VUB SCIENCES & BIOENGINEERING SCIENCES

The Research Group
Software Languages Lab

has the honor to invite you to the public defence of the PhD thesis of

Ward Muylaert

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:

Data flow and Control flow Analysis of Problematic Commits

Promotor:

Prof. dr. Coen De Roover

The defence will take place on

Monday, April 22, 2024 at 4:00 p.m. in auditorium D.2.01

Members of the jury

Prof. dr. Viviane Jonckers (VUB, chair)
Prof. dr. Lynn Houthuys (VUB, secretary)
Prof. dr. Kris Steenhaut (VUB)
Prof. dr. Josep Silva Galiana (Universitat Politècnica de València, Spain)
Prof. dr. Jean-Rémy Falleri (Université de Bordeaux, France)

Curriculum vitae

Ward Muylaert previously obtained a Bachelor's degree in Mathematics and a Master's degree in Computer Science at the VUB. Throughout his PhD, he published four papers supporting this work. Ward has TAd three different first bachelor courses over the years as well as guiding a bachelor's thesis and two master's theses. In his free time, Ward has contributed to various open-source projects. He is a board member in a non-profit organisation with a special focus on disadvantaged children and has been board member in another NPO focused on the spread of data and knowledge.

Abstract of the PhD research

When creating and maintaining programs, software developers make use of version control software to store different versions of the source code. Version control software uses commits as building blocks. Commits play a dual role: they represent a version of the software program at a certain point in time; they also represent the changes compared to the preceding version of the program.

In this dissertation, we investigate two types of commits that are harder for developers to understand. First, we consider composite commits. Composite commits group many unrelated changes. The changes target different tasks, such as fixing a bug or introducing a new feature. Besides being harder to understand, composite commits also prove harder for software developers to revert or integrate and for empirical researchers to analyse. Second, we consider merge commits. Versions of the program start to diverge when different developers work on different features or bug fixes. A merge commit recombines divergent versions. An overlap in source code or an interaction in behaviour demands a resolution from the software developer, requiring them to understand the source code of the different versions.

We propose an algorithmic foundation for tool support for these two types of problematic commits. Our first algorithm untangles composite commits using a data flow driven approach. Our algorithm considers a program dependence graph and fine-grained changes to the abstract syntax tree. The algorithm groups fine-grained changes according to the program dependence graph slices they belong to. Our second algorithm analyses merge commits using a control flow driven approach. It uses symbolic execution to gather path conditions of the different versions of the program. We define the program semantics in function of these path conditions. The path conditions are checked against rules that indicate presence of a semantic merge conflict.

We evaluate both algorithms. By analysing, refining, and using an established dataset of composite commits, we find our untangling algorithm can determine whether a commit is composite. The groups of fine-grained changes tend to be smaller than the commit's tasks, but stay within their boundaries. We evaluate our semantic merge conflict detection algorithm in two ways. First, we evaluate its correctness through mutation testing. Second, we evaluate it empirically by applying it to real-world merges. We discuss challenges in the empirical evaluation of semantic merge conflicts. Our evaluation shows that in specific cases our approach is a promising extension to existing mechanisms in semantic merge conflict detection.