



Notes for State of the Art Report on Type Systems and Static Analysis for Distributed Programming Languages

J. Rathke

Introduction

- Why use typed languages?
- What is typed static analysis?
- What are the advantages of typed vs non-typed static analysis?
- What are (dependent) type and effect systems?
- What are the benefits to distributed programming languages?

Memory Management and Regions

- Region-based memory management. (Tofte, Talpin)
- Type and effect systems. (Nielsen, Nielsen)

Resource Access Control and Secure Code

- From system F to typed assembly language. (Morrisett, Walker, Crary, Glew)
- Typed memory management in a calculus of capabilities. (Crary, Walker, Morrisett)
- Typing and subtyping for mobile processes. (Pierce, Sangiorgi)
- Resource bound certification. (Crary, Weirich)
- A type system for expressive security policies. (Walker)
- Resource Usage Analysis (Igarashi, Kobayashi)
- Resource access control in systems of mobile agents. (Hennessy, Riely)
- Typing non-uniform concurrent objects. (Ravara, Vasconcelos)
- Explicit behavioural typing for object interfaces. (Najm, Nimour)

Non-interference and secure information flow

- Secure information flow via linear continuations. (Zdancewic, Myers)
- Secure information flow as typed process behaviour. (Honda, Vasconcelos, Yoshida)
- Type-based Information Flow Analysis for JVMML. (Kobayashi, Shirane)
- The security π -calculus and non-interference (Hennessy)
- Non-interference for concurrent programs (Boudol, Castellani)

Types for mobility control

- Types for the Ambient Calculus. (Cardelli, Ghelli, Gordon)
- Secure Safe Ambients. (Bugliesi, Castagna)
- Safe Dynamic Binding in the Join Calculus. (Schmitt)
- Typing mobility in the Seal Calculus - (Castagna, Ghelli, Nardelli)
- Security Types for Mobile Safe Ambients (Dezani-Ciancaglini, Salvo)
- ...

Verification of Cryptoprotocols

- Typing correspondence assertions for communication protocols. (Gordon, Jeffrey)