



Software Engineering Seminar (WS 2015)

Java Bytecode Instrumentation

Description

Dynamic analysis of software systems requires the instrumentation of application programs with functionality to measure events of interest that occur during program execution, e.g., method calls, runtime exceptions, and variable accesses. For modern programming languages that are executed by virtual machines (Java, .NET, etc.), instrumentation techniques based on bytecode manipulation are available. Bytecode manipulation enables program instrumentation without the need to modify the source code. Modern aspect-oriented programming approaches for these languages are based on bytecode manipulation. This seminar paper should introduce program instrumentation in general, but then focus on approaches based on bytecode instrumentation for the Java virtual machine. The provided references are to be considered a starting point and it is expected to extend the literature search and present a coherent view on the current state of the art in this area.

References

- [1] Asm java bytecode manipulation and analysis. <http://asm.ow2.org/>.
- [2] DiSL: DSL for Java bytecode instrumentation. <http://dis1.ow2.org>.
- [3] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and WilliamG. Griswold. An overview of AspectJ. In *Proceedings of the 2001 European Conference on Object-Oriented Programming (ECOOP 2001)*, volume 2072 of *LNCS*, pages 327–354. Springer, 2001.
- [4] Gregor Kiczales, John Irwin, John Lamping, Jean-Marc Loingtier, Cristina Lopes, Chris Maeda, and Anurag Mendhekar. Aspect-oriented programming. Position paper from the xerox parc aspect-oriented programming project, Xerox Palo Alto Research Center, 1996.
- [5] David J. Lilja. *Measuring Computer Performance: A Practitioner's Guide*. Cambridge University Press, 2005.
- [6] Chi-Keung Luk, Robert Cohn, Robert Muth, Harish Patil, Artur Klauser, Geoff Lowney, Steven Wallace, Vijay Janapa Reddi, and Kim Hazelwood. Pin: Building customized program analysis tools with dynamic instrumentation. In *Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '05)*, pages 190–200. ACM, 2005.
- [7] Lukáš Marek, Alex Villazón, Yudi Zheng, Danilo Ansaloni, Walter Binder, and Zhengwei Qi. DiSL: A domain-specific language for bytecode instrumentation. In *Proceedings of the 11th Annual International Conference on Aspect-Oriented Software Development (AOSD '12)*, pages 239–250. ACM, 2012.
- [8] Amitabh Srivastava and Alan Eustace. Atom: A system for building customized program analysis tools. In *Proceedings of the ACM SIGPLAN 1994 Conference on Programming Language Design and Implementation (PLDI '94)*, pages 196–205. ACM, 1994.

Contacts

Lars Grunske (grunske@informatik.hu-berlin.de)
Software Engineering Group
Institut für Informatik
Humboldt-Universität zu Berlin