Structuring Communication

Formally Grounded Languages and Tools for Safe Distributed Programming

Dr Raymond Hu (r.z.h.hu@herts.ac.uk) 13 May 2019

Distributed systems are fundamental to our modern computing and social infrastructures–communication is everywhere. Good software is designed from structured data and control flows: but how should we **structure communications**? Indeed, the development of rigorous *and* practical languages and tools for programming distributed software is lacking, in comparison to the support enjoyed for more traditional sequential and centralised models of computing. Key challenges: **asynchrony** of communications, **separate implementation** of components by different parties, **heterogeneity** of languages and platforms, partial **failures**, ...

We are looking for PhD students to work in the **Scribble** project. Scribble is an open source toolchain being developed by a collaboration of researchers (Univ. of Hertfordshire, Imperial College London, Brunel University London) and industry partners (Red Hat, Estafet, Cognizant, the Ocean Observatories Initiative) for **structured, type-safe distributed programming**. The Scribble toolchain comprises two main elements:

- Specification and validation of a multiparty message passing protocol from a high-level, global perspective–a *choreography*;
- Projection of the protocol onto each endpoint, giving a *protocol- and endpoint-specific API* that promotes compliant distributed implementations in the target language.

The key principles behind Scribble stem from the theory of **Multiparty Session Types** (MPST).

Scribble research is very active! Recent publications have featured at academic conferences on programming languages, compiler construction and software engineering: POPL '19, CC '18, FASE '17, ECOOP '17, FASE '16. Scribble featured at the F# Exchange '18 developer conference.

Scribble is especially fun for people enthusiastic about:

- Rigorous approaches to practical communications-oriented **programming**
- Practically-motivated development and extensions of **formal** concurrent languages (π-calculus)

Currently, the actively supported target languages of Scribble are: Java, Go and F# (.NET)

Potential Topics

Hot topics:

- Scribble and MPST for Function-as-as-Service and Serverless Computing
- Distributed software tracing: Scribble-based run-time monitoring of Istio service behaviour

Other key topics and themes:

- Types-based model checking of message passing systems
- Code generation for distributed APIs
- Protocols with parametrised and dynamic participants
- Formal specification of real-world application protocols
- Type-safe event-driven programming
- Type providers and on-demand code generation
- Type refinements in protocol specifications
- Types for fault-tolerance and exceptions

If you are interested in any of the above topics, or would simply like to know more:

Contact: Raymond Hu r.z.h.hu@herts.ac.uk

Scribble-Java tutorial: http://www.scribble.org/docs/scribble-java.html

Fork me on GitHub

Scribble: Describing Multi Party Protocols