Safe and Reliable Programming for Distributed Applications and Cloud Computing

Distributed systems are fundamental to our modern computing and social infrastructures. The Internet, online banking and shopping, social media, video streaming, document and media processing in the cloud, … – a huge part of our daily lives depend on **communications and interactions between globally distributed computers**. To date, however, the understanding and development of *programming languages* has centred on traditional models of sequential computation localised within a single computer. Programmers are typically offered only weak or second-class language support for dealing with the intrinsic challenges of distributed computation in the real world: concurrency and non-determinism, asynchrony, communication safety, heterogeneity of languages and platforms, protocol compliance, partial failures, ...

We are looking for research students to work on the design and implementation of **languages and tools for safe and reliable programming of distributed applications and systems**. They will build on and further our active research directions, including the Scribble project:

- Scribble is an open source toolchain [1] for the specification, implementation and verification of **distributed protocols** based on ideas from **multiparty session types** (a type system for concurrent message passing) and **model checking**.
- It is developed by a collaboration of **academic researchers** (e.g., Univ. of Hertfordshire, Imperial College London, Brunel University London) and **industry partners** (Red Hat, Estafet, Cognizant, and others).
- It has been applied to many languages, including Java, Go, .NET (F#), Scala and Python.
- This active and recent work is presented and published in conferences and journals on programming and software engineering: e.g.,
  OOPSLA '20, POPL '19, CC '18, FASE '17, ECOOP '17 and FASE '16.

Many potential topics!

- Real-world applications of Scribble and session types: e.g., serverless computing, distributed tracing.
- Extensions of session types and the Scribble tools: e.g., reliability, fault-tolerance.
- Advanced programming techniques and type systems: e.g., F# type providers, F* refinement types.

A suitable candidate will be interested in one or more of:

- **Practical programming** in a language such as Java, Kotlin, Go, .NET (e.g., F# or C#).
- Implementation of programming languages: e.g., **compilers** and **code generation** tools.
- **Distributed systems** and cloud technologies: e.g., Kubernetes, Istio, Knative, Spark.
- **Type systems** for message passing concurrency, e.g., session types.

Students will receive training and guidance on areas they are less familiar with.

Interested in working on any of the above topics or others in this area? (open to discussion)

Feel free to contact: **Dr. Raymond Hu** (r.z.h.hu@herts.ac.uk)

[1] scribble-java tutorial. http://www.scribble.org/docs/scribble-java.html

Scribble: Describing Multi Party Protocols