

Using Architecture Knowledge to Improve Automated Software Architecture Design Space Exploration

Axel Busch¹, Anne Koziolk¹

Abstract: During the development of modern software systems software architects make more and more trade-off decisions between many quality attributes. Quality attributes such as performance, reliability, but also others such as security and usability are important aspects in software projects. While the former are comparatively easy to quantify, this is for the latter more difficult or costly. However, existing approaches usually consider either only quantified or only qualitatively modelled quality attributes. With existing approaches, it is not possible to model, analyse, and optimize a subset of quantified attributes and another subset of qualitatively modeled quality attributes together. In this paper, we extend PerOpteryx, an approach to optimize software architectures by meta models and mechanisms for the simultaneous analysis and optimization of quantitatively and qualitatively modelled quality attributes. This makes it possible to include even estimated values for individual quality dimensions in systematic optimization processes.

This is an extended abstract of the paper *Considering Not-quantified Quality Attributes in an Automated Design Space Exploration* published in *QoSA'16* proceedings [BK16].

Keywords: Architecture knowledge; Design Decision; Architecture Trade-off; Automated

1 Analysing Quantified and Not-Quantified Quality Attributes

In modern design processes for software systems, often trade-off decisions have to be made between a variety of quality attributes such as performance, security or reliability. Design decisions regarding the software architecture thereby play a critical role. The quality of the software architecture has proved to be one of the decisive factors influencing the overall quality of the software system [Ko12].

Different approaches, such as Palladio and PerOpteryx, have been shown as promising approaches to predict [Re16] and optimize [Ma10] quality attributes based on a software architecture design. However, these and other approaches can only analyse either quantified or purely qualitatively modeled quality attributes that represents the informal architecture knowledge of the software architect. Approaches that enable the analysis of qualitatively modeled quality attributes are often completely or at least partially automatic processes. However, approaches that only require quantified quality attributes are often not applicable due to the high costs of the quantification.

¹ Karlsruhe Institute of Technology, {axel.busch, anne.koziolk}@kit.edu

Our approach combines both approaches and allows the analysis of qualitative modeled attributes together with quantified analysis functions. Quality attributes can thus always be analysed and optimized quantitatively whenever a quantitative objective function is available. If no function is available or practicable, values can be modeled qualitatively. This enables software architects to decide individually for each quality attribute whether a complex quantification step should be carried out or whether a qualitative assessment would be sufficient. Although estimated values are usually less accurate than measured values or values obtained by quantitative objective functions, they still model important architecture reasoning of the software architect. Furthermore, architecture reasoning is often the only resource when quantitative methods are not available or too costly to perform.

In order to implement our approach, we first extend the Quality Modeling Language (QML) used in PerOpteryx. QML allows the specification of quality attributes, quality dimensions and quality requirements. We extend QML so that values can be modeled on different scale levels, so that we can model ordinal relations, nominal relations and values within a certain range of values as possible manifestations of a dimension. In addition, we extend the meta model of PerOpteryx with elements for the annotation of software components with quality attributes. This allows the annotation of (estimated or measured) quality values on components for arbitrary quality attributes. Finally, we extend the exploration mechanism of PerOpteryx so that not-quantified quality attributes can be optimized together with already existing quantified quality attributes. For this purpose, we consider the quality values of individual components as separate quality dimensions. If an order is defined within a dimension on all possible values, this dimension can be used as objective.

For our evaluation, we consider two case studies. The results of our evaluation show how our approach can be used to include the architecture reasoning of the software architect in a simple manner. One of the results demonstrate how the software architect can analyse the impact of an architecture decision on other quality attributes with low overhead. The evaluation demonstrates that our approach can also be used to analyse the effects of architecture decisions on quantitatively determined quality attributes, even if the architecture decision actually targets at improving a not-quantified quality attribute of the system.

References

- [BK16] Busch, Axel; Koziolek, Anne: Considering Not-quantified Quality Attributes in an Automated Design Space Exploration. In: Proceedings of the 12th International ACM SIGSOFT Conference on the Quality of Software Architectures. QoSA'16. IEEE, pp. 50–59, 2016.
- [Ko12] Koziolek, Anne: Architecture-Driven Quality Requirements Prioritization. In: TwinPeaks. IEEE CS, 2012.
- [Ma10] Martens, Anne; Koziolek, Heiko; Becker, Steffen; Reussner, Ralf H.: Automatically Improve SW Models for Perform., Reliab. and Cost Using Gen. Algor. WOSP/SIPEW ICPE, 2010.
- [Re16] Reussner, Ralf H.; Becker, Steffen; Happe, Jens; Heinrich, Robert; Koziolek, Anne; Koziolek, Heiko; Kramer, Max; Krogmann, Klaus: Modeling and Simulating Software Architectures: The Palladio Approach. The MIT Press, 2016.