

Automated API Refactoring with Clang



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Applicable for students as HiWi, Bachelor of Science, Master of Science

Keywords: *Clang, Source Analysis, Source Transformation, Software Quality*

Introduction

When developing and adding new features to libraries or programs, interoperability with existing code is one of the highest priorities. Sometimes API-breaking changes are unavoidable, leading to a snowball effect of necessary changes in other code. In many cases, these API-breaking changes are either the removal or addition of a new parameter to a function or a slight change in return-type. In object-oriented languages there is the additional possibility to introduce a change that transfers state knowledge into a different object.

Task

Develop a Clang-based refactoring tool, that reads a description of the API changes of a given program revision, and automatically refactors a program accordingly.

```
1 int lib_func(int a, int b);  
2 //changed to  
3 double lib_func(int a, int b, bool useUnsafeMath);
```

Figure 1: A library function that had a change in its return value, and signature

```
1 node.getChildren();  
2 //changed to  
3 graph.getChildren(node)
```

Figure 2: Graph object now handles node relations, instead of each node

What you will be doing

- Propose a machine- and human-readable format for documenting API changes
- Develop a Clang refactoring tool, which reads the API changes, and modifies a given source code accordingly
- Evaluate your approach w.r.t. applicability on larger code bases like Clang

Qualifications

- Knowledge of the Clang tooling library [1].
- Experience with C++.
- Optional: Experience in generating and parsing text format files in C++

References

- [1] <https://clang.llvm.org/docs/LibTooling.html>



Tim Heldmann
tim.heldmann@tu-darmstadt.de

Office: S1J03 Room 4a
Hochschulstraße 1
64283 Darmstadt
Tel. 06151 16-27275

Date: 3rd November, 2022

