Introducing a Benchmark for Evaluating Reverse Engineering Tools

Lajos Jenő Fülöp, Péter Hegedűs and Rudolf Ferenc

Useful information can be discovered from source code with the examination of software systems. First step of the examination is to analyze the source code and making an *abstract representation*. In the second step different structures and problems can be discovered from the abstract representation, e.g. design patterns, code duplications and coding rule violations. These structures and problems are discovered by different kind of *reverse engineering tools*.

Reverse engineering tools present their results in different formats, which makes them very difficult to compare. In our previous works [5][6] we presented a tool called DEEBEE(DEsign pattern Evaluation BEnchmark Environment) which supports the evaluation and comparison of design pattern miner tools only. We realized that DEEBEE can be extended to support the evaluation and comparison of every kind of reverse engineering tools as well. In this paper we introduce a benchmark - called BEFRIEND (BEnchmark For Reverse engInEering tools workiNg on source coDe) - with which the outputs of reverse engineering tools can be evaluated and compared easily and efficiently. The extension of DEEBEE into a general benchmark BEFRIEND is motivated by other works which try to evaluate and compare code duplication finder tools [3][4] and rule violation checker tools[1][7] as well.

BEFRIEND supports different kinds of tool families, programming languages and software systems, and it enables the users to define their own evaluation criteria. Furthermore, it is a web-application open to the community and freely available [2]. We hope it will be accepted and used by the community in the future to evaluate and compare their tools with others.

References

[1] N. Ayewah, W. Pugh, J. D. Morgenthaler, J. Penix, and Y. Zhou. Evaluating static analysis defect warnings on production software. In *PASTE '07: Proceedings of the 7th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering*, pages 1–8. ACM, 2007.

[2] DEEBEE and BEFRIEND Homepage.
`http://www.inf.u-szeged.hu/designpatterns/`.

[3] S. Bellon, R. Koschke, G. Antoniol, J. Krinke, and E. Merlo. Comparison and Evaluation of Clone Detection Tools. In *IEEE Transactions on Software Engineering*, volume 33, pages 577–591, 2007.

[4] E. Burd and J. Bailey. Evaluating clone detection tools for use during preventative maintenance. In *Proceedings of the 2th International Workshop on Source Code Analysis and Manipulation (SCAM 2002)*, pages 36–43. IEEE Computer Society, 2002.

[5] L. J. Fülöp, R. Ferenc, and T. Gyimóthy. Towards a Benchmark for Evaluating Design Pattern Miner Tools. In *Proceedings of the 12th European Conference on Software Maintenance and Reengineering (CSMR 2008)*. IEEE Computer Society, Apr. 2008.

[6] L. J. Fülöp, A. Ilia, A. Z. Végh, P. Hegedűs, and R. Ferenc. Comparing and Evaluating Design Pattern Miner Tools. In *ANNALES UNIVERSITATIS SCIENTIARUM DE ROLANDO EÖTVÖS NOMINATAE Sectio Computatorica*. Department of Computer Algebra, Eötvös Loránd University, 2008.

[7] S. Wagner, F. Deissenboeck, M. Aichner, J. Wimmer, and M. Schwalb. An evaluation of two bug pattern tools for java. In *Proceedings of the 1st IEEE International Conference on Software Testing, Verification and Validation (ICST 2008)*, pages 1–8. ACM, 2008.