

Contents

Workshop on Formal Techniques for Java-like Programs (FTfJP),
ECOOP 2006 in Nantes, France, July 3-7, 2006 Page

Editorial 3

ARTICLES

Inspector Methods for State Abstraction

By Bart Jacobs and Frank Piessens

5

This paper proposes an approach, based on the Boogie methodology, to the verification of programs that use inspector methods in method contracts and object invariants, to support state abstraction in specifications. However, performing state abstraction in a programming language that allows aliasing through object references poses a framing problem, which is solved thanks to a novel logical encoding of the heap, without breaking information hiding.

Modelling a JVM for polymorphic bytecode

By Giovanni Lagorio

29

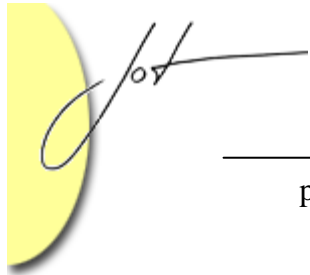
The paper analyzes how polymorphic bytecode can be dynamically linked by presenting a deterministic model of a Java Virtual Machine which interleaves loading and linking steps with execution. In this model, loading and execution phases are basically standard, whereas verification handles also type constraints, which are part of polymorphic bytecode, and resolution blends in verification.

A Parameterized Type System for Simple Loose Ownership Domains

By Jan Schaefer and Arnd Poetzsch-Heffter

61

This paper develops so-called loose domains which abstract over several precise domains in order to overcome the limitations of expressiveness of ownership domains. Furthermore, ownership domains are simplified by reducing the number of domains per object to two and hard-wiring the access permissions between domains. The resulting type system is formalized for anOO core language and proofs of type soundness and of a fundamental accessibility property are



provided.

OUTLOOK

A brief outlook to the next issue

87