



Luiz Rodolfo Neves Caldeira

Geração semi-automática de massas de testes funcionais a partir da composição de casos de uso e tabelas de decisão

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio.

Orientador: Arndt von Staa

Rio de Janeiro
Agosto de 2010



Luiz Rodolfo Neves Caldeira

**Geração semi-automática de massas de
testes funcionais a partir da composição
de casos de uso e tabelas de decisão**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Arndt von Staa

Orientador

Departamento de Informática – PUC-Rio

Prof. Carlos José Pereira de Lucena

Departamento de Informática – PUC-Rio

Prof. Alessandro Fabricio Garcia

Departamento de Informática – PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 17 de agosto de 2010

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Luiz Rodolfo Neves Caldeira

Graduou-se em Engenharia de Computação na Pontifícia Universidade Católica do Rio de Janeiro (Brasil, Rio de Janeiro).

Ficha Catalográfica

Caldeira, Luiz Rodolfo Neves

Geração semi-automática de massas de testes funcionais a partir da composição de casos de uso e tabelas de decisão / Luiz Rodolfo Neves Caldeira ; orientador: Staa, Arndt von. – 2010.

98 f. : il. (col.) ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2010.

Inclui bibliografia

1. Informática – Teses. 2. Engenharia de software. 3. Caso de uso. 4. Tabela de decisão. 5. Teste automatizado. 6. Geração semi-automática de casos de teste. I. Staa, Arndt von. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para meus pais, João Luiz e Edilamar.

Agradecimentos

Aos meus pais, João Luiz Caldeira e Edilamar Maciel Neves Caldeira, pelo apoio, educação, carinho e dedicação.

A Lorena Melo Viana da Costa, pelo apoio e compreensão.

Ao meu orientador, prof. Arndt von Staa, por sua dedicação, ensinamentos e orientação.

À Faperj, Capes e PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos professores da Comissão examinadora.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

A todos os amigos e familiares que de alguma forma contribuíram para a realização deste trabalho.

Resumo

Caldeira, Luiz Rodolfo Neves; Staa, Arndt von. **Geração semi-automática de massas de testes funcionais a partir da composição de casos de uso e tabelas de decisão.** Rio de Janeiro, 2010. 98p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho tem por objetivo propor um processo e ferramentas para a geração semi-automática de scripts de teste funcional para sistemas web, a partir de casos de uso e tabelas de decisão, com vistas à geração de eficazes scripts de teste automatizados, simultaneamente reduzindo o tempo gasto para gerá-los. As especificações dos testes dão-se através de descrições de casos de uso escritas usando português restrito semi-estruturado e obedecendo a uma estrutura precisamente definida. Com o auxílio de uma ferramenta, monta-se manualmente uma tabela de decisão a partir desses casos de uso. Os casos de teste semânticos são gerados automaticamente a partir destas tabelas de decisão. Outra ferramenta é responsável por gerar os scripts de testes a partir dos casos de teste semânticos. Os scripts de teste gerados devem adequar-se à ferramenta de execução automatizada dos testes. Neste trabalho, utilizou-se a ferramenta Selenium para a automação da interação com o navegador. A avaliação da eficácia deu-se através da aplicação do processo em um sistema real e através de comparação com técnicas tradicionais de geração de testes automatizados aplicadas ao mesmo sistema.

Palavras-chave

Engenharia de software; Caso de uso; Tabela de decisão; Teste automatizado; Geração semi-automática de casos de teste.

Abstract

Caldeira, Luiz Rodolfo Neves; Staa, Arndt von. **Semi-automatic generation of functional test scripts by composing use cases with decision tables.** Rio de Janeiro, 2010. 98p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This work aims at developing a process and tools for the semi-automatic generation of functional test scripts for web based systems. The process and tools depart from use cases and decision tables, in order to produce high quality automated tests as well as to reduce the time spent generating them. The tests specifications are provided by use cases written in semi-structured Portuguese and obeying a well defined structure. By means of a tool, decision tables are manually built from the use case descriptions. Afterwards semantic test cases are automatically generated from these decision tables. Another tool generates executable test scripts from these test cases. The generated test scripts must suit the tool used for automated testing. In this work, the Selenium tool was used for automating test interaction with the browser. The evaluation of the efficacy of the process and tools was performed applying them to a real system and comparing the result with traditional techniques of automated test generation regarding this same system.

Keywords

Software engineering; Use case; Decision table; Automated test; Semi-automatic test case generation.

Sumário

1	Introdução	13
1.1.	Objetivos	15
2	Estado da arte	18
2.1.	Desenvolvimento dirigido por comportamentos	18
2.1.1.	<i>Frameworks</i> de testes em DDC	20
2.1.1.1.	JBehave	20
2.1.1.2.	RSpec	22
2.2.	Geração de testes a partir de “ <i>capture and replay</i> ”	23
2.3.	Geração de testes a partir de casos de uso	24
2.4.	Geração de testes a partir de tabelas de decisão	25
2.5.	Limitações das propostas publicadas	27
3	Processo	29
3.1.	Redação da descrição dos casos de uso	30
3.2.	Criação da tabela de decisão	39
3.3.	Geração automática dos casos de teste	41
3.4.	Geração automática dos scripts de teste	42
4	Ferramentas	43
4.1.	Editor de tabela de decisão	43
4.1.1.	Edição da tabela	44
4.1.1.1.	Condições	44
4.1.1.2.	Ações	51
4.1.1.3.	Regras	52
4.1.1.4.	Pré-condição	54
4.1.2.	Validação da tabela	56
4.1.3.	Persistência da tabela	57
4.1.4.	Geração dos casos de teste	60
4.1.4.1.	Persistência dos casos de teste	60
4.1.5.	Diagrama de classes	63
4.2.	Gerador de scripts de teste	65

