A Mechanized Framework for Aspects in Isabelle/HOL

F. Kammüller and H. Sudhof, Technische Universität Berlin

We are working on a semantic embedding of aspect-oriented programming in the interactive theorem prover Isabelle/HOL. The goal is to arrive at a framework supporting a flexible investigation of (a) different features of aspect-orientation, like *call* or *cflow*, in relation to (b) various safety and security properties, like standard type-safety, or information flow control.

We have already succeeded in formalizing the ς-calculus in Isabelle/HOL and proved confluence [2].

In this first formalization we simply used lists to represent the labelled fields of objects – a sound abstraction for confluence yet too coarse once it comes to typing. More recent work that we wish to present on the workshop includes the following extensions of our earlier experiments.

- Conservative construction of finite maps and corresponding induction principles.
- Object terms of the ς-calculus based on de Bruijn indices and the aforementioned finite maps.
- A simple type system for the ς-calculus.
- A proof of type safety, i.e. progress and preservation.

Currently we are working on the integration of aspect and weaving functionality into our mechanized model of the ς-calculus. In addition we experiment with type systems for aspects. On this basis we are planning to follow the outline of [3] in constructing a high-level language including all standard aspect-oriented constructs. In a second step we then want to provide a type-preserving compilation to our mechanized ς-calculus thereby guaranteeing an extension of our type safety results to realistic aspect-oriented constructs.

We would like to discuss and exchange experiences centered around the following subjects.

- Nominal Techniques versus de Bruijn indices versus Higher Order Abstract Syntax
- Derivation of executable prototypes
- Structural vs Nominal Type Systems (for Objects with Subtypes)

References

[1] Martín Abadi and Luca Cardelli. *A Theory of Objects.* Springer, New York, 1996.

[2] L. Henrio and F. Kammüller. A Mechanized Model of the Theory of Objects. Accepted at *9th IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems, FMOODS 2007.* To appear in Springer LNCS, 2007.

[3] Jay Ligatti, David Walker and Steve Zdancewic. A type-theoretic interpretation of pointcuts and advice. *Science of Computer Programming: Special Issue on Foundations of Aspect-Oriented Programming.* Springer 2006.