

Imperial College London

Activity Report

Nobuko Yoshida

2019

The Mobility Reading Group at Imperial College

Our research group is **specialised in π -calculus** and **session types** — both **theory** and **applications**

<http://mrg.doc.ic.ac.uk/>

SESSION TYPES
 π MobilityReadingGroup
 π -calculus, Session Types research at Imperial College

Home People Publications Grants Talks Tutorials Tools Awards Kohei Honda

NEWS

The paper 'Language primitives and type discipline for structured communication-based programming' by Kohei Honda, Vasco T. Vasconcelos and Makoto Kubo, has received the ETAPS 2019 Test-of-Time Award.

» more

11 Apr 2019

The paper 'A fully abstract game semantics for general references' by Samson Abramsky, Kohei Honda and Guy McCusker, has received the LICS Test-of-Time Award.

SELECTED PUBLICATIONS

2019

Simon Castellan, Nobuko Yoshida: [Causality in Linear Logic](#). FoSSaCS 2019 : 150 - 168.

David Castro, Raymond Hu, Sung-Shik Jongmans, Nicholas Ng, Nobuko Yoshida: [Distributed Programming Using Role Parametric Session Types in Go](#). POPL 2019 : 29:1 - 29:30.

Tiago Cogumbreiro, Raymond Hu, Francisco Martins, Nobuko Yoshida: [Dynamic deadlock verification for general barrier synchronisation](#). TOPLAS : 1 - 38.

Session types in programming languages

(Elements of) the **multiparty session types** theory have been **implemented in many languages and libraries**, e.g.:

Java, Go, Python, Scala, C, MPI-C, F#, F*, Erlang, Haskell, OCaml, Dotty3, PureScript, TypeScript and Rust.



Industrial Collaborations

- ▶ Red Hat, **IBM**
- ▶ **Cognizant**, Thoughtworks, Estafet, **RChain**
- ▶ Ocean Observatories Initiative, **November Group**
- ▶ VMWare, Povotal, **Weaveworks**
- ▶ **Actyx**, **Barclays**, JPMorgan, AstraZeneca, **Preqin**
- ▶ Amazon, Google, Microsoft, Facebook

Impacts (I): Scribble



www.scribble.org

The screenshot shows the homepage of the Scribble website. At the top, there is a navigation bar with links for Home, Getting Started, Downloads, Documentation, and Community. Below this is a large blue header with the text "Scribble: Describing Multi Party Protocols". The main content area features a paragraph of text explaining the purpose of Scribble. Below the text are five columns, each with a title, an icon, and a brief description of a feature or concept.

Home Getting Started Downloads Documentation - Community -

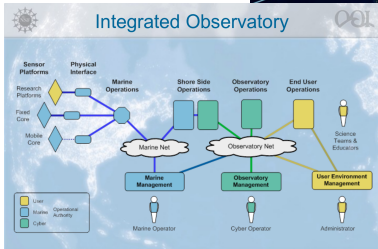
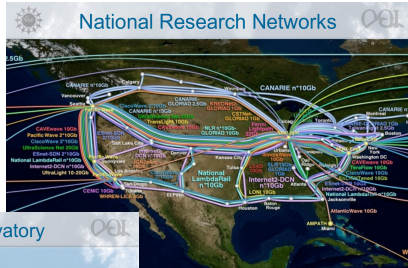
Scribble: Describing Multi Party Protocols

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.

- Describe** Scribble is a language for describing multiparty protocols from a global, or endpoint neutral, perspective.
- Verify** 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.
- Project** Endpoint projection is the term used for identifying the responsibility of a particular role (or endpoint) within a protocol.
- Implement** 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.
- Monitor** 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.

