

Leveraging the External Resources and Meta-data to Highlight the Gap between a Program's Implementation and its Documentation

ON

17th August @ 7:30 PM IST

Advisor: Dr. Rahul Purandare

Examiners:

- Serge Demeyer, Professor, University of Antwerp
- Tefvik Bultan, Professor, University of California, Santa Barbara
- Andy Zaidman, Professor, Delft University of Technology

Ph.D. Thesis Defense



Devika Sondhi

Please visit <https://cse.iiitd.ac.in/events-seminars/>
for more details

Abstract:

With the ever-growing dependency on softwares, testing for their unexpected behavior is as important as verifying for their known properties, to avoid potential losses. Existing software testing approaches assume either the specification as a baseline to test for the intended behaviour of the program's implementation or propose to find ambiguities in the documentation or the specification, with respect to the implementation.

In reality, the issue may arise from either of the two sources and hence, it would be more appropriate to build testing techniques to highlight the inconsistencies between the two. There are two limitations that exist in restricting the testing approach to only the implementation and its associated documentation: 1) formulating complete and precise specifications is a hard problem, and 2) test cases giving high coverage on the program may not suffice in ensuring a bug-free implementation. In this direction, the dissertation proposes to leverage resources apart from the documentation and the source code to generate tests to highlight the inconsistencies and understand the introduction of these inconsistencies as a code project evolves.

We leverage existing resources such as test suites from other similar programs, domain knowledge of the developers, external resources such as RFCs etc., for effective test generation. We propose two test generation approaches: 1) Mining existing test suites associated with similar functions in other libraries, to generate test cases. 2) Obtaining a differential model highlighting semantic gaps arising from inconsistencies in an input structure, as inferred from the function's implementation and its associated documentation. These approaches generated tests to reveal defects in real programs, which indicates the effectiveness of leveraging external resources. The first approach was shown to reveal 67 defects through a study on leveraging similar libraries for test generation. These defects were then used to assess the proposed tool to automatically recommend test cases obtained by mining similar functions across open source libraries. The tool revealed 22 defects from the dataset of 67 defects, and additional 24 previously unknown defects, thus, revealing a total of 46 defects. The second approach, based on building a differential model, revealed 80% of the defects in the evaluation dataset and additional 6 previously unknown defects.

We then delve deeper into reasoning about the introduction of such inconsistencies in the process of code evolution where we analyse how the description in a documentation relates to the code changes made at several points in a library. We leverage the commit patterns and the developer discussions to understand the nature of these changes made in the evolution process. We observe relations between methods, such as a call-graph relation, inheritance or interface implementation, and their references in the other method's documentation, to explain the presence of dependencies that can eventually lead to inconsistencies when one entity is modified without updating the dependents.