UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

Aperfeiçoamento na detecção de conflitos utilizando merge semiestruturado

PROPOSTA DE TRABALHO DE GRADUAÇÃO

Aluno: Giovanni Antonio Araujo de Barros e Silva (gaabs@cin.ufpe.br)
Orientador: Paulo Borba (phmb@cin.ufpe.br)
Área: Engenharia de Software

Recife, Agosto de 2018

Sumário

1. Contexto

No desenvolvimento de software, é comum que membros diferentes trabalhem simultaneamente em atividades de um mesmo projeto. Como ambas tarefas podem alterar ou fazer uso de uma parte comum do código, é possível que surjam efeitos indesejados decorrentes da combinação das mudanças, sendo estes denominados conflitos.

Atualmente, é bastante disseminado o uso de sistemas de versionamento de código, como o Git. Fazendo uso de sistemas como esse, dois desenvolvedores podem podem trabalhar paralelamente em versões diferentes do código, cada um fazendo suas modificações. Quando versões diferentes são combinadas, é necessário verificar se ocorreram conflitos e resolvê-los corretamente.

A forma mais comum de combinação de versões de código é utilizando o merge não estruturado. Nela, os arquivos que foram modificados por ambas as partes são comparados textualmente, de forma a detectar conflitos se ambas versões alteraram a mesma parte do arquivo. Essa forma de verificação, no entanto, é bastante simples e propensa a erros, uma vez que os conflitos não surgem apenas nas linhas que ambos modificaram.

O merge estruturado, por sua vez, leva em conta os elementos da linguagem em questão verificar resolver automaticamente os conflitos a partir de comparações estruturais. É uma forma bem mais precisa de detectar os conflitos, porém, é menos eficiente em relação ao tempo de execução.

Outra técnica existente é o merge semiestruturado, que se trata de uma mistura das duas abordagens anteriores: faz-se uso da comparação estruturada quando possível, mas também utiliza merge não estruturado, de forma a atingir um bom desempenho na detecção de conflitos e com custo computacional razoável [1].

A resolução manual de conflitos, no entanto, pode ser uma tarefa cansativa e tediosa, sendo um empecilho à produtividade, além de ser passível de erros. É possível, ainda, que os conflitos não sejam detectados e comprometam a corretude do sistema [2]. Inclusive, estudos mostram que mais de 23% dos cenários de merge apresentam conflitos [3]. Isso evidencia a necessidade de técnicas de detecção e resolução automática de conflitos.

2. Objetivos

O objetivo do trabalho em questão é aperfeiçoar as técnicas atuais de merge semiestruturado, de forma a possibilitar sua adoção por parte de empresas e desenvolvedores. Espera-se, que com isso, haja uma melhoria na qualidade do trabalho, diminuindo o tempo demandado para resolução manual de conflitos e tornando a combinação de versões mais confiável.

Para tanto, vai ser utilizada a ferramenta jFSTMerge[1], desenvolvida por Guilherme Cavalcanti. Trata-se de uma ferramenta de merge semiestruturado para a linguagem Java.

A proposta, então, tem como meta a criação de módulos para aperfeiçoar a ferramenta existente, de modo a melhorar a detecção e resolução de conflitos.

Além disso, este experimento pretende fazer um estudo comparativo de maneira a avaliar a ferramenta com e sem os módulos que serão desenvolvidos, tendo em vista uma análise qualitativa e de performance.

[1] Disponível em https://github.com/guilhermejccavalcanti/jFSTMerge

3. Cronograma

As atividades serão desenvolvidas de acordo com o cronograma abaixo.

Atividade	**Agosto**	**Setembro**	**Outubro**	**Novembro**	**Dezembro**
Elaboração da Proposta	X				
Revisão da Literatura	X	X			
Implementação		X	X	X	
Análise e Homologação de resultados			X	X	
Elaboração do relatório final			X	X	X
Apresentação					X

4. Possíveis Avaliadores

Possíveis avaliadores do trabalho a ser produzido são os professores Leopoldo Teixeira e Henrique Rebêlo.

5. Referências

[1] APEL, Sven et al. Semistructured merge: rethinking merge in revision control systems. In: Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering. ACM, 2011. p. 190-200.

[2] CAVALCANTI, Guilherme; BORBA, Paulo; ACCIOLY, Paola. Evaluating and improving semistructured merge. Proceedings of the ACM on Programming Languages, v. 1, n. OOPSLA, p. 59, 2017.

[3] ZIMMERMANN, Thomas. Mining workspace updates in CVS. In: Mining Software Repositories, 2007. ICSE Workshops MSR'07. Fourth International Workshop on. IEEE, 2007. p. 11-11.

6. Assinaturas

_____

Giovanni Antonio Araujo de Barros e Silva
(Aluno)

_____

Paulo Borba
(Orientador)