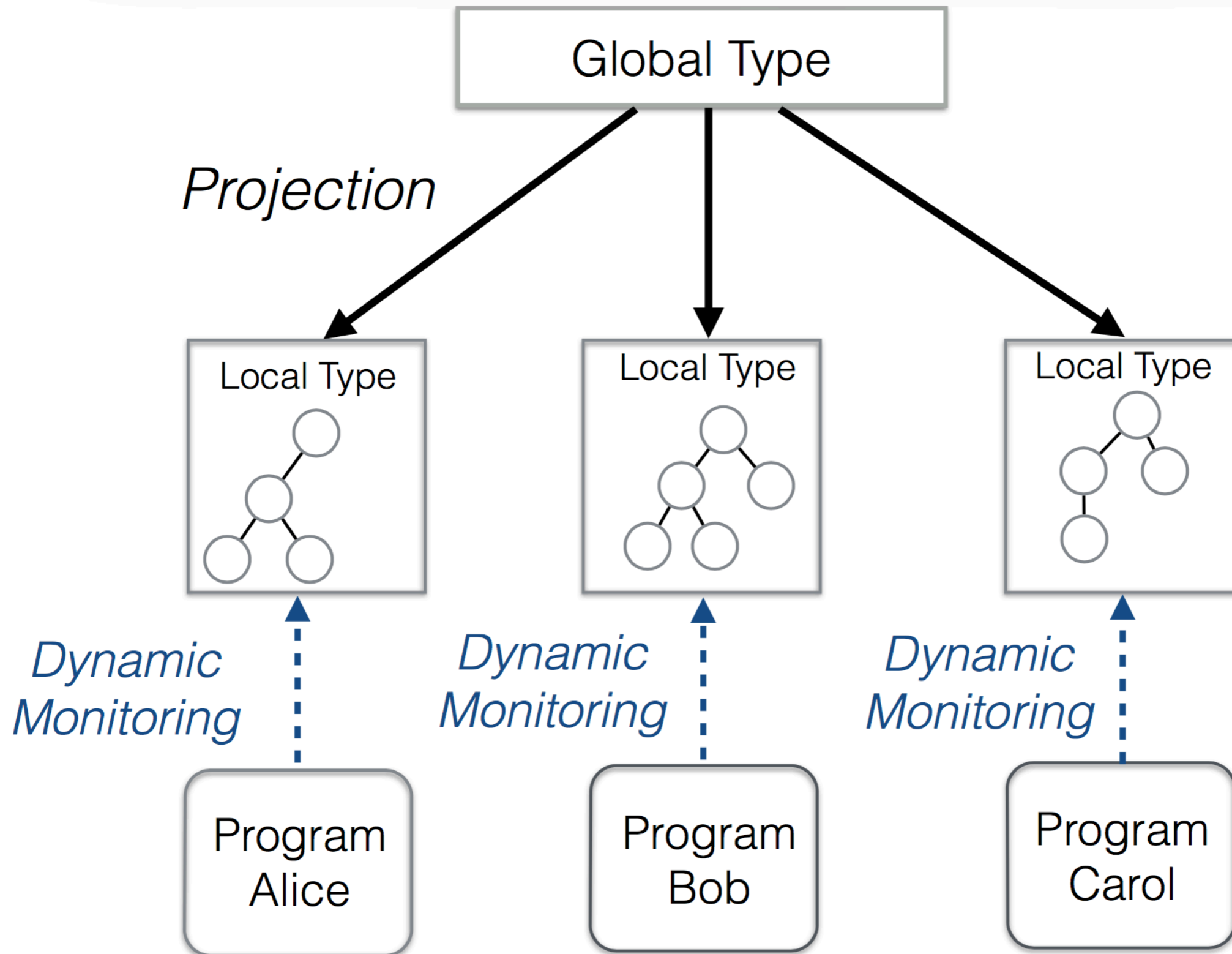
Impacts (II): Ocean Observatories Initiative

Collaboration: use session types on top of OOI network to **guarantee global safety**



