Chapter 6
Balancing Security and Performance Properties During System Architectural Design

Siv Hilde Houmb
Telenor GBDR Platform for Service Innovation Group, Norway

Geri Georg
Colorado State University, USA

Dorina C. Petriu
Carleton University, Canada

Behzad Bordbar
University of Birmingham, UK

Indrakshi Ray
Colorado State University, USA

Kyriakos Anastasakis
University of Birmingham, UK

Robert B. France
Colorado State University, USA

ABSTRACT

Developers of critical systems need to address several quality properties, such as security and performance, in the early stages of the development cycle to ensure that the system under construction meets its requirements. Sometimes quality properties conflict with each other and/or with the system’s functionalities, so the developers need to make trade-off decisions. Unreasonable costs, added developer resources and tight project schedules may be other reasons for having to trade-off between alternative solutions. In the context of Model-Driven Development, the analysis of quality properties is done by

DOI: 10.4018/978-1-61520-837-1.ch006
Balancing Security and Performance Properties During System Architectural Design

INTRODUCTION

Developers of critical systems need to address several quality properties in the early stages of the development cycle, to ensure that the system under construction meets its requirements. Basing design decisions on quality analysis results early during system design can help avoid drastic and expensive changes in later development stages. In the context of Model-Driven Development, the analysis of quality properties is done by transforming software design models into different analysis models based on various formalisms, which are then analyzed with existing tools. The challenge consists in integrating different models, transformations and tools into a consistent and coherent process. In this chapter the authors present a methodology called Aspect-Oriented Risk Driven Development (AORDD), which integrates the analysis of two quality properties, namely security and performance, into the development process of critical systems. Each quality property is analyzed separately, and then all results are input to a trade-off analysis that identifies conflicts between the properties. Trade-off analysis aims at supporting designers and developers in choosing the security and performance solutions that best fit their needs, without introducing unacceptable development delays or costs. The security analysis consists of identifying the assets (critical components, such as sensitive information) of an application and the attacks that can compromise these assets, and formally analyzing whether these attacks are actually possible using the tools UML2Alloy and Alloy Analyzer. If the system is vulnerable to the attack, some security solution, modeled as an aspect according to Aspect Oriented Modeling (AOM), is added to the system. The analysis must be repeated to ensure that the resulting system is secure. Performance analysis is accomplished using Layered Queuing Network (LQN) models. Annotated system models are transformed into LQN models and performance experiments are executed on them. If the performance results are unacceptable, the system design has to be changed and the analysis repeated. Finally, the results of the security and performance analysis are input to the system quality property trade-off analysis, which is implemented as a Bayesian Belief Network (BBN) topology, and which also takes as input external parameters, such as time to market and budget constraints. The results of the trade-off analysis help identify how well a particular design meets performance, security and other project goals, which, in turn, can guide the developer in making informed design decisions. The approach is illustrated using a transactional web e-commerce benchmark (TPC-W) originally developed by the Transaction Processing Performance Council.

transforming software design models into different analysis models based on various formalisms, which are then analyzed with existing tools. A major challenge is to integrate different models, transformations and tools into a consistent and coherent process. In this chapter the authors present a methodology called Aspect-Oriented Risk Driven Development (AORDD), which integrates the analysis of two quality properties, namely security and performance, into the development process of critical systems. Each quality property is analyzed separately, and then all results are input to a trade-off analysis that identifies conflicts between the properties. Trade-off analysis aims at supporting designers and developers in choosing the security and performance solutions that best fit their needs, without introducing unacceptable development delays or costs. The security analysis consists of identifying the assets (critical components, such as sensitive information) of an application and the attacks that can compromise these assets, and formally analyzing whether these attacks are actually possible using the tools UML2Alloy and Alloy Analyzer. If the system is vulnerable to the attack, some security solution, modeled as an aspect according to Aspect Oriented Modeling (AOM), is added to the system. The analysis must be repeated to ensure that the resulting system is secure. Performance analysis is accomplished using Layered Queuing Network (LQN) models. Annotated system models are transformed into LQN models and performance experiments are executed on them. If the performance results are unacceptable, the system design has to be changed and the analysis repeated. Finally, the results of the security and performance analysis are input to the system quality property trade-off analysis, which is implemented as a Bayesian Belief Network (BBN) topology, and which also takes as input external parameters, such as time to market and budget constraints. The results of the trade-off analysis help identify how well a particular design meets performance, security and other project goals, which, in turn, can guide the developer in making informed design decisions. The approach is illustrated using a transactional web e-commerce benchmark (TPC-W) originally developed by the Transaction Processing Performance Council.

INTRODUCTION

Developers of critical systems need to address several quality properties in the early stages of the development cycle, to ensure that the system under construction meets its requirements. Basing design decisions on quality analysis results early during system design can help avoid drastic and expensive changes in later development stages. In the context of Model-Driven Development, the analysis of quality properties is done by transforming software design models into different analysis models based on various formalisms, which are then analyzed with existing tools. The challenge consists in integrating different models, transformations and tools into a coherent development process.

Sometimes quality properties conflict with each other and/or with the system’s functional-
35 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage: www.igi-global.com/chapter/balancing-security-performance-properties-during/48409?camid=4v1


Related Content

Process Evolution in a Distributed Process Execution Environment
www.igi-global.com/article/process-evolution-distributed-process-execution/80245?camid=4v1a

Facilitating eLearning with Social Software: Attitudes and Usage from the Student's Point of View
www.igi-global.com/chapter/facilitating-elearning-social-software/29490?camid=4v1a

Architecting Virtual Reality Systems
www.igi-global.com/chapter/architecting-virtual-reality-systems/8238?camid=4v1a

FTT: A System to Refactor Traditional Forms into Ajax-Enabled Forms
Ming Ying and James Miller (2011). International Journal of Systems and Service-Oriented Engineering (pp. 1-20).
www.igi-global.com/article/ftt-system-refactor-traditional-forms/61313?camid=4v1a