Let it Recover: Multiparty Protocol-Induced Recovery

Rumyana Neykova, Nobuko Yoshida
Imperial College London
Session Type Mobility Group

www.mrg.doc.ic.ac.uk
Us ∈ Mobility Research Group

MobilityReadingGroup
π-calculus, Session Types research at Imperial College

NEWS

SELECTED PUBLICATIONS

2017


Julien Lange, Nicholas Ng, Bernardo Toninho, Nobuko Yoshida: Fencing off Go: Liveness and Safety for Channel-based Programming. POPL 2017.


http://mrg.doc.ic.ac.uk/
**OOI Collaboration**

- **TCS’16:** Monitoring Networks through Multiparty Session Types. Laura Bocchi, Tzu-Chun Chen, Romain Demangeon, Kohei Honda, Nobuko Yoshida
- **LMCS’16:** Multiparty Session Actors. Rumyana Neykova, Nobuko Yoshida
- **FMSD’15:** Practical interruptible conversations: Distributed dynamic verification with multiparty session types and Python. Romain Demangeon, Kohei Honda, Raymond Hu, Rumyana Neykova, Nobuko Yoshida
- **TGC’13:** The Scribble Protocol Language. Nobuko Yoshida, Raymond Hu, Rumyana Neykova, Nicholas Ng
Scribble is a language to describe application-level protocols among communicating systems. A protocol represents an agreement on how participating systems interact with each other. Without a protocol, it is hard to do meaningful interaction: participants simply cannot communicate effectively, since they do not know when to expect the other parties to send data, or whether the other party is ready to receive data. However, having a description of a protocol has further benefits. It enables verification to ensure that the protocol can be implemented without resulting in unintended consequences, such as deadlocks.

<table>
<thead>
<tr>
<th>Describe</th>
<th>Verify</th>
<th>Project</th>
<th>Implement</th>
<th>Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scribble is a language for describing multiparty protocols from a global, or endpoint neutral, perspective.</td>
<td>Scribble has a theoretical foundation, based on the Pi Calculus and Session Types, to ensure that protocols described using the language are sound, and do not suffer from deadlocks or livelocks.</td>
<td>Endpoint projection is the term used for identifying the responsibility of a particular role (or endpoint) within a protocol.</td>
<td>Various options exist, including (a) using the endpoint projection for a role to generate a skeleton code, (b) using session type APIs to clearly describe the behaviour, and (c) statically verify the code against the projection.</td>
<td>Use the endpoint projection for roles defined within a Scribble protocol, to monitor the activity of a particular endpoint, to ensure it correctly implements the expected behaviour.</td>
</tr>
</tbody>
</table>
module examples;

global protocol HelloWorld(role Me, role World) {
  hello() from Me to World;
  choice at World {
    goodMorning1() from World to Me;
  } or {
    goodMorning1() from World to Me;
  }
}
Interactions with Industries

Strange Loop
SEPTEMBER 15-17 2016 / PEABODY OPERA HOUSE / ST. LOUIS, MO

Adam Bowen @adamnbowen · Sep 15
I didn’t even know that session types existed an hour ago, but thanks to Nobuko Yoshida’s great talk at #pwIconf, I want to learn more.

Nobuko Yoshida
Imperial College, London

DoC researcher to speak at Golang UK conference
by Vicky Kapogianni
20 July 2016

DoC researcher to speak at industry-focused Golang UK conference on results of concurrency research

@nicholascwng rocking on @GolangUKconf about static deadlock detection in #golang #gouk16

The Golang UK Conference
Interactions with Industries

F#unctional Londoners Meetup Group
6 days ago · 6:30 PM
Session Types with Fahd Abdeljallal
43 Members

Synopsis: Session types are a formalism to codify the structure of a communication, using types to specify the communication protocol used. This formalism provides the...

Distributed Systems vs. Compositionality
Dr. Roland Kuhn
@rolandkuhn — CTO of Actyx

Current State
- behaviors can be composed both sequentially and concurrently
- effects are not yet tracked
- Scribble generator for Scala not yet there
- theoretical work at Imperial College, London (Prof. Nobuko Yoshida & Alceste Scalas)
Go concurrency verification research at DoC grabs headline

A paper by DoC researchers at POPL on Go concurrency verification was featured in a tech blog and generates a buzz outside of the research community.

A paper by researchers at the department was recently featured in the morning paper, a blog by venture capitalist Adrian Colye, which summarises an important, influential, topical or otherwise interesting paper in the field of computer science every weekday in an easily digestible way by non-researchers. On the 2 Feb 2017 issue of the morning paper, it was highlighted as "the true spirit of POPL (Principles of Programming Languages)".
Selected Publications 2016/2017

• **[FoSSaCS’17]** Julien Lange, NY: On the Undecidability of Asynchronous Session Subtyping.
• **[FASE’17]** Raymond Hu, NY: Explicit Connection Actions in Multiparty Session Types.
• **[CC’17]** Rumyana Neykova, NY: Let It Recover: Multiparty Protocol-Induced Recovery.
• **[POPL’17]** Julien Lange, Nicholas Ng, Bernardo Toninho, NY: Fencing off Go: Liveness and Safety for Channel-based Programming.
• **[FPL’16]** Xinyu Niu, Nicholas Ng, Tomofumi Yuki, Shaojun Wang, NY, Wayne Luk: EURECA Compilation: Automatic Optimisation of Cycle-Reconfigurable Circuits.
• **[ECOOP’16]** Alceste Scala, NY: Lightweight Session Programming in Scala
• **[CC’16]** Nicholas Ng, NY: Static Deadlock Detection for Concurrent Go by Global Session Graph Synthesis.
• **[FASE’16]** Raymond Hu, NY: Hybrid Session Verification through Endpoint API Generation.
• **[TACAS’16]** Julien Lange, NY: Characteristic Formulae for Session Types.
• **[POPL’16]** Dominic Orchard, NY: Effects as sessions, sessions as effects.
Selected Publications 2016/2017

• [CC’16] Nicholas Ng, NY: Static Deadlock Detection for Concurrent Go by Global Session Graph Synthesis.
Let’s Start

Let it Recover:
Multiparty Protocol-Induced Recovery