Software Architecture Optimization Methods: A Systematic Literature Review

Aldeida Aleti,* Barbora Buhnova,† Lars Grunske,‡
Anne Koziolek,§ Indika Meedeniya¶

aldeida.aleti@monash.edu, buhnova@fi.muni.cz,
lars.grunske@informatik.uni-stuttgart.de, koziolek@kit.edu, indikamee@gmail.com

Architecture specifications and models [ISO07] are used to structure complex software systems and to provide a blueprint that is the foundation for later software engineering activities. Thanks to architecture specifications, software engineers are better supported in coping with the increasing complexity of today's software systems. Thus, the architecture design phase is considered one of the most important activities in a software engineering project [BCK03]. The decisions made during architecture design have significant implications for economic and quality goals. Examples of architecture-level decisions include the selection of software and hardware components, their replication, the mapping of software components to available hardware nodes, and the overall system topology.

Problem Description and Motivation. Due to the increasing system complexity, software architects have to choose from a combinatorially growing number of design options when searching for an optimal architecture design with respect to a defined (set of) quality attribute(s) and constraints. This results in a design space search that is often beyond human capabilities and makes the architectural design a challenging task [GLB+06]. The need for automated design space exploration that improves an existing architecture specification has been recognized [PBKS07] and a plethora of architecture optimization approaches based on formal architecture specifications have been developed. To handle the complexity of the task, the optimization approaches restrict the variability of architectural decisions, optimizing the architecture by modifying one of its specific aspects (allocation, replication, selection of architectural elements etc.). Hence the research activities are scattered across many research communities, system domains (such as embedded systems or information systems), and quality attributes. Similar approaches are proposed in multiple domains without being aware of each other.

Research Approach and Contribution. To connect the knowledge and provide a comprehensive overview of the current state of the art, we provided a systematic literature review of the existing architecture optimization approaches [ABG+13]. As a result, a gateway

*Monash University, Australia
†Masaryk University, Czech Republic
‡Universität Stuttgart, Germany
§Karlsruher Institut für Technologie, Germany
¶The Portland House Group, Australia

to new approaches of architecture optimization can be opened, combining different types of architectural decisions during the optimization or using unconventional optimization techniques. Moreover, new trade-off analysis techniques can be developed by combining results from different optimization domains. All this can bring significant benefits to the general practice of architecture optimization. In general, with the survey we aimed to achieve the following objectives:

- Provide a basic classification framework in form of a taxonomy to classify existing architecture optimization approaches.
- Provide an overview of the current state of the art in the architecture optimization domain.
- Point out current trends, gaps, and directions for future research.

We examined 188 papers from multiple research sub-areas, published in software-engineering journals and conferences. Initially, we derived a taxonomy by performing a formal content analysis. More specifically, based on the initial set of keywords and defined inclusion and exclusion criteria, we collected a set of papers, which we iteratively analyzed to identify the taxonomy concepts. The taxonomy was then used to classify and analyze the papers, which provided a comprehensive overview of the current research in architecture optimization. The data was then used to perform a cross analysis of different concepts in the taxonomy and derive gaps and possible directions for further research.

The full paper has been published in the IEEE Transactions on Software Engineering [ABG+13].

References

[ABG+13] Aldeida Aleti, Barbora Buhnova, Lars Grunske, Anne Koziolek, and Indika Meedeniya. Software Architecture Optimization Methods: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 39(5):658–683, 2013.

[BCK03] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. AddisonWesley, second edition, 2003.

[GLB+06] Lars Grunske, Peter A. Lindsay, Egor Bondarev, Yiannis Papadopoulos, and David Parker. An Outline of an Architecture-Based Method for Optimizing Dependability Attributes of Software-Intensive Systems. In Rogério de Lemos, Cristina Gacek, and Alexander B. Romanovsky, editors, *Architecting Dependable Systems*, volume 4615 of *Lecture Notes in Computer Science*, pages 188–209. Springer, 2006.

[ISO07] International-Standard-Organization. ISO/IEC Standard for Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems. *ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15*, pages c1 –24, 6 2007.

[PBKS07] Alexander Pretschner, Manfred Broy, Ingolf H. Krüger, and Thomas Stauner. Software Engineering for Automotive Systems: A Roadmap. In *FOSE '07: 2007 Future of Software Engineering*, pages 55–71. IEEE Computer Society, 2007.