



Thiago Pinheiro de Araújo

**SDiff: Uma ferramenta para comparação de documentos com
base nas suas estruturas sintáticas**

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: Prof. Arndt von Staa

Rio de Janeiro
Março de 2010



Thiago Pinheiro de Araújo

SDiff: Uma ferramenta para comparação de documentos com base nas suas estruturas sintáticas

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Informática do Departamento de Informática do Centro Técnico e Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Arndt von Staa
Orientador

Departamento de Informática - PUC-Rio

Prof. Renato Fontoura de Gusmão Cerqueira
Departamento de Informática - PUC-Rio

Prof. Marcus Vinicius Soledade Poggi de Aragão
Departamento de Informática - PUC-Rio

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

Rio de Janeiro, 8 de março de 2010

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

Thiago Pinheiro de Araújo

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

Ficha Catalográfica

Araújo, Thiago

SDiff: Uma ferramenta para comparação de documentos com base nas suas estruturas sintáticas / Thiago Pinheiro de Araújo; orientador: Arndt von Staa. — Rio de Janeiro : PUC–Rio, Departamento de Informática, 2010.

v., 95 f.: il. ; 29,7 cm

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

Inclui referências bibliográficas.

1. Gerência de configuração. 2. Controle de versão. 3. Ferramenta de desenvolvimento de software. 4. Engenharia de Software. I. Staa, Arndt von. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Para meus pais, Ítalo e Antônia.

Agradecimentos

Ao meu pai, Ítalo José de Araújo, meu melhor amigo, que se despediu há pouco tempo. Sem sua presença, experiência de vida, participação, incentivo, motivação, alegria e interesse na minha formação, não seria possível conquistar tudo que conquistei. Muito obrigado por tudo, e olhe por nós.

A minha mãe, Antônia dos Santos Cunha Pinheiro, e a minha tia, Maria do Carmo Cunha Pinheiro, pelo apoio e carinho nos momentos mais difíceis. Obrigado pela incansável dedicação.

A Ana Carolina Andrade, pelo seu amor, pela sua paciência, pelos seus conselhos e pelo seu apoio e compreensão em cada momento de nossas vidas.

Ao meu orientador, prof. Arndt von Staa, por toda sua dedicação e interesse na minha formação acadêmica. Obrigado pelos seus conselhos, pela sua orientação, pelo seu entusiasmo com a pesquisa, e por todas as conversas que tivemos, sempre construtivas para minha vida acadêmica e pessoal.

A Maria Isabel Cristovão Prado, pela sua amizade em todos esse anos, pelo carinho e atenção que têm com a nossa família, e também pelo papel fundamental que teve na minha carreira. Muito obrigado.

A Carlos Eduardo Crestana e Ricardo Gomes, pela grande amizade e confiança no meu potencial. Agradeço também a disponibilidade e paciência nas discussões que contribuíram para a realização deste trabalho.

A João Magalhães pela sua orientação, apoio e compreensão.

Ao CNPq e à 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 meus familiares, amigos de infância, amigos da graduação e do mestrado, que de alguma forma contribuíram para a realização deste trabalho.

Resumo

Araújo, Thiago; Staa, Arndt von. **SDiff: Uma ferramenta para comparação de documentos com base nas suas estruturas sintáticas**. Rio de Janeiro, 2010. 95p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Associado a cada sistema de controle de versão existe uma ferramenta de comparação responsável pela extração das diferenças entre duas versões de um documento. Estas ferramentas costumam realizar a comparação baseando-se na informação textual dos documentos, em que o elemento indivisível na comparação é a linha ou a palavra. Porém, o conteúdo versionado normalmente é fortemente estruturado (como exemplo, linguagens de programação) e a utilização deste mecanismo pode desrespeitar limites sintáticos e outras propriedades do documento, dificultando a interpretação das alterações. Nesse trabalho foi construída uma ferramenta para identificar as diferenças entre duas versões de um documento utilizando um mecanismo de comparação baseado na sua estrutura sintática. Desta forma, é possível identificar com maior precisão as diferenças relevantes ao leitor, reduzindo o esforço para compreender a semântica das alterações. A ferramenta construída é capaz de suportar diferentes tipos de documentos a partir da implementação de componentes que tratem das sintaxes desejadas. O componente implementado como exemplo neste trabalho trata a sintaxe da linguagem de programação C++.

