Architecture-Based Reliability Prediction with the Palladio Component Model

Franz Brosch, Heiko Koziolek, Barbora Buhnova, Ralf Reussner

FZI Karlsruhe, Germany ABB Corporate Research, Germany
Masaryk University, Czech Republic
Karlsruhe Institute of Technology, Germany
heiko.koziolek@de.abb.com

Software-intensive systems are increasingly used to support critical business and industrial processes, such as in business information systems, e-business applications, or industrial control systems. The reliability of a software system is defined as the probability of failure-free operation of a software system for a specified period of time in a specified environment. To manage reliability, reliability engineering gains its importance in the development process. Reliability is compromised by faults in the system and its execution environment, which can lead to different kinds of failures during service execution: Software failures occur due to faults in the implementation of software components, hardware failures result from unreliable hardware resources, and network failures are caused by message loss or problems during inter-component communication.

To support fundamental design decisions early in the development process, architecture-based reliability prediction can be employed to evaluate the quality of system design, and to identify reliability-critical elements of the architecture. Existing approaches suffer from the following drawbacks that limit their applicability and accuracy.

First, many approaches do not explicitly model the influence of the system usage profile (i.e., sequences of system calls and values of parameters given as an input to these calls) on the control and data flow throughout the architecture, which in turn influences reliability. For example, if faulty code is never executed under a certain usage profile, no failures occur, and the system is perceived as reliable by its users. Existing models encode a system usage profile implicitly into formal models, typically in terms of transition probabilities in the Markov Models characterizing the execution flow among components. Since the models are tightly bound to the selected usage profile, evaluating reliability for a different usage profile requires repeating much of the modeling effort.

Second, many approaches do not consider the reliability impact of a systems execution environment. Even if the software is totally free of faults, failures can occur due to unavailability of underlying hardware resources and communication failures across network links. Neglecting these factors tends to result in less accurate and overoptimistic reliability prediction. On the other hand, approaches that do consider the execution environment typically offer no means to model application-level software failures, which also results in a limited view of software system reliability.

Third, many approaches use Markov models as their modeling notation, which is not aligned with concepts and notations typically used in software engineering (e.g., UML or SysML). They represent the system through a low-level set of states and transition probabilities between them, which obscures the original software-engineering semantics. Direct creation and interpretation of Markov models without any intermediate notation may be uncomfortable and hard to accomplish for software developers, especially when it is to be done repeatedly during the development process.

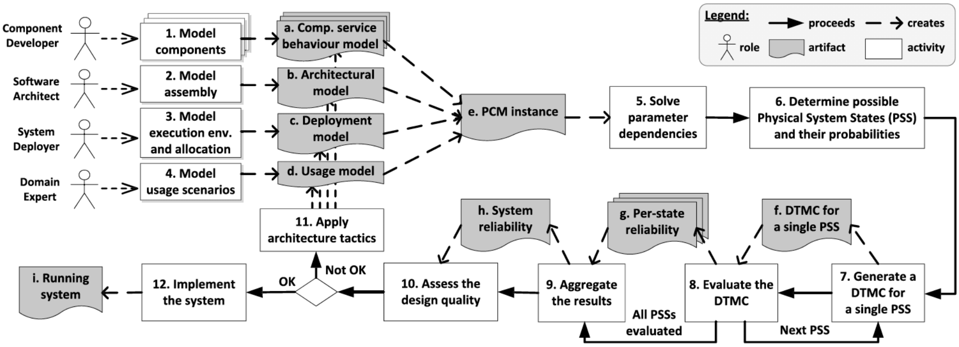

Figure 1: Palladio Component Model Reliability Prediction Approach

Our contribution is a novel technique for architecture-based software reliability modeling and prediction that explicitly considers and integrates the discussed reliability-relevant factors [BKBR12]. The technique offers usage profile separation and propagation through the concept of parameter dependencies [Koz08] and accounts for hardware unavailability through reliability evaluation of service execution under different hardware availability states. We realize the approach as an extension of the Palladio Component Model (PCM) [BKR09], which offers a UML-like modeling notation. We provide tool support for an automated transformation of PCMs into Markov chains and space-effective evaluation of these chains. We discuss how software engineers can use architecture tactics to systematically improve the reliability of the software architecture. Furthermore, we validate the approach in two case studies.

References

[BKBR12] Franz Brosch, Heiko Koziolek, Barbora Buhnova, and Ralf Reussner. Architecture-Based Reliability Prediction with the Palladio Component Model. *IEEE Transactions on Software Engineering*, 38(6):1319–1339, November 2012.

[BKR09] Steffen Becker, Heiko Koziolek, and Ralf Reussner. The Palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 82(1):3–22, January 2009.

[Koz08] Heiko Koziolek. *Parameter Dependencies for Reusable Performance Specifications of Software Components*. PhD thesis, University of Oldenburg, Germany, March 2008.