Master’s Thesis

JOHANNES KEPLER UNIVERSITÄT LINZ

DEVELOPING AN EXTRACTOR FOR MINING VARIABILITY FROM PRODUCT VARIANTS

Keywords: code comparison, variability mining, variant analysis

Current Situation

Extracting variability information from source code or other artifact types of a set of given product variants is a complex task that requires differencing files. This is currently mostly done by performing simple line-by-line comparisons. Even though these approaches are sufficient for detecting major additions or deletions, they are not well-suited for identifying and extracting variability. While there exist tools that better support these tasks, they are mostly restricted to specific use-cases.

Betreuung:

DI Alexander Stummer
alexander.stummer@jku.at

Background

Software Product Lines (SPL) allow systematic reuse of common code across different variants of a system. While this approach is well-known, due to high upfront effort this technique is often not applied from project startup. When the number and size of variants increase over time, at some point it is no longer possible to maintain and manage variability. At this point, the transformation into a SPL is desired. However, doing this manually is not feasible. Therefore, automatically extracting and analyzing variability information is needed. An approach, that can perform this process for any type and combination of artifact types is currently in development, which utilizes extractors for individual artifacts.

Content of the Thesis

The goal of this thesis is to develop an extractor, that can compare an arbitrary number of variant files of a specific artifact type (e.g., source code of a programming language, spreadsheets, config files etc.) for commonalities/variabilities, filter them for relevance (e.g., moved code, split statements can be omitted) and provides information about their occurrences (in which variant does variability x appear). The artifact type, for which the extractor is written, can be chosen based on the interest and knowledge of the student. Integrating this extractor into a flexible framework for variability mining is also desired. The extraction process itself should be implemented and tested using Java.

Requirements

- Java programming skills required
- Knowledge about the artifact type, for which the extractor is implemented, is advantageous
- Excellent German or English skills

Learning Outcomes

- Learning about software variability, it’s location and extraction
- Getting to know and apply comparison and analysis methods
- Apply versioning systems in practice (Git)
- Working in a scientific process