

Computer Science Department

Presents

“Towards a Principled Multi-Language Infrastructure”

**Zhong Shao, Professor
Yale University**

Monday, December 2, 2002
12:30 – 1:30 pm.
CS Main Classroom (room 381)
1890 Maple Avenue, 3rd floor

Abstract:

Component software platforms, such as Microsoft's .NET Common Language Runtime (CLR), provide a secure virtual machine in which components developed independently can smoothly interoperate. CLR supports flexible interoperability 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.

Biography:

Zhong Shao is an Associate Professor in the Department of Computer Science at Yale University. His research interests include programming languages and compilers, typed intermediate representations, secure mobile code, and formal methods. He is a key developer of the Standard ML of New Jersey (SML/NJ) compiler and the architect of the new, higher-order typed intermediate representation (named FLINT) now used in this compiler. He currently leads the Yale FLINT Project which is to develop a principled common compiler infrastructure and runtime system for certified binaries. He received his Ph.D. in Computer Science from Princeton University in 1994.