Towards controlling software architecture erosion through runtime conformance monitoring

View/Open
LakshithadeSilvaPhDThesis.pdf (3.730Mb)

Date
01/12/2014

Author
de Silva, Lakshitha R.

Supervisor
Balasubramaniam, Dharini

Keywords
Software
Architecture
Runtime
Conformance
Monitoring
Dynamic
Compliance
Checking

Metadata
Show full item record

Abstract

The software architecture of a system is often used to guide and constrain its implementation. While the code structure of an initial implementation is likely to conform to its intended architecture, its dynamic properties cannot always be fully checked until deployment. Routine maintenance and changing requirements can also lead to a deployed system deviating from this architecture over time. Dynamic architecture conformance checking plays an important part in ensuring that software architectures and corresponding implementations stay consistent with one another throughout the software lifecycle. However, runtime conformance checking strategies often force changes to the software, demand tight coupling between the monitoring framework and application, impact performance, require manual intervention, and lack flexibility and extensibility, affecting their viability in practice. This thesis presents a dynamic conformance checking framework called PANDArch framework, which aims to address these issues. PANDArch is designed to be automated, pluggable, non-intrusive, performance-centric, extensible and tolerant of incomplete specifications. The thesis describes the concept and design principles behind PANDArch, and its current implementation, which uses an architecture description language to specify architectures and Java as the target language. The framework is evaluated using three open source software products of different types. The results suggest that dynamic architectural conformance checking with the proposed features may be a viable option in practice.

Type
Thesis, PhD Doctor of Philosophy

Collections
Computer Science Theses

URI
http://hdl.handle.net/10023/5220

Eroded software often goes through a process of re-engineering, though this may not always yield the expected benefits. The alternative is to build a replace- ment system from scratch, which clearly would require a sizeable investment. 1. Classification framework of existing methods for controlling architecture erosion. 4. Process-oriented architecture conformance Architecture conformance, which is vital for minimising archi- tecture erosion, is generally achieved through process-centric activities during software development. Architecture compliance monitoring, and. Dependency analysis We discuss the survey results under the above categories in Sections 4.14.4. 136 L. de Silva, D. Balasubramaniam / The Journal of Systems and Software 85 (2012) 132151 4.1. 2012. Controlling software architecture erosion: A survey. Journal of Systems and Software (2012), 132--151. Google Scholar Digital Library. 2008. Towards a Dependency Constraint Language to Manage Software Architectures. In Software Architecture. 256--263. Google Scholar Digital Library. Rainer Weinreich and Georg Buchgeher. 2014. Automatic Reference Architecture Conformance Checking for SOA-Based Software Systems. In 2014 IEEE/IFIP Conference on Software Architecture, WICSA 2014, Sydney, Australia, April 7--11, 2014. 95--104. Thereby, architectural changes can happen at runtime and depend on the state of the components. Thus, analysis of erosion for dynamic architectures requires the analysis of component behavior, and therefore it is only difficult, not to say impossible, to detect with static analysis techniques. Monitoring-Oriented Programming (MOP) [8], is a formal software development and analysis framework for RV. In MOP the developer specifies desired properties, or generates monitors, using specification formalisms. Although different techniques for controlling software architectural erosion have been proposed across the literature, previous work has mainly focused on static analysis methods. The software architecture of a system is often used to guide and constrain its implementation. While the code structure of an initial implementation is likely to conform to its intended architecture, its dynamic properties cannot always be fully checked until deployment. Routine maintenance and changing requirements can also lead to a deployed system deviating from this architecture over time. This thesis presents a dynamic conformance checking framework called PANDArch framework, which aims to address these issues. PANDArch is designed to be automated, pluggable, non-intrusive, performance-centric, extensible and tolerant of incomplete specifications. de Silva, L.: Towards controlling software architecture erosion through runtime conformance monitoring. Dissertation, University of St. Andrews, St. Andrews (2014)Google Scholar. 38. Cite this paper as: Jasser S. (2019) Constraining the Implementation Through Architectural Security Rules: An Expert Study. In: Franch X., Männistö T., Martínez-Fernández S. (eds) Product-Focused Software Process Improvement. PROFES 2019.