Palavras-chave

Gerência de configuração; Controle de versão; Ferramenta de desenvolvimento de software; Engenharia de software.

Abstract

Araújo, Thiago; Staa, Arndt von (Advisor). **SDiff: a comparison tool based in syntactical document structure**. Rio de Janeiro, 2010. 95p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Associated with each version control system there's a comparison tool for extracting the differences between two versions of a document. These tools tend to make a comparison based on textual information from documents, in which the indivisible element is the line or word. But the content versioned is usually highly structured (for example, programming languages) and the use of this mechanism can disrespect syntactical limits and other properties of the document, becoming difficult to interpret what really changed. In this work we created a tool to identify differences between two versions of a document using a comparison mechanism based on the syntactic structure. Thus, it is possible to identify more precisely the relevant differences to the reader, reducing the effort to understand the semantics of the changes. The tool can support different types of documents by implementing components that interprets the desired syntax. The example syntax component implemented in this work deals with the syntax of the programming language C++.

Keywords

Software Configuration Management; Version control system; Software development tool; Software Engineering.

Sumário

1. Introdução	15
2. Motivação	18
3. Estado da Arte	23
3.1 Baseado em operações	24
3.2 Baseado em refatoração	25
3.3 Baseado em sintaxe	26
3.4 Orientado a modelos	27
3.5 SCMs com granularidade variável	27
3.6 Comparação híbrida	28
4. O mecanismo de comparação híbrido	29
4.1 Conceitos gerais e nomenclaturas	29
4.2 Estrutura sintática canônica	32
4.3 Mecanismo de comparação	34
4.3.1 Estruturas <i>Document</i> e <i>DocumentElement</i>	34
4.3.2 Estrutura <i>Diff</i>	36
4.3.3 Algoritmo de comparação	39
4.4 Escolha entre comparação textual ou sintática	56
4.5 Calibração das propriedades	57
4.6 Comparação com o estado da arte	59
5. A ferramenta SDiff	61
5.1 Informações técnicas	61
5.2 O Framework DMF	62
5.2.1 Ponto de extensão: tipo de documento	65
5.2.2 Ponto de extensão: algoritmo de comparação de elementos	65
5.3 Visualização das diferenças	66
5.4 Estatísticas gerais do código-fonte	67

5.5	Controle da qualidade	69
6.	Experimentos	70
6.1	Comparação dos resultados com os de uma ferramenta tradicional	70
6.1.1	Inserção e remoção simples	70
6.1.2	Modificação	71
6.1.3	Movimentação	72
6.1.4	Alteração no contexto	73
6.1.5	Reformatação	75
6.1.6	Comparação de elementos com tipos diferentes	76
6.1.7	Extração de código	77
6.2	Experimentos em aplicações reais	78
6.3	Estatísticas da execução da ferramenta SDiff	80
7.	Conclusão	85
8.	Trabalhos futuros	87
8.1	Melhorias no mecanismo de comparação	87
8.2	Melhorias na ferramenta	88
8.3	Novas ferramentas	89
	Referências Bibliográficas	90

Lista de figuras

Figura 1 – Exemplo de código de duas versões de um documento onde a ordem das declarações é modificada.	18
Figura 2 – Exemplo de comparação textual sobre duas versões de um documento onde a ordem das declarações é modificada.	19
Figura 3 – Exemplo de código de duas versões de um documento onde ocorre a extração de um trecho de código para um novo método.	19
Figura 4 – Exemplo de comparação textual sobre duas versões de um documento onde um trecho de código é extraído para um novo método.	20
Figura 5 – Processo realizado pelo mecanismo de comparação	31
Figura 6 – Exemplo da estrutura de um trecho de código em uma linguagem de programação qualquer.	32
Figura 7 – Trecho de código utilizado como exemplo.	33
Figura 8 – XML gerado a partir do trecho de código apresentado na Figura 7.	33
Figura 9 – Estrutura interna utilizada pelo algoritmo de comparação.	35
Figura 10 – Estrutura que representa a resposta do mecanismo de comparação.	36
Figura 11 – Estruturas sintáticas que representam trechos de código a serem comparados para exemplificar a estrutura de resposta.	38
Figura 12 – Estrutura de diferenças resultante da comparação das estruturas apresentadas na Figura 11.	39
Figura 13 – <i>Pseudo-código</i> do algoritmo de comparação de elementos.	41
Figura 14 – Exemplo de código utilizado para exemplificar a classificação de propriedades.	43
Figura 15 – Exemplo do XML que representa o elemento sintático da declaração do destrutor na linha 26 da Figura 14.	43
Figura 16 – <i>Pseudo-código</i> do algoritmo de comparação textual.	45
Figura 17 – Definição da estrutura <i>TextDiff</i> .	45
Figura 18 – <i>Pseudo-código</i> do pós-processamento de união de	

