

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Eduardo Kinder Almentero

**Re-engenharia do software C&L para plataforma
Lua-Kepler utilizando princípios de transparência**

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: Julio Cesar Sampaio do Prado Leite

Rio de Janeiro, 08 abril de 2009

Eduardo Kinder Almentero

**Re-engenharia do software C&L para plataforma
Lua-Kepler utilizando princípios de transparência**

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. Julio Cesar Sampaio do Prado Leite
Orientador
Departamento de Informática – PUC-Rio

Prof. Carlos José Pereira de Lucena
Departamento de Informática – PUC-Rio

Prof. Roberto Ierusalimschy
Departamento de Informática – PUC-Rio

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 08 de abril de 2009

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

Eduardo Kinder Almentero

Graduou-se em Bacharelado em Informática na Universidade do Estado do Rio de Janeiro (UERJ) em 2007. Desde 2007 participa do grupo de pesquisa de engenharia de requisitos da PUC-Rio, coordenado pelo professor Julio Leite, e trabalha no Laboratório de Engenharia de Software (LES).

Ficha Catalográfica

Almentero, Eduardo Kinder

Re-engenharia do software C&L para plataforma Lua-Kepler utilizando princípios de transparência / Eduardo Kinder Almentero ; orientador: Julio Cesar Sampaio do Prado Leite. – 2009.

112 f. ; 30 cm

Dissertação (Mestrado em Informática)–Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

Inclui bibliografia.

1. Informática – Teses. 2. Engenharia de Requisitos. 3. Transparência de Software. 4. Cenário. 5. Léxico Ampliado da Linguagem. I. Leite, Julio Cesar Sampaio do Prado. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

A meus pais, Emilio e Meire, pelo incentivo, motivação e carinho. Sem vocês nada disto seria possível.

Ao professor Julio Leite pela enorme paciência, dedicação, confiança e pelo conhecimento compartilhado ao longo destes anos.

Aos integrantes do Grupo de Requisitos da PUC-Rio, Antonio de Padua, Maurício, Claudia, Fillipe, Herbet, Vera e Elizabeth, pelas suas valiosíssimas opiniões e sugestões sempre construtivas.

Aos amigos, Andrew, Evelin, Camila, Isela, Baldoino, Elder e Manoel, pela ajuda indispensável, pela motivação e pelo companheirismo durante toda essa jornada.

Resumo

Almentero, Eduardo Kinder; Leite, Julio Cesar Sampaio do Prado. **Re-engenharia do software C&L para plataforma Lua-Kepler utilizando princípios de transparência.** Pontifícia Universidade Católica do Rio de Janeiro, 2009. 112p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A transparência é um termo chave que está presente em diversos contextos como o econômico e político, e atualmente um dos novos contextos em que se apresenta é a transparência de software. Software livre é um bom exemplo de transparência de software, onde a grande vantagem é que podemos acessar o código fonte e então escolher entre suas características as que desejamos, porém esta possibilidade está direcionada somente aqueles que entendem seu código fonte. Entender o código fonte de um software poder ser uma tarefa árdua, especialmente se nenhuma técnica foi utilizada para facilitar sua leitura. Neste trabalho exploramos um método de desenvolvimento para software livre baseado no uso de cenários. O resultado da aplicação deste método será um documento único, o código fonte, onde teremos os cenários integrados com o código, facilitando sua leitura e entendimento, trazendo assim mais transparência para o software. Este método foi refinado durante sua aplicação na re-engenharia do software C&L. Para produzir uma documentação complementar aos cenários inclusos no código fonte, utilizamos a técnica LAL (Léxico Ampliado da Linguagem) para mapear o espaço de nomes do novo software C&L.

Palavras-chave

Engenharia de requisitos, Transparência de software, Cenário, Léxico Ampliado da Linguagem.

Abstract

Almentero, Eduardo Kinder; Leite, Julio Cesar Sampaio do Prado (Advisor). **Reengineering the C&L software towards Lua-Kepler platform using principles of transparency.** Pontifícia Universidade Católica do Rio de Janeiro, 2009. 112p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Transparency is a keyword present in different contexts such as the economic and the political ones, and, currently, one of the new contexts, in which it stands, is software. Free (open source) software is a good example of transparency, where the great advantage is that one can access the source code and then choose the characteristics he/she wants, but, in this way, we will be serving only those who understand the source code. Understanding software source code can be an arduous task, especially if no technique has been used for facilitate reading. In this work we explore a method for developing free software based on the use of scenarios. The result of applying this method is a single document, the source code, in which the scenarios will be integrated within the code, making it easier to read and understand, thus bringing more transparency to the software. This method was refined during its application to reengineer the “C&L” software. In order to produce additional documentation, besides the scenarios embedded in the code, we used the LEL (Language Extended Lexicon) technique to map the namespace of the new “C&L”.

Keywords

Requirements Engineering, Software Transparency, Scenario, Language Extended Lexicon.

