

**Ulf Bergmann**

**Evolução de Cenários Através de um Mecanismo de  
Rastreamento Baseado em Transformações**

**Tese de Doutorado**

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

Orientador: Julio Cesar Sampaio do Prado Leite

Rio de Janeiro - RJ, maio de 2003

**Ulf Bergmann**

**Evolução de Cenários através de um Mecanismo de Rastreamento Baseado em Transformações**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico 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. Arndt von Staa**

Departamento de Informática – PUC-Rio

**Prof. Antonio Francisco do Prado**

Departamento de Computação - UFSCar

**Profa. Claudia Maria Lima Werner**

COPPE – UFRJ

**Profa. Karin Koogan Breitman**

Departamento de Informática – PUC-Rio

**Prof. Christiano de Oliveira Braga**

UFF

**Prof. Ney Dumont**

Coordenador Setorial do Centro Técnico Científico

Rio de Janeiro, 18 de dezembro de 2002

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

### **Ulf Bergmann**

É graduado em Engenharia de Computação e Mestre em Sistemas e Computação pelo IME. Trabalhou como consultor e foi mentor de projetos de transferência de tecnologia de orientação a objetos. Trabalhou no desenvolvimento de Jogos de Guerra e de Sistemas de Comando e Controle para o Exército Brasileiro. Na área acadêmica foi professor de diversas instituições de ensino superior e possui vários artigos publicados em eventos científicos.

#### Ficha Catalográfica

Bergmann, Ulf

Evolução de cenários através de um mecanismo de rastreamento baseado em transformações / Ulf Bergmann; orientador: Julio Cesar Sampaio do Prado Leite. – Rio de Janeiro : PUC, Departamento de Informática, 2003.

[13], 211 f, : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

1. Informática – Teses. 2. Rastreamento. 3. Sistemas transformacionais. 4. Evolução de software. I. Leite, Julio Cesar Sampaio do Prado. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para minha esposa, Ana Christina, pelo amor e estímulo constantes, e para meus  
filhos, Pedro Henrique, Ana Beatriz e Bruna.

## Agradecimentos

Ao meu orientador Professor Julio César Sampaio do Prado Leite, pela orientação segura, estímulo constante e exemplo de profissional que, tenho certeza, irá nortear meus caminhos profissionais.

À minha esposa e meus filhos, pela compreensão de que a busca por um sonho impõe sacrifícios e obstáculos a serem vencidos.

Aos meus pais, pela educação, carinho e exemplo de vida.

Ao Exército Brasileiro, pela oportunidade oferecida para ampliar meus conhecimentos.

Ao CNPq, pelo auxílio concedido e que se mostrou fundamental para a execução deste trabalho.

À todas as pessoas que emprestaram seu nome ao Departamento de Informática da PUC-Rio durante a realização deste trabalho, pelo apoio constante e amigo.

A todos os familiares e amigos, pelo estímulo e ajuda.

A DEUS, por me conceder os atributos necessários à execução deste trabalho, bem como a saúde e disposição necessárias.

## Resumo

Bergmann, Ulf. **Evolução de Cenários Através de um Mecanismo de Rastreamento Baseado em Transformações**. Rio de Janeiro - RJ, 2003. 224p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A importância da rastreabilidade na evolução de sistemas é um fato amplamente reconhecido, mas apesar dos esforços despendidos até agora, ainda encontramos vários obstáculos a sua efetiva utilização. Esses obstáculos são relacionados a geração e validação das informações sobre os relacionamentos e interdependências entre o mundo real, os requisitos e o sistema de software. Estas informações freqüentemente não são confiáveis devido a inconsistência causada pela evolução distinta dos vários artefatos resultantes do processo de desenvolvimento de um sistema.

