owners used their corresponding use cases.

[8] AM demo - https://drive.google.com/file/d/17X8VcQKwZ7dp4F7R4nN3CRAGmnXUe5pJ/view

The experiments with TOSCA experts showed the SODALITE can help to achieve 27.73% improvement over the baseline consisting in using their typical textual editor for creating a TOSCA topology. The experiments with both groups showed that the users consider the SODALITE IDE very useful, easy to use, and think it has a high adoption potential. As a weak point, some missing features of the IDE , the partial stability and incomplete documentation of the IDE have negatively impacted on the user's effort and time. We plan to work on these aspects in the next period.

5 Conclusion

In this paper, we presented an overview of the SODALITE IDE, a smart suite for IaC, providing full support for the modeling of infrastructure resources, application deployment topologies, application optimizations and operation implementations, and also for the management of deployed applications at runtime. In future, we intend to: (i) provide more advanced context-assistance and validation services (ii) abstract the application model by permitting the concretization of the AADM at deployment time by relying on Semantic Reasoning services (iii) improve the AADM visual modelling authoring from palettes, (iv) enhance OM authoring and (v) further interconnect the different DSLs.

Acknowledgments This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 825480, SODALITE. We thank all members of the SODALITE consortium for their inputs and feedbacks to the development of this paper.

References

1. Alien4cloud 1.1 overview (2016), `https://drive.google.com/file/d/0B-bJgbmOz4ipNlNfYkdsOUlocm8/view`
2. Bhattacharjee, A., Barve, Y.D., Gokhale, A.: Cloudcamp : A model-driven generative approach for automating cloud application deployment and management (2017)
3. Carlson, M., et al.: Cloud application management for platforms (2012), `\url{https://www.oasis-open.org/committees/download.php/47278/CAMP-v1.0.pdf}`
4. Ferry, N., et al.: Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In: 2013 IEEE Sixth International Conference on Cloud Computing. pp. 887–894 (2013)
5. Kopp, O., Binz, T., Breitenbücher, U., Leymann, F.: Winery - a modeling tool for tosca-based cloud applications. In: ICSOC (2013)
6. Oasis: Oasis topology and orchestration specification for cloud applications version 1.0 (2013), `\url{http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.pdf}`
7. Palyart, M., Lugato, D., Ober, I., Bruel, J.M.: Mde4hpc: An approach for using model-driven engineering in high-performance computing. In: SDL 2011: Integrating System and Software Modeling. pp. 247–261. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
8. Śmiałek, M., Rybiński, K., Roszczyk, R., Marek, K.: Towards a unified requirements model for distributed high performance computing. pp. 1–20. Springer International Publishing, Cham (2020)