The Architectural Template Method: Design-Time Engineering of SaaS Applications

Sebastian Lehrig**

Software Engineering Group & Heinz Nixdorf Institute
University of Paderborn, Paderborn, Germany

sebastian.lehrig@uni-paderborn.de

Abstract. Typical requirements of SaaS applications target scalability, elasticity, and cost-efficiency. However, these quality properties lack an engineering method for software architects, allowing them to precisely and efficiently analyze these properties already at early design-time. To tackle this lack, we propose the architectural template (AT) method, an efficient design-time engineering method for analyzing scalability, elasticity, and efficiency properties of SaaS applications. Our method quantifies such properties based on reusable analysis templates of cloud computing environments. Architects only have to fill-in the parts specific to their SaaS application.

Keywords: Cloud Computing, SaaS, Engineering, Quality Analysis

1 Problems in Current Practice

Typical requirements of Software-as-a-Service (SaaS) applications target scalability, elasticity, and cost-efficiency. However, these quality properties lack an engineering method for software architects, both in industry and in academia.

In industry, software architects test finally implemented SaaS applications deployed in the target cloud computing environment (staging or operation phase), without prior quality analysis. Therefore, architects may detect unsatisfying quality properties in these late development phases, potentially leading to expensive re-implementations. Also the testability of such applications itself is limited: generating realistic workloads (over 10,000 requests per minute) is economically (and often also technically) infeasible. Furthermore, the industry has adopted approaches for cost-efficiency analysis (e.g., cost calculators of Amazon or RightScale). These approaches compare typical configurations of different cloud computing environments in terms of operation costs. However, these comparisons are inaccurate because they neglect the concrete SaaS application finally operating in these environments.

In academia, design-time engineering methods for analyzing performance properties exist. These approaches currently have a limited support for scalability, elasticity, and efficiency analyses. Scalability analyses (e.g., Palladio [2]) are semi-automated, i.e., based on a series of manually conducted and interpreted performance analyses. Elasticity analyses (e.g., SimuLizar [1]) allow to model and

** The research leading to these results has received funding from the EU Seventh Framework Programme (FP7/2007-2013) under grant no 317704 (CloudScale).
analyze self-adaptations, typically used by cloud computing environments. However, they have a high modeling effort and provide non-statistically significant results. Efficiency analyses (e.g., CDOSim [3]) require an implemented SaaS application to determine the most cost-efficient cloud computing environment but are limited to IaaS environments and lack support for early design-time analyses.

2 Contributions of the AT Method

To tackle the previously described problems, we propose the architectural template (AT) method [4], an efficient design-time engineering method for analyzing scalability, elasticity, and efficiency properties of SaaS applications. Our method quantifies such properties based on reusable analysis templates. These templates manifest all quality-relevant characteristics intrinsic to given cloud computing environments. Architects only have to fill-in the parts specific to their SaaS application. By leveraging existing approaches, our concrete contributions include:

- **Support** for scalability, elasticity, and efficiency: the AT method will provide support for quantifying scalability, elasticity, and efficiency. For this support, we extend existing approaches like Palladio [2] and SimuLizar [1] by novel metrics and according measurements for such properties.

- **Higher accuracy** of analysis results (compared to competing approaches like Amazon’s cost calculator) by making analyses application-dependent.

- **Reusability** of created analysis models in templates for cloud computing environments. After creation, these templates can be reused for repeating analyses and even for other SaaS applications deployed in such environments. To achieve reusability, we reuse ideas from CDOSim [3] where a set of micro-benchmarks allows to re-calibrate analysis models for their reuse.

- **Efficiency** of the AT method (compared to competing approaches): due to reusability, using our method is more efficient than directly using an existing analysis approach like Palladio [2] or SimuLizar [1].

3 Conclusion

We outline problems in current practices when engineering SaaS applications and propose the AT method [4] to cope with these problems. The AT method promises project managers reducing risks by early quality analyses and researchers an improved accuracy of such quality analyses. In our future work, we plan to provide extensive evaluations of asserted benefits of the AT method.

References