Esta tese tem como objetivo estudar a utilização da tecnologia transformacional na implementação de um mecanismo de rastreamento de maneira a se obter a atualização automática das informações de rastreamento, através da identificação das mudanças ocorridas entre versões consecutivas dos artefatos produzidos e o seu subsequente armazenamento na forma de transformações. Os benefícios obtidos pelo uso deste mecanismo permitirão: a eliminação dos problemas da falta de atualização das informações de rastreamento e o conseqüente aumento do valor destas informações para o desenvolvedor; a sistematização do processo de rastreamento; e a diminuição do custo para se obter, atualizar e validar as informações de rastreamento. Além destes benefícios, o armazenamento das informações de rastreamento na forma de transformações incorpora um maior conhecimento sobre a modelagem realizada, ao contrário de outros mecanismos que armazenam somente os artefatos e seus inter-relacionamentos estáticos.

## Palavras-chave

Rastreamento; sistemas transformacionais; Evolução de Software

## **Abstract**

Bergmann, Ulf. **Evolução de Cenários Através de um Mecanismo de Rastreamento Baseado em Transformações**. Rio de Janeiro - RJ, 2003. 224p. Tese de Doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The importance of traceability in software evolutions was quite established, but despite the effort spent, they are many issues that difficult they effective use. These issues are related to the generating and validating of trace information about the relationships and interdependencies between the real world, the requirements and the software system. This information is often mistrusted since they may have become obsolete due to separate evolution of models and systems.

This thesis study the use of the transformational technology in the implementation of a traceability mechanism that automatically obtain trace information by identifying the changes made in consecutive artifact versions and store it as transformations. The main advantages gained with it uses are the elimination of the problems of the lack of trace information updating, the systematization of the trace process and the decrease of the cost to obtain, update and validate trace information. Besides these benefits, the storage of trace information in the form of transformations incorporates a larger knowledge on the accomplished modeling, unlike other mechanisms that only store the artifacts and their static inter-relationships.

## **Keywords**

Traceability; Transformation System; Software Evolution

## Sumário

1	Introdução	14
1.1.	Motivação	14
1.2.	Objetivos	16
1.3.	Benefícios Esperados	16
1.4.	Revisão da Literatura	17
1.4.1.	Análise Léxica	17
1.4.2.	Análise Sintática	18
1.4.3.	Análise Dinâmica de Sistemas	19
1.4.4.	Reestruturação de Programas	20
1.4.4.1.	Identificação de Tipos Abstratos de Dados (ADT)	21
1.4.4.2.	Decomposição	22
1.4.5.	Técnicas Baseadas em Conhecimento	23
1.4.6.	Métodos Formais	24
1.4.7.	Técnicas Manuais	25
1.4.8.	Paradigma Transformacional	26
1.5.	Organização da Tese	27
2	Sistema Transformacional Draco-PUC	28
2.1.	Rede de Domínios	34
2.1.1.	Implementação de Rede de Domínios usando a Máquina Draco-PUC	36
2.1.2.	Implementação de Rede de Domínios usando XSL	38
2.1.3.	Rede de Domínios Integrada Draco-PUC – XSL	40
2.2.	Ferramentas Desenvolvidas	44
2.3.	Conclusão	46
3	Cenários	47
3.1.	Modelo de Cenário	48
3.2.	Evolução de Cenários	51
3.3.	Rastreamento de Cenários	54

3.3.1. Classificação das Ligações de Rastreamento	55
3.3.1.1. Modelo de Referência de Ramesh & Jarke	56
3.3.1.2. Classificação quanto ao conteúdo proposta por Hughes & Martin	58
3.3.1.3. Classificação quanto ao sentido proposta por Gotel e Finkelstein	59
3.3.2. Requisitos de um sistema de rastreamento	60
4 Mecanismo de Rastreamento Baseado em Transformações	63
4.1. Representação dos Artefatos	65
4.2. Geração das Transformações de Rastreamento	69
4.2.1. Identificação de Diferenças	70
4.2.1.1. Algoritmo de Identificação de Diferenças ( <i>TreeDiff</i> )	70
4.2.1.2. Descrição das Diferenças	75
4.2.1.3. Geração dos Fatos Observados sobre as Diferenças	83
4.2.2. Geração das Transformações	84
4.2.3. Reconhecimento de Planos	87
4.3. Refinamento de Modelos	97
4.4. Conclusão	99
4.4.1. Análise da Satisfação dos Requisitos	100
5 Uso do Mecanismo de Rastreamento Baseado em Transformações na Evolução de Cenários	102
5.1. Descrição do Sistema	103
5.1.1. Identificação de Relacionamentos	104
5.1.2. Identificação das Operações	112
5.2. Protótipo da Ferramenta de Suporte ao Processo	116
5.3. Estudo de Caso	119
5.3.1. Exemplo	120
5.3.2. Resultados Obtidos	126
5.4. Conclusão	127
6 Conclusão	132
6.1. Contribuições da Tese	132
6.2. Comparação com Propostas existentes	134
6.3. Trabalhos Futuros	136

