What semantics fits with my aspects?

Ralf Lämmel [1,2]

[1] Vrije Universiteit, De Boelelaan 1081a, NL-1081 HV Amsterdam

[2] CWI, Kruislaan 413, NL-1098 SJ Amsterdam

mailto:ralf@cwi.nl
http://homepages.cwi.nl/~ralf/

Abstract: In this talk, we review the available semantics for aspect-oriented programming (AOP), and we connect this theme of recent research to pre-AOP age.

Most AOP semantics are operational or compiler-oriented in style, and they focus on idioms of AspectJ, which is the trend-setting, Java-based AOP language. A typical AOP semantics is based on a down-scaled Java, or perhaps on a simple functional language. The AOP-specific parts of the semantics are normally related to some form of aspect registry to keep track of intercepted join points and associated advice. Still these semantics differ with regard to the cunning peculiarities of describing pointcuts and managing the aspect registry. Furthermore, the semantics might or might not address static typing issues and other static guarantees. In addition to semantics, there are also foundational issues that are being studied, e.g., interpretations of AOP using process algebra (CSP) or the pi-calculus, and the static analysis of aspects for interference.

In fact, AOP foundations have been studied before the existence of AOP: think of structural and behavioural reflection as the most obvious example. Such imports from pre-AOP age also include continuation-passing style, wrapping, parallel programming, algorithmic debugging, and dynamic scoping.

Ongoing work on the semantics of AOP aims at "fluid" AOP, simpler semantical concepts, coverage of practical languages, modular reasoning, aspect composition, and semantics-preserving transformations of aspect-oriented programs.

Keywords: Aspect-Oriented Programming; Foundations; Formal Semantics

References

[1] R. Lämmel. A Semantical Approach to Method-Call Interception. In *Proc. of the 1st International Conference on Aspect-Oriented Software Development (AOSD 2002)*, pages 41–55, Twente, The Netherlands, Apr. 2002. ACM Press.

[2] R. Lämmel. Adding Superimposition To a Language Semantics — Extended Abstract. In G. T. Leavens and C. Clifton, editors, *FOAL'03 Proceedings: Foundations of Aspect-Oriented Languages Workshop at AOSD 2002; Technical Report CS Dept., Iowa State Univ.*, Mar. 2003. 6 pages.

[3] R. Lämmel and K. Ostermann. Semantics of Fluid Lambdas (working title). Ongoing work, Apr. 2004.

[4] R. Lämmel and C. Stenzel. Semantics-Directed Implementation of Method-Call Interception. To appear in Journal IEE Proceedings Software; Special Issue on Unanticipated Software Evolution, 2004.