diferenças próximas.	47
Figura 19 – <i>Pseudo-código</i> do procedimento que transforma a lista de diferenças textuais em diferenças estruturais.	48
Figura 20 – <i>Pseudo-código</i> do algoritmo de comparação sintática de sub-elementos.	50
Figura 21 – Inicialização do peso de um par de elementos.	52
Figura 22 – <i>Pseudo-código</i> da heurística baseada na semelhança dos elementos.	52
Figura 23 – <i>Pseudo-código</i> da heurística baseada na distância dos elementos.	53
Figura 24 – Exemplo da heurística baseada na distância dos elementos no caso que os elementos devem ser considerados diferentes.	53
Figura 25 - Exemplo da heurística baseada na distância dos elementos no caso que os elementos não devem ser considerados diferentes.	54
Figura 26 - <i>Pseudo-código</i> da heurística baseada no nome dos elementos.	54
Figura 27 – Diagrama de classes da arquitetura do DMF.	62
Figura 28 – Interface <i>DocumentTypePlugin</i> .	65
Figura 29 – Interface <i>ComparatorStrategy</i> .	66
Figura 30 – Interface gráfica da ferramenta SDiff.	66
Figura 31 – Exemplo de inserção e remoção mostrado na aplicação <i>KDiff3</i> .	70
Figura 32 – Exemplo de inserção e remoção mostrado na aplicação <i>SDiff</i> .	71
Figura 33 – Exemplos de modificação mostrado na aplicação <i>KDiff3</i> .	71
Figura 34 – Exemplos de modificação mostrado na aplicação <i>SDiff</i> .	71
Figura 35 – Exemplos de movimentação mostrado na aplicação <i>KDiff3</i> .	72
Figura 36 – Exemplos de movimentação mostrado na aplicação <i>SDiff</i> .	72
Figura 37 – Exemplos de alteração de propriedades contextuais mostrado na aplicação <i>KDiff3</i> .	74
Figura 38 – Exemplos de alteração de propriedades contextuais mostrado na aplicação <i>SDiff</i> .	74

Figura 39 – Exemplos de reformatação mostrado na aplicação <i>KDiff3</i> .	75
Figura 40 – Exemplos de reformatação mostrado na aplicação <i>SDiff</i> .	75
Figura 41 – Exemplo de comparação de elementos sintáticos com tipos diferentes mostrado na aplicação <i>KDiff3</i> .	76
Figura 42 – Exemplo de comparação de elementos sintáticos com tipos diferentes mostrado na aplicação <i>SDiff</i> .	76
Figura 43 – Exemplo de extração de código para um novo método mostrado na aplicação <i>KDiff3</i> .	77
Figura 44 – Exemplo de extração de código para um novo método mostrado na aplicação <i>SDiff</i> .	77
Figura 45 – Exemplo de bloco formado por comentário multi-linha.	78
Figura 46 – Exemplo de bloco formado por comentário simples.	79
Figura 47 – Número de linhas na representação XML versus Número de linhas no documento.	81
Figura 48 – Número total de elementos sintáticos versus Número de linhas no documento.	81
Figura 49 – Número de comparações versus Número de linhas no documento.	83
Figura 50 – Tempo de execução versus Número de linhas no documento.	84

Lista de tabelas

Tabela 1 – Estatísticas sobre o número de linhas do código-fonte.	68
Tabela 2 – Estatísticas sobre o controle da qualidade	68
Tabela 3 – Outras estatísticas.	68
Tabela 4 – Estatísticas relacionadas ao tamanho de cada instância.	80
Tabela 5 – Estatísticas relacionadas ao número de comparações em cada instância.	82
Tabela 6 – Apresenta os tempos de execução para cada instância.	83