



Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação

Extensão e análise de performance da ferramenta de merge textual CSDiff para novas linguagens

Aluno: Heitor Sammuel Carvalho Souza (hscs@cin.ufpe.br)

Orientador: Paulo Henrique Monteiro Borba (phmb@cin.ufpe.br)

Área: Engenharia de Software e Sistemas

20 de Setembro de 2021.

1. Resumo

Conflitos de *merge* são presentes no dia a dia do desenvolvimento de software e podem causar impacto negativo na performance dos desenvolvedores. Uma das linhas de pesquisa para tentar reduzir esse problema é a evolução das ferramentas utilizadas para auxiliar no processo de *merge* automatizado. Hoje, a abordagem mais comum é o *merge* textual, onde a comparação entre os arquivos de código é feita considerando apenas as linhas do texto. No outro extremo, existe o *merge* estruturado, que compara os arquivos levando em consideração sua estrutura sintática e semântica. Neste trabalho, damos continuidade no desenvolvimento de uma ferramenta (CSDiff) que utiliza o *merge* textual mas visa simular o estruturado, utilizando os marcadores sintáticos de cada linguagem para guiar o processo de *merge*. Mostramos uma extensão da ferramenta capaz de analisar conflitos de código nas linguagens *Ruby* e *TypeScript*, bem como uma análise de sua acurácia em *commits* de *merge* em repositórios de código aberto.

Merge conflicts are present on a daily basis in software development, and can cause a negative impact on the performance of the developers. One of the research tracks that try to reduce this problem is the evolution of the tools used to help on the automated merge process. Today, the most common approach is the textual merge, where the comparison between the code files is done considering only the text lines. At the other end, we have the structured merge which compares the archives taking into account their syntactic and semantic structure. In this work we continue the development of a tool (CSDiff) that uses textual merge but simulates the structured, by using the syntactic markers of each language to guide the merge process. We show an extension of the tool capable of analyzing code conflicts on the Ruby and TypeScript languages, as well as an analysis of its accuracy at merge commits on open source repositories.

2. Introdução

Sabemos que conflitos em um mesmo arquivo de código devido a mudanças simultâneas realizadas por dois ou mais desenvolvedores são, no presente momento, parte do cotidiano na área de desenvolvimento de software. Tais conflitos podem impactar na produtividade[1] do desenvolvimento de código, e além disso, a ferramenta de *merge* mais utilizada atualmente baseia-se em uma abordagem textual, linha a linha - isso significa que a estrutura sintática e semântica do código não é levada em consideração, podendo levar a falsos negativos (isto é, não existe um conflito na linha, porém existe um conflito semântico) ou a falsos positivos (o conflito reportado na linha poderia ser resolvido automaticamente sem impactar na funcionalidade original do código).

Uma abordagem possível para a criação de uma ferramenta de *merge* mais robusta que o textual, porém sem um possível impacto negativo na performance ou o esforço extra de adequar a ferramenta a cada nova linguagem de programação (como é o caso do merge estruturado)[2][3], é a de levar em consideração os separadores sintáticos específicos da linguagem, de forma a simular o *merge* estruturado[4].

Considerando isto, a proposta do presente trabalho é de dar continuidade aos estudos iniciais e ao desenvolvimento da ferramenta CSDiff, que mostrou resultados positivos, ainda que pouco expressivos, em sua versão inicial[4]. Tais resultados abrem portas para melhorias na solução proposta, bem como sua extensão para suporte de novas linguagens, a fim de obter mais dados e métricas para investigar mais profundamente sobre seu desempenho e eficácia como ferramenta de *merge*.

3. Objetivos

O objetivo **geral** deste trabalho é estender a funcionalidade original da ferramenta CSDiff, inicialmente elaborada por Clementino [4], de forma a suportar duas novas linguagens. Em sequência, será analisado o desempenho da nova versão do CSDiff na identificação de conflitos de merge em repositórios de projetos de software *open source*.