Referências Bibliográficas	138
Apêndice A - Definição de Tipos de Documentos (DTD) Utilizados	146
A.1. Léxico Ampliado da Linguagem (LAL)	146
A.2. Cenários	147
Apêndice B - Gramáticas dos Domínios Draco-Puc	148
B.1. Domínio Draco-Puc para definição de DTDs	148
B.2. Léxico Ampliado da Linguagem (LAL)	150
B.3. Cenários	152
B.4. Domínio de representação de Diferenças (Diff)	155
B.5. Domínio de Planos (Plan)	156
Apêndice C - Transformadores Draco-Puc	159
C.1. Transformador Dtd2Grm	159
C.2. Transformador HandleDiff	176
C.3. Transformador Refine	184
C.4. Transformador de Reconhecimento de Planos	188
Apêndice D - Modelos do protótipo da Ferramenta DiffTraceTool	218
D.1. Diagrama de Classes – Controle de Configuração	218
D.2. Diagrama de Classes - Modelo	219
D.3. Diagrama de Classes – Modelo de Cenários	220
Apêndice E - Bibliotecas de Planos	221
E.1. Reconhecimento de Relacionamentos	221
E.2. Reconhecimento de Operações	223

## Lista de figuras

Figura 1 - Exemplo de Construção de Domínio na Máquina Draco-PUC	29
Figura 2 – Notação utilizada na descrição das transformações	30
Figura 3 - Visão Geral do Uso da Máquina Draco-PUC	31
Figura 4 – Padrões de utilização da Máquina Draco-PUC	32
Figura 5 – Exemplo de transformação dirigida por estratégia	33
Figura 6- Classificação dos Domínios de acordo com o Nível de Abstração	35
Figura 7 – Rede de Domínios	35
Figura 8 - Visão Parcial de uma Rede de Domínios e suas Conexões	37
Figura 9 - Rede de Domínios Draco-PUC	38
Figura 10 - Descrição dos domínios Draco-PUC	38
Figura 11 – Exemplo de construção de um domínio XSL	40
Figura 12 – Uso da Rede de Domínios XSL	40
Figura 13 – Comparação das formas de implementação de Redes de Domínios	41
Figura 14 - Integração Draco-PUC – XSL	42
Figura 15 – Transformador Dtd2Grm	43
Figura 16 – Exemplo de geração da definição da sintaxe Draco-PUC	43
Figura 17 – Ferramenta DracoDebugger	45
Figura 18 – Diagrama de entidades e relacionamentos para cenários	49
Figura 19 – Exemplo de descrição de um cenário	50
Figura 20 – Exemplo de definição do LAL	51
Figura 21 – Taxonomia para evolução de cenários	52
Figura 22 – Operações sobre cenários	54
Figura 23 – Modelo de referência segundo Ramesh e Jarke	56
Figura 24 - Classificação das ligações de rastreamento quanto ao conteúdo	58
Figura 25 - Classificação das ligações de rastreamento quanto ao sentido	59
Figura 26 - Pré-Rastreamento de Requisitos Estendido	60
Figura 27 – Visão Geral do Mecanismo de Rastreamento	63
Figura 28 – Definição do tipo do documento (DTD) de descrição de cenários	66
Figura 29 – Cenário <i>malfunction occurs</i>	66

