Chapter 3
OntoArch Reliability–Aware Software Architecture Design and Experience

Jiehan Zhou
University of Oulu, Finland

Eila Ovaska
VTT Technical Research Centre of Finland, Finland

Antti Evesti
VTT Technical Research Centre of Finland, Finland

Anne Immonen
VTT Technical Research Centre of Finland, Finland

ABSTRACT

Reliability-aware software architecture design has recently been gaining growing attention among software architects. This chapter tackles the issue by proposing an ontology-based, reliability-aware software architecture design and evaluation approach, called OntoArch, which incorporates quantitative reliability evaluation in software architecture design by the means of the OntoArch ontology and the OntoArch tool. The OntoArch approach is characterized by: (1) integration of software reliability engineering and software architecture design; (2) proposing a reliability-aware software architecture design process model; (3) developing the OntoArch ontology in the context of software architecture design and software reliability engineering; and (4) the OntoArch tool not only enabling software architects to design architectures and model reliabilities, but also functioning as a knowledge management platform relying on reliability-aware software architecture design. The OntoArch approach is validated for a software architecture design; for example, Personal Information Repository (PIR), with the use cases of OntoArch-based software architecture knowledge management, software reliability profiling, and software architecture modeling and evaluation.

DOI: 10.4018/978-1-60960-215-4.ch003

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

Reliability evaluation taking place prior to software development has been attracting a growing attention among software architects and reliability experts. In particular, software architecture design has been regarded as the first phase of evaluating reliability in the development of software systems. Software architectural decisions (i.e. architecture design) have a direct impact on such system aspects as cost, time-to-market, and quality. This consideration results in software reliability evaluation in the phase of software architecture design (i.e. software architecture-based reliability evaluation and reliability evaluation at software architecture level) (Goseva-Popstojanova, Mathur, & Trivedi, 2001; Wang, Pan, & Chen, 2006) (Roshandel & Medvidovic, 2004). We use the term of reliability-aware software architecture design and evaluation, which aims to obtain software architecture design in the context of reliability measurement and evaluation.

This paper is based on our previous work (Zhou & Niemelä, et al. 2008) in terms of reliability-aware software architecture, especially extending the scope of the OntoArch ontology design and its usage in PIR prototype implementation. Our contribution is proposing an ontology-based software architecture design and evaluation method, i.e., the OntoArch method. The method integrates software reliability engineering, quality driven software architecture design and quality evaluation approaches. The method embodies reliability engineering in the OntoArch ontology, which is used and exploited in software architecture design and reliability evaluation.

The remainder of the paper is organized as follows. The second section addresses the motivations of creating a reliability-aware software architecture ontology, i.e., the OntoArch ontology. The third section presents the OntoArch engineering process. The fourth section examines knowledge domains of reliability-aware software architecture design and develops the OntoArch ontology. The fifth section presents the experiences on the development of the OntoArch tool and validates the OntoArch method within the PIR system architecture design in the use cases of PIR architecture knowledge management, reliability profiling, architecture modeling, and architecture evaluation. The sixth section reviews related work and outlines our future work. The final section draws a conclusion.

BACKGROUND AND MOTIVATIONS

Traditionally, reliability analysis is performed after system's implementation, when corrections are difficult and modifications are time consuming and expensive. Currently the interest of reliability evaluation has turned to quality evaluation at the software architecture level. Several proposals have been made to predict reliability at the architecture level. In (Immonen & Niemelä, 2007) we compare six most promising reliability evaluation methods, i.e. the methods from Cortellessa et al. (Cortellessa, Singh, & Cukic, 2002), Rodrigues et al. (Oreizy & Taylor et al., 1999), Yacoub et al. (Yacoub, Cukic, & Ammar, 1999), Reussneer et al. (Reussner, Schmidt, & Poernomo, 2003), and Wang et al. (Wang, Pan, & Chen, 1999) from the viewpoint of software architecture. In summary, all the surveyed methods require additional work, especially in the development of the analysis model and application of mathematical algorithms. Although many of these methods have been created for years ago, only few evidences of their maturity are available; i) the methods have been experimented only in laboratory settings by the authors; ii) no comparison of the predicted and measured reliability values exists; and iii) no cost estimation of using the methods could be found. The above mentioned methods are targeted to system architects, software architects, software integrators or service assemblers. It is also common to these methods that traceability of reliability requirements to software architecture is missing.

25 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/ontoarch-reliability-aware-software-architecture/51968

Related Content

Self-Boosted With Dynamic Semi-Supervised Clustering Method for Imbalanced Big Data Classification
Akkala Abhilashaand Annan Naidu P. (2022). *International Journal of Software Innovation (pp. 1-24).*
www.irma-international.org/article/self-boosted-with-dynamic-semi-supervised-clustering-method-for-imbalanced-big-data-classification/297990

How to Systematically Embed Cycles in Balanced Hypercubes
Hsuan-Han Chang, Kuan-Ting Chenand Pao-Lien Lai (2017). *International Journal of Software Innovation (pp. 44-56).*
www.irma-international.org/article/how-to-systematically-embed-cycles-in-balanced-hypercubes/169917

Design and Transformation of a Domain-Specific Language for Reconfigurable Conveyor Systems
Kyoungho An, Adam Trewyn, Aniruddha Gokhaleand Shivakumar Sastry (2013). *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments (pp. 553-571).*
www.irma-international.org/chapter/design-transformation-domain-specific-language/71832

Wireless Sensor Network as Enabling Technology for Cyber-Physical System
Jiwa Abdullahand Nayef Abdulwahab Alduais (2018). *Cyber-Physical Systems for Next-Generation Networks (pp. 26-53).*
www.irma-international.org/chapter/wireless-sensor-network-as-enabling-technology-for-cyber-physical-system/204666

Towards a UMLsec-Based Proctored Examination Model
Ibukun Fadahunsi, Oluwasefunmi 'Tale Arogundade, Adesina S. Sodiyaand Bakai Olajuwon (2019). *International Journal of Systems and Software Security and Protection (pp. 44-67).*
www.irma-international.org/article/towards-a-umlsec-based-proctored-examination-model/247491