Available online at www.sciencedirect.com

SCIENCE DIRECT

Electronic Notes in Theoretical Computer Science

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Invited Talk: Towards a Principled Multi-Language Infrastructure

Zhong Shao

Abstract

Sun's Java architecture introduced a safe virtual machine (VM) in which an ensemble of software components developed independently could smoothly interoperate. The goal of Microsoft's Common Language Runtime (CLR) is to generalize this approach and allow components in many source languages to interoperate safely. CLR supports flexible interoperation by compiling various source languages into a common intermediate language and by using a unified type system. However, the type system in CLR (and Java VM) enforces only conventional type safety in an object-oriented system. Therefore, higher-level specifications (e.g., resource bounds, generalized access control, formal software protocols) cannot be enforced. Because conventional type systems are too inflexible for real applications, developers often bypass the type system, producing code that steps outside the managed part of the VM; such components cannot be verified.

At Yale we have been developing typed common intermediate languages (named FLINT) that can support safely not only the standard object-oriented model, but also higher-order generic (polymorphic) programming and Java-style reflection (introspection). Unlike CLR, our type system is independent of any particular programming model, yet it is capable of expressing all valid propositions and proofs in higher-order predicate logic (so it can be used to capture and verify advanced program properties). The rich type system of FLINT makes it possible to typecheck both compiler intermediate code and low level machine code; this allows typechecking to take place at any phase of compilation, even after optimizations and register allocation. It also leads to a smaller and more extensible VM because low-level native routines that would otherwise be in VM can now be verified and moved into a certified library. This talk describes our vision of the FLINT system, outline our approach to its design, and survey the technologies that can be brought to support its implementation.

10.1016/S1571-0661(05)80449-9