Figura 30 – Descrição XML do cenário <i>malfunction occurs</i>	67
Figura 31 – Definição da sintaxe do domínio de cenários Draco-PUC	67
Figura 32 – Representação HTML para o cenário <i>malfunction occurs</i>	68
Figura 33 – Processo de Geração das Transformações de Rastreamento	69
Figura 34 – Exemplo de aplicação do algoritmo TreeDiff	71
Figura 35 – Comparação DAST – DOM	71
Figura 36 - Uso do Algoritmo TreeDiff	72
Figura 37 – Exemplo do LAL com notação estendida	74
Figura 38 - Cenário <i>malfunction occurs</i> – Versão II	74
Figura 39 – Cenário <i>malfunction occurs</i> – Versão III	74
Figura 40 – Resultado da aplicação do algoritmo	75
Figura 41 – Alternativas estudadas para a descrição das diferenças	76
Figura 42 – Domínio de diferenças na Máquina Draco-PUC (Diff)	77
Figura 43 – Uso de descrições de diversos domínios na Máquina Draco-PUC	78
Figura 44 – Exemplo de descrição de diferença no domínio <i>Diff</i>	78
Figura 45 - Exemplo dos níveis de descrição de uma diferença	80
Figura 46 – Geração do programa representativo das mudanças - 1	81
Figura 47 - Geração do programa representativo das mudanças - 2	82
Figura 48 – Domínio de Planos Draco-PUC / Descrição de fatos observados	83
Figura 49 – Exemplos de fatos observados sobre diferenças entre artefatos	84
Figura 50 – Transformador de Geração das Transformações de Rastreamento	85
Figura 51 – Linguagem de descrição de planos	88
Figura 52 – Exemplo de definição de biblioteca de planos	90
Figura 53 – Exemplo de descrição de fatos observados	91
Figura 54 – Transformador de Reconhecimento de Planos	93
Figura 55 – Transformador de Reconhecimento de Planos (continuação)	94
Figura 56 – Exemplo de biblioteca de planos	95
Figura 57 – Visualização do funcionamento do reconhecimento de planos	96
Figura 58 – Processo de Refinamento de Modelos	97
Figura 59 – Transformador de Refinamento	98
Figura 60 – Exemplo de refinamento de modelo	99
Figura 61 – Visão geral do sistema de rastreamento para cenários	103
Figura 62 – Processo de identificação dos relacionamentos entre cenários	105

Figura 63 – Equações para o cálculo do nível de similaridade de um nó	106
Figura 64 – Exemplo de cálculo do nível de similaridade. A) Árvore do cenário original. B) Árvore do novo cenário. C) LAL dos termos utilizados	107
Figura 65 – Níveis de similaridade para a geração dos fatos observados	107
Figura 66 – Biblioteca de Planos de Relacionamentos – Parte I	108
Figura 67 – Cenário <i>malfunction</i> – Versão I	108
Figura 68 – Cenário <i>malfunction occurs</i> – Versão I	109
Figura 69 – Diferenças encontradas entre cenários	109
Figura 70 - Heurísticas para identificação de relacionamentos	110
Figura 71 - Biblioteca de Planos de Relacionamentos – Parte II	111
Figura 72 - Processo de identificação das operações entre cenários	113
Figura 73 – Cenário <i>malfunction occurs</i> – Versão II	114
Figura 74 – Resultado da comparação entre as versões I e II	114
Figura 75 – Tela principal da ferramenta <i>DiffTraceTool</i>	118
Figura 76 – Tela de edição de cenários	118
Figura 77 – Exemplos dos arquivos HTML gerados	119
Figura 78 – Informações disponíveis sobre o o estudo de caso.	120
Figura 79 – Cenário <i>define light scene</i> – Versão I	121
Figura 80 – Cenário <i>user defines light scene</i> – Versão I	121
Figura 81 - Cenário <i>facility manager defines light scene</i> – Versão I	122
Figura 82 – Resultado da comparação entre cenários	122
Figura 83 – Fatos observados sobre as similaridades existentes	123
Figura 84 – Resultado do reconhecimento de planos de relacionamentos	124
Figura 85 – Fatos observados sobre os cenários.	124
Figura 86 – Resultado da técnica de reconhecimento de planos de operações	125
Figura 87 – Resultados da aplicação do mecanismo de rastreamento	125
Figura 88 – Partes do mecanismo dependentes do tipo de artefato	130