

# Software Architecture Optimization: Acting the Way Human Architects Do It

J. Andres Diaz-Pace, Sebastian Frank, André van Hoorn, Alejandro Rago, Santiago Vidal

{adiaz, arago, svidal}@exa.unicen.edu.ar - franksn@studi.informatik.uni-stuttgart.de - van.hoorn@informatik.uni-stuttgart.de

## Abstract

Designing a software system in such a way that it meets its main quality-attribute requirements (e.g., performance, modifiability, etc.) is a complex, and error-prone activity, even for experienced architects. This complexity stems from the existence of multiple alternative solutions for the same requirements, which make tradeoffs inevitable. Design can be seen as a search through a large space, in which the solution space is n-dimensional and each dimension represents a different quality attribute to be optimized. In this context, tool support for exploring alternatives becomes an essential asset for assisting architects to make design decisions. Since the design space has many degrees of freedom, it is usually not possible for a tool to enumerate all possible alternatives. Thus, tools need to resort to heuristic search strategies. Furthermore, since many tradeoffs can be generated by such tools, human architects “in the loop” play a key role in the optimization process.

Over the last years, several heuristic optimizations for software architectures have been developed [1]. Existing approaches have focused on specific qualities (e.g., performance, availability, or modifiability) and have assumed an underlying architectural model (e.g., Palladio or AADL). However, a common limitation of most works is that the knowledge for analyzing and optimizing a predefined (and small) set of quality attributes tends to be hard-wired in the tools, with a high coupling between quality-attribute models. Therefore, evolving a tool to consider new quality attributes and corresponding models is difficult. We argue for a more modular strategy, in which loosely coupled quality-attribute models can be integrated into an architecture optimization platform. The ArchE approach [2] was an early effort in this direction. Furthermore, another limitation of existing works is that (automated) search engines do not often allow for architect’s interventions to drive the search based on the architect’s preferences and goals, which hinders human acceptance of the resulting solutions.

In our collaborative research project SQuAT (Search Techniques for Managing Quality-Attribute Tradeoffs in Software Design Optimizations) — conducted by the University of Stuttgart and the UNICEN University — we jointly investigate semi-automated techniques for software design optimizations, particularly focusing on i) distributed search strategies, ii) modularization of design knowledge, and iii) application of negotiation techniques for managing quality-attribute tradeoffs.

The approach builds on the DesignBots approach [3], which divides the design knowledge into a set of cooperative agents, each of which encapsulates a body of knowledge for analyzing and optimizing architectural scenarios for a particular quality attribute. Like in ArchE, each agent is equipped with architectural transformations (tactics) for changing a given architectural model and acts as a recommendation system (for improvements) to the human architect. The underlying agent architecture of DesignBots ensures a flexible integration of quality-attribute models. Moreover, we have extended the approach with support for concession-based negotiation techniques, so that the agents can interchange their (local) optimal solutions in order to derive (global) architectural solutions with quality-attribute tradeoffs that better reflect architect’s expectations.

While our approach is not tied to specific quality attributes, we are currently focusing on tradeoffs between performance and modifiability in Palladio models [6]. To this end, we are leveraging on existing work on KAMP [5] and PerOpteryx [4]. As for the tactic-based transformations of Palladio models, we are exploring the Henshin engine [7]. The results of the project will be publicly available as open-source software: <https://github.com/SQuAT-Team>

## References

1. A. Aleti et al.: Software Architecture Optimization Methods: A Systematic Literature Review. *IEEE Trans. Software Eng.* 39(5): 658-683 (2013)
2. J. Diaz-Pace et al.: Integrating Quality-Attribute Reasoning Frameworks in the ArchE Design Assistant. In *Proceedings of QoSA '08*. Springer-Verlag, 171-188.
3. J. Díaz-Pace et al.: Exploring Alternative Software Architecture Designs: A Planning Perspective, *IEEE Intelligent Systems*, vol.23, no. 5, pp. 66-77, 2008.
4. A. Koziolk: Automated Improvement of Software Architecture Models for Performance and Other Quality Attributes. KIT Scientific Publishing. PhD thesis, Karlsruhe Institute of Technology (2014)
5. K. Rostami et al.: Architecture-based Assessment and Planning of Change Requests. *Proc. QoSA 2015*: 21-30
6. S. Becker et al.: The Palladio component model for model-driven performance prediction. *J. Syst. Softw.* 82, 1 (January 2009), 3-22.
7. T. Arendt et al.: Henshin: Advanced Concepts and Tools for In-place EMF Model Transformations. *Proceedings of MoDELS'10*, LNCS 6394, 121-135, Springer-Verlag, 2010.