Sumário

1 Introdução	12
1.1. Definições do Problema	12
1.2. Abordagem Proposta	13
1.3. Trabalhos relacionados	14
1.4. Organização do Documento	15
2 Representações utilizadas	16
2.1. Léxico Ampliado da Linguagem (LAL)	16
2.2. Cenários	20
3 Software C&L	25
3.1. Introdução	25
3.2. Arquitetura	26
3.3. Funcionamento	27
4 Construindo a nova arquitetura	38
4.1. Método de desenvolvimento	38
4.1.1. Construção do LAL	39
4.1.2. Descrição das situações do sistema através de cenários	39
4.1.3. Integração dos cenários	41
4.1.4. Refinamento dos cenários	46
4.2. Mapeamento do espaço de nomes	59
4.2.1. Modelo para inclusão de variável	60
4.2.2. Modelo para inclusão de função	61
4.2.3. Modelo para inclusão de arquivo	62
5 Re-arquitetura do software C&L	64
5.1. Arquitetura do novo C&L e tecnologias utilizadas	64
5.2. Aplicação do método na construção do novo C&L	65
5.2.1. Descrição das situações do sistema através de cenários	66
5.2.2. Divisão dos cenários em grupos	77
5.2.3. Identificar os cenários raiz	82
5.2.4. Construir cenário integrador	85

5.2.5. Refinamento dos cenários	86
5.2.6. Operacionalização dos cenários	91
6 Conclusão	104
6.1. Comparação com trabalhos relacionados	105
6.2. Dificuldades encontradas	106
6.3. Trabalhos Futuros	107
Referências bibliográficas	108

Lista de figuras

Figura 1 – Modelo entidade relacionamento do LAL.	18
Figura 2 – Exemplo de um símbolo do LAL [Leite98].	19
Figura 3 – Exemplo de um grafo de relacionamentos.	19
Figura 4 – Modelo entidade relacionamento para um cenário.	21
Figura 5 – Exemplo de cenário.	23
Figura 6 – Grafo de relacionamentos entre cenários e símbolos do léxico.	24
Figura 7 – Mapa do relacionamento entre arquivos [C&L09].	26
Figura 8 – Página inicial do sistema.	28
Figura 9 – Tela de login do sistema.	28
Figura 10 – Formulário de cadastro de usuário.	29
Figura 11 – Página principal do sistema.	30
Figura 12 – Formulários de inclusão de cenário e símbolo do léxico.	31
Figura 13 – Exemplos de atalhos criados pelo C&L.	32
Figura 14 – Menu “info”.	33
Figura 15 – Tela para adicionar colaboradores ao projeto.	34
Figura 16 – Exemplo de visualização do código.	35
Figura 17 – Exemplo de XML gerado pelo C&L.	36
Figura 18 – Exemplo de grafo gerado pelo C&L.	37
Figura 19 – Exemplo de foco no grafo gerado pelo C&L.	37
Figura 20 – Exemplo de uma entrada do LAL	39
Figura 21 – Exemplo de descrição de uma situação do sistema.	41
Figura 22 – Processo de integração de cenários.	42
Figura 23 – Exemplo de um grupo de cenários.	43
Figura 24 – Exemplo de cenário raiz.	45
Figura 25 – Exemplo de cenário integrador.	46
Figura 26 – Processo de refinamento dos cenários em camadas.	47
Figura 27 – Exemplo de cenário da camada de visão.	49
Figura 28 – Exemplo de cenário da camada de modelo.	50
Figura 29 – Exemplo de cenário da camada de controle.	52
Figura 30 – Trecho de uma função <i>JavaScript</i> documentada através de um cenário.	54
Figura 31 – Trecho de cenário da camada de visão operacionalizado	55

Figura 32 – Trecho de cenário da camada de controle operacionalizado.	56
Figura 33 – Cenário da camada modelo operacionalizado.	57
Figura 34 – Cenário que descreve acesso ao banco de dados implementado.	58
Figura 35 – Exemplo de variável descrita através do LAL.	61
Figura 36 – Exemplo de função mapeada através do LAL.	62
Figura 37 – Exemplo de arquivo mapeado através do LAL.	63
Figura 38 – Arquitetura do novo C&L.	65
Figura 39 – Grupo correspondente ao módulo usuário.	78
Figura 40 – Grupo correspondente ao módulo projeto.	79
Figura 41 – Grupo correspondente ao módulo administração de colaboradores.	80
Figura 42 – Grupo correspondente ao módulo léxico.	81
Figura 43 – Grupo correspondente ao módulo cenário.	82
Figura 44 – Relacionamentos entre os cenários do grupo correspondente ao módulo usuário.	83
Figura 45 – Relacionamentos entre os cenários do grupo correspondente ao módulo projeto.	83
Figura 46 – Relacionamentos entre os cenários do grupo correspondente ao módulo colaboradores.	84
Figura 47 – Relacionamentos entre os cenários do grupo correspondente ao módulo léxico.	84
Figura 48 – Relacionamentos entre os cenários do grupo correspondente ao módulo cenário.	85
Figura 49 – Relacionamentos entre os cenários raiz.	85
Figura 50 – Cenário integrador do software C&L.	86
Figura 51 – Exemplo de texto sendo visualizado.	92
Figura 52 – Exemplo de texto com o elo software.	92
Figura 53 – Exemplo de texto com o elo engenharia de software.	93
Figura 54 – Exemplo de texto sendo visualizado.	93
Figura 55 – Vetor de identificadores ordenado.	93
Figura 56 – Passos da aplicação do algoritmo antigo.	94
Figura 57 – Passos da aplicação do algoritmo novo.	96
Figura 58 – Montagem dos elos do algoritmo novo.	97
Figura 59 – Relacionamentos entre os cenários do algoritmo.	98

Lista de tabelas

Tabela 1 – Descrição dos tipos de entrada do LAL.	17
Tabela 2 – Representação de cenários adotada.	40
Tabela 3 – Heurísticas para construção de um cenário da camada de visão.	48
Tabela 4 – Heurísticas para construção de um cenário da camada de modelo.	50
Tabela 5 – Heurísticas para construção de um cenário da camada de controle.	51
Tabela 6 – Heurísticas para descrever funções <i>JavaScript</i>	54
Tabela 7 – Heurísticas para criação de cenários que descrevem acessos ao banco de dados.	58
Tabela 8 – Descrição de uma variável através do LAL.	60
Tabela 9 – Descrição de uma função através do LAL.	61
Tabela 10 – Descrição de um arquivo através do LAL.	62