4.2.1. Diagrama de classes	66
5 Estudo de caso	69
5.1. Visão geral sobre a aplicação alvo dos testes	69
5.2. Aplicação do processo	69
5.2.1. Exemplo “Efetuar autenticação”	72
5.2.2. Resultados e considerações	78
5.2.2.1. Comparação com testes existentes produzidos com Squish for Web	81
5.2.2.2. Comparação com testes produzidos manualmente	83
5.2.2.3. Comparação com a abordagem “ <i>capture and replay</i> ”	84
5.2.2.4. Comparação com testes produzidos por desenvolvimento dirigido por comportamento	87
5.2.2.5. Dificuldades encontradas	89
6 Conclusão	92
7 Trabalhos futuros	94
8 Referências bibliográficas	96

Lista de figuras

Figura 1 - Exemplo de cenário em DDC.	19
Figura 2 - Exemplo de estória de usuário em DDC.	19
Figura 3 - Exemplo de cenário em JBehave.	20
Figura 4 - Classe em Java que implementa um cenário JBehave.	21
Figura 5 - Classe que implementa os passos descritos em um cenário JBehave.	21
Figura 6 - Teste em RSpec.	22
Figura 7 - Saída das especificações testadas em RSpec.	23
Figura 8 – Interface Selenium IDE.	24
Figura 9 – Visão geral do processo.	29
Figura 10 – Realce do processo de redação do caso de uso.	30
Figura 11 - Padrão para passo de fluxo.	33
Figura 12 - Exemplo de tela de autenticação.	34
Figura 13 – Fluxo principal do caso de uso Realizar Autenticação.	34
Figura 14 – Restrições dos campos do caso de uso Realizar Autenticação.	34
Figura 15 – Fluxos alternativos do caso de uso Realizar Autenticação.	35
Figura 16 – Primeira parte do caso de uso Cadastrar Sensor.	36
Figura 17 – Segunda parte do caso de uso Cadastrar Sensor.	38
Figura 18 - Realce do processo de criação da tabela de decisão.	39
Figura 19 - Realce do processo de geração automática de casos de teste semânticos.	41
Figura 20 - Realce do processo de geração automática do script de teste.	42
Figura 21 – Tela principal do Editor de tabela de decisão.	43
Figura 22 – Elementos constituintes de uma condição.	44
Figura 23 – condições com mais de um grupo condicional.	49
Figura 24 – grupo condicional com condição principal.	49
Figura 25 – Tela de adição de grupo condicional sem condição principal.	50
Figura 26 – Tela de adição de grupo condicional com condição	

principal.	50
Figura 27 – configuração de arquivo de ações.	51
Figura 28 – Esquema de dependência entre o arquivo de teste gerado e o arquivo de oráculos.	52
Figura 29 – Células de regra para condição do tipo Texto.	53
Figura 30 – Células de regra para condição que não é do tipo Texto.	53
Figura 31 – Diagrama de estado para o caso hipotético.	54
Figura 32 – Campo pré-condição com teste.	55
Figura 33 - Campo pré-condição com URL.	55
Figura 34 – Estrutura do XML de uma tabela de decisão.	57
Figura 35 - Estrutura do XML de uma <i>suite</i> de testes derivada da tabela de decisão.	61
Figura 36 – Diagrama de classes dos pacotes tableeditor e tableeditor.control.	63
Figura 37 – Diagrama de classes do pacote tableeditor.model.	64
Figura 38 - Diagrama de classes do pacote tableeditor.view.	65
Figura 39 - Diagrama de classes do pacote scriptwriter.model.	66
Figura 40 - Diagrama de classes dos pacotes scriptwriter e scriptwriter.control.	67
Figura 41 – Diagrama de estado dos casos de uso do sistema alvo.	71
Figura 42 – Tela para “Efetuar autenticação”.	72
Figura 43 – Descrição de caso de uso para “Efetuar autenticação”.	73
Figura 44 – Primeira parte da tabela de decisão para “Efetuar autenticação”.	74
Figura 45 – Segunda parte da tabela de decisão para “Efetuar autenticação”.	74
Figura 46 – Teste gerado automaticamente para “Efetuar autenticação”.77	
Figura 47 – Oráculos para teste “Efetuar autenticação”.	78

Lista de tabelas

Tabela 1 – Exemplo de tabela de decisão.	25
Tabela 2 – Layout de formulário de caso de uso.	31
Tabela 3 – Padrões para tipos de elementos.	33
Tabela 4 – Tempo gasto para a geração dos testes a partir da tabela de decisão.	79
Tabela 5 – Número de casos de teste por funcionalidade.	80
Tabela 6 – Comparação de número de casos de teste gerados.	81
Tabela 7 - Comparação com tempo gasto para geração manual.	84
Tabela 8 – Comparação com tempo gasto para geração com <i>capture and replay</i> .	86
Tabela 9 - Comparação com tempo gasto para geração com <i>JBehave</i> .	88
Tabela 10 – Segunda comparação com tempo gasto para geração com <i>JBehave</i> .	89