

DiSL: An Extensible Language for Efficient and Comprehensive Dynamic Program Analysis

Lukáš Marek

Charles University, Czech Republic
lukas.marek@d3s.mff.cuni.cz

Yudi Zheng

Shanghai Jiao Tong University, China
zheng.yudi@sjtu.edu.cn

Danilo Ansaloni

University of Lugano, Switzerland
danilo.ansaloni@usi.ch

Walter Binder

University of Lugano, Switzerland
walter.binder@usi.ch

Zhengwei Qi

Shanghai Jiao Tong University, China
qizhwei@sjtu.edu.cn

Petr Tůma

Charles University, Czech Republic
petr.tuma@d3s.mff.cuni.cz

Abstract

Dynamic program analysis tools support numerous software engineering tasks, including profiling, debugging, and reverse engineering. Prevailing techniques for building dynamic analysis tools are based on low-level abstractions that make tool development tedious, error-prone, and expensive. To simplify the development of dynamic analysis tools, some researchers promoted the use of aspect-oriented programming (AOP). However, as mainstream AOP languages have not been designed to meet the requirements of dynamic analysis, the success of using AOP in this context remains limited. For example, in AspectJ, join points that are important for dynamic program analysis (e.g., the execution of bytecodes or basic blocks of code) are missing, access to reflective dynamic join point information is expensive, data passing between woven advice in local variables is not supported, and the mixing of low-level bytecode instrumentation and high-level AOP code is not foreseen. In this talk, we present DiSL [1], a new domain-specific aspect language for bytecode instrumentation. DiSL uses Java annotation syntax such that standard Java compilers can be used for compiling DiSL code. The language features an open join point model, novel constructs inspired by weave-time evaluation of conditional join points and by staged execution, and access to custom static and dynamic context information. Moreover, the DiSL weaver guarantees complete bytecode coverage. We have implemented several dynamic analysis tools in DiSL, including profilers for the inter- and intra-procedural control

flow, debuggers, dynamic metrics collectors integrated in the Eclipse IDE to augment the static source views with dynamic information, and tools for workload characterization. These tools are concise and perform equally well as implementations using low-level techniques. DiSL has also been conceived as an intermediate language for future domain-specific analysis languages, as well as for AOP languages.

Categories and Subject Descriptors D.3.3 [Programming Languages]: Language Constructs and Features—Frameworks

General Terms Languages, Measurement, Performance

Keywords Bytecode instrumentation, dynamic program analysis, aspect-oriented programming, JVM

Biographies of the Authors

Lukáš Marek is a PhD student under the supervision of Prof. Petr Tůma at Charles University, Czech Republic. He is currently a visiting researcher at University of Lugano, Switzerland, hosted by Prof. Walter Binder. His research topics include dynamic program analysis and performance evaluation.

Yudi Zheng is a MSc. student at Shanghai Jiao Tong University, China. He was a visiting researcher at University of Lugano, Switzerland, hosted by Prof. Walter Binder. He currently studies under the supervision of Prof. Zhengwei Qi.

Danilo Ansaloni holds a MSc. in Computer Science from the University of Modena and Reggio Emilia, Italy. He is now a PhD candidate at the University of Lugano, Switzerland, in the group of Prof. Walter Binder. His research interests include dynamic program analysis, parallel computing, and programming languages.

Walter Binder is a senior assistant professor at the Faculty of Informatics, University of Lugano, Switzerland. He holds

a MSc., a PhD, and a *venia docendi* from the Vienna University of Technology, Austria. Before joining the University of Lugano, he was post-doctoral researcher at the Artificial Intelligence Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. His main research interests are in the area of dynamic program analysis, virtual execution environments, programming languages, and service-oriented computing.

Zhengwei QI is an associate professor at Shanghai Jiao Tong University, China. His research interests include program analysis, model checking, virtual machines, and distributed systems. Now he is a visiting scholar at CMU SCS, hosted by Prof. Edmund M. Clarke.

Petr Tůma is an associate professor and head with the Department of Distributed and Dependable Systems, Charles University. His main research interests include distributed middleware and performance evaluation of software systems.

Acknowledgments

The research presented here was conducted while L. Marek and Y. Zheng were with the University of Lugano. It was supported by the Scientific Exchange Programme NMS-CH (project code 10.165), by a Sino-Swiss Science and Technology Cooperation (SSSTC) Exchange Grant (project no. EG26-032010) and Institutional Partnership (project no. IP04-092010), by the Science and Technology Commission of Shanghai Municipality (project no. 11530700500), by the Swiss National Science Foundation (project CRSII2_136225), and by the Czech Science Foundation (project GACR P202/10/J042).

References

- [1] L. Marek, A. Villazón, Y. Zheng, D. Ansaloni, W. Binder, and Z. Qi. DiSL: a domain-specific language for bytecode instrumentation. In *AOSD '12: Proceedings of the 11th International Conference on Aspect-Oriented Software Development*, Mar. 2012.