São objetivos **específicos** deste projeto:

- Estender a funcionalidade da ferramenta CSDiff para duas novas linguagens: *Ruby* e *TypeScript*;
- Ajustar a infraestrutura de experimento (*Mining Framework*¹) para a nova versão do CSDiff
- Analisar o desempenho do CSDiff modificado em *commits* de *merge* em repositórios *open source* que utilizem as linguagens, através do *Mining Framework*;
- Buscar um conjunto de marcadores ótimos para as linguagens *Ruby* e *TypeScript*;
- Escrever a monografia para defesa oral deste trabalho;

¹<https://github.com/spgroup/miningframework>

4. Metodologia

O primeiro passo do desenvolvimento do trabalho é o estudo da ferramenta original, CSDiff, desenvolvida por Clementino [4], para entender seu funcionamento e estender o comportamento original para suportar duas novas linguagens, *Ruby* e *TypeScript*. Para isso, será necessário definir um conjunto de marcadores sintáticos que serão utilizados para cada linguagem.

Em seguida, será utilizado o *Mining Framework* para obter *commits* de *merge* em repositórios *open source* que utilizam, majoritariamente, as duas novas linguagens implementadas no CSDiff.

Então, os *commits* obtidos serão analisados e os resultados da ferramenta CSDiff serão comparados com os resultados finais presentes nos repositórios. Dessa forma, será possível obter dados sobre a eficácia da ferramenta em diferentes cenários de *merge*, a fim de responder as perguntas de pesquisa.

Por fim, os resultados serão compilados e desenvolvidos no corpo do trabalho. Adicionalmente, será possível buscar por um conjunto ideal de marcadores sintáticos para cada linguagem, de forma a maximizar a eficácia do CSDiff.

Durante toda a elaboração do trabalho estão previstas reuniões semanais com o orientador, para entrega e discussão de resultados e possíveis adaptações do cronograma, assim como para definir os objetivos de curto prazo para a semana seguinte.

5. Cronograma

Atividade	Set	Out	Nov	Dez
Configuração do Ambiente para Experimentação	X	X		
Extensão do CSDiff para <i>Ruby</i> e <i>TypeScript</i>		X		
Execução do Experimento com o <i>Mining Framework</i>		X	X	
Análise dos <i>commits</i> de <i>merge</i> e obtenção de métricas		X	X	
Análise dos Resultados			X	
Escrita do Texto		X	X	X
Preparação da Defesa				X

Referências

- [1] G. Cavalcanti, P. Borba e P. Accioly, “Should We Replace Our Merge Tools?” Em *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017, pp. 325–327. DOI: 10.1109/ICSE-C.2017.103.
- [2] G. Cavalcanti, P. Borba, G. Seibt e S. Apel, “The Impact of Structure on Software Merging: Semistructured Versus Structured Merge,” em *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2019, pp. 1002–1013. DOI: 10.1109/ASE.2019.00097.
- [3] S. Apel, J. Liebig, B. Brandl, C. Lengauer e C. Kästner, “Semistructured Merge: Rethinking Merge in Revision Control Systems,” em *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, sér. ESEC/FSE ’11, Szeged, Hungary: Association for Computing Machinery, 2011, pp. 190–200, ISBN: 9781450304436. DOI: 10.1145/2025113.2025141. endereço: <https://doi.org/10.1145/2025113.2025141>.
- [4] J. Clementino, P. Borba e G. Cavalcanti, “Textual Merge Based on Language-Specific Syntactic Separators,” em *Brazilian Symposium on Software Engineering*, sér. SBES ’21, Joinville, Brazil: Association for Computing Machinery, 2021, pp. 243–252, ISBN: 9781450390613. DOI: 10.1145/3474624.3474646. endereço: <https://doi.org/10.1145/3474624.3474646>.

6. Possíveis Avaliadores

São possíveis avaliadores do trabalho os professores:

- Daniel Carvalho da Cunha (dcunha@cin.ufpe.br)
- Odilon Maroja da Costa Pereira Filho (odilon@cin.ufpe.br)
- José Augusto Suruagy Monteiro (suruagy@cin.ufpe.br)

7. Assinaturas

Paulo Henrique Monteiro Borba
Orientador

Heitor Sammuel Carvalho Souza
Orientando