Reverse Engineering Source Code: Empirical Studies of Limitations and Opportunities.
Dhr. D. Landman
SUMMARY

Reverse Engineering Source Code: Empirical Studies of Limitations and Opportunities

The goal of software renovation is to modernize software. One way to achieve this is to first reverse engineer the essential concepts and abstractions used in the software and then use these during renovation. Reverse engineering can use several sources: users, documentation, or source code. We have focused on reverse engineering from source code. Scaling reverse engineering to large software systems requires at the very least partially automated analysis. Automation often comes at the cost of over- or underapproximation. We have formulated three research questions to explore limits and opportunities for these approximations.

To answer the first question, we have explored the limits of domain model recovery by manually recovering two domain models from two software systems. Comparing these models to a manually constructed reference domain model based on a reference book of the domain and two manually constructed reference applications models we found that most domain information could be recovered – with high quality – by reading the source code of the software system. This motivates future work in automating the domain model recovery from source code.

In trying to automate domain model recovery, we have identified challenges that hold for a wider range of reverse engineering methods than just domain model recovery. The second and third question address these challenges in the broader context of reverse engineering.

To answer the second question, we have explored the opportunity of using both Cyclomatic Complexity (cc) and Source Lines of Code (sloc) for automating reverse engineering. Metrics, such as cc and sloc, are used in a wide variety of reverse-engineering methods to filter methods or files of interest. Almost all of the literature on the relation between the two metrics – identified using a Systematic Literature Review (SLR) – claim a strong linear correlation between them ($R^2$ between 0.51 and 0.96). This is often interpreted as indication that cc and sloc measured the same property. Often this is further interpreted that measuring cc and sloc next to each other is redundant. In two large corpora – with 362 MSLOC of Java and 186 MSLOC of C – we did not observe a strong correlation ($R^2$ of 0.40 and 0.44). We have identified two transformations of the data that did increase the correlations to the more commonly reported strengths. However these transformations complicate the interpretation of the relationship between cc and sloc. Our final interpretation is that there is a lack evidence for cc being redundant to sloc, which supports the continued used of both metrics next to each other.
In order to answer the final question, we have explored the limits of statically analyzing Java – with respect to the Reflection Application Programming Interface (API) – for a corpus of 462 Java projects (80 MSLOC). Using a SLR of all static analysis approaches – that published new heuristics for handling reflection – we have identified the common assumptions and limitations. Analyzing the corpus revealed that 78% of all projects use the parts of the Reflection API that are hard to model with static analysis. Common challenges for analysis tools such as “non-exceptional exceptions”, “programmatic filtering meta objects”, “semantics of collections”, and “dynamic proxies” widely occur in the corpus. We support Java software engineers with tactics to obtain more easy to analyze reflection code. We also propose new opportunities for static analysis tools to significantly impact the analyses of real Java code.

All three results have been obtained with empirical studies on corpora of open source software. The corpora and the scripts used to analyze them are available online to support critique from other researchers and enable future work on different challenges with the same corpora. We have used empirical studies to both answer open questions and identify new opportunities in reverse engineering research and practice.