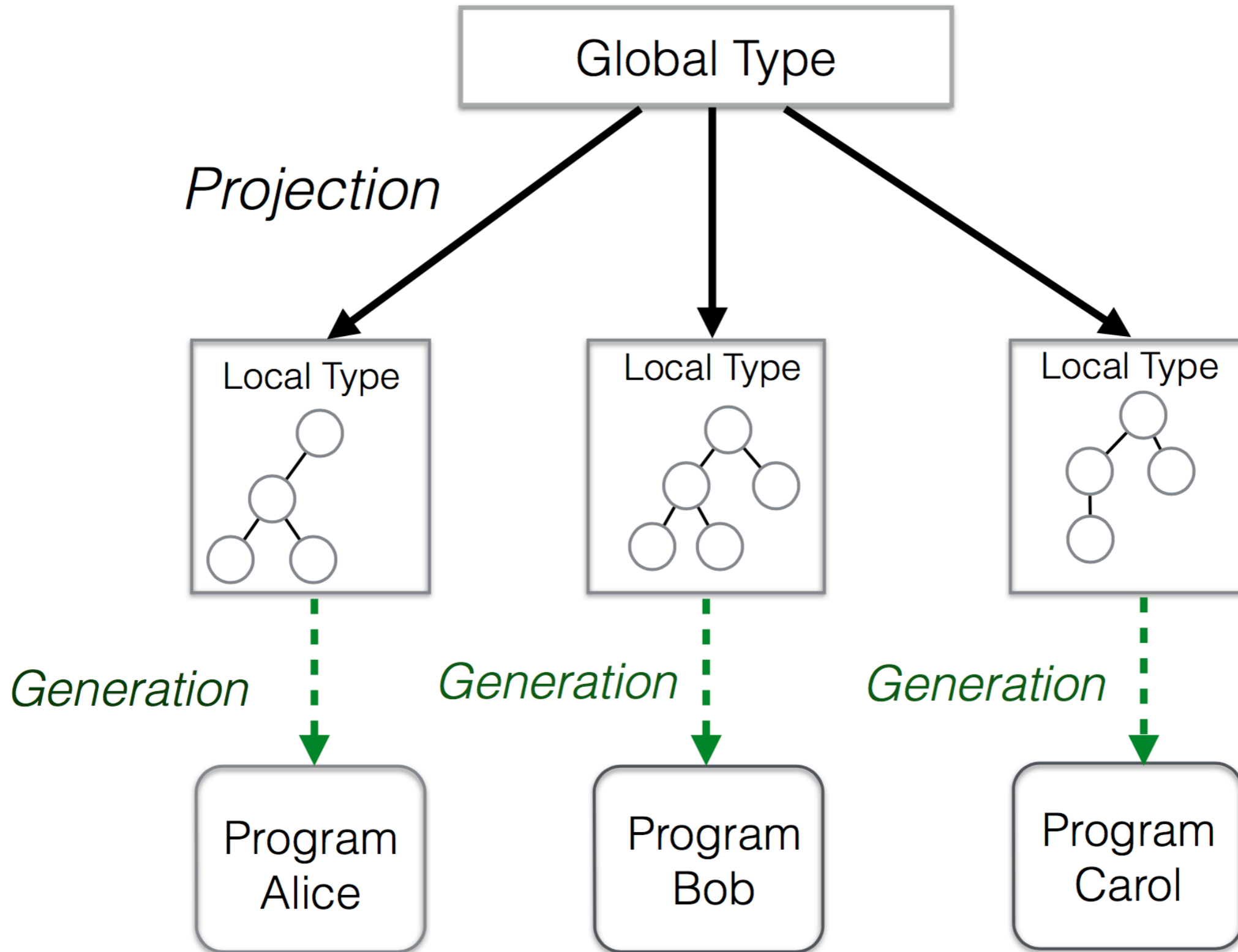
Dynamic Monitoring

[RV'13, COORDINATION'14, FMDS'15, LMCS'17, CC'17]



Code Generation

[CC'15, FASE'16, FASE'17]



Java API Generation [FASE'16]

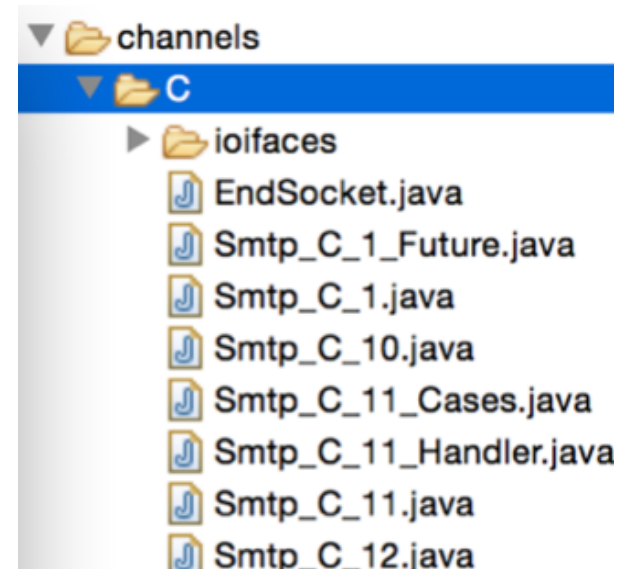


RFC 821 August 1982
Simple Mail Transfer Protocol

TABLE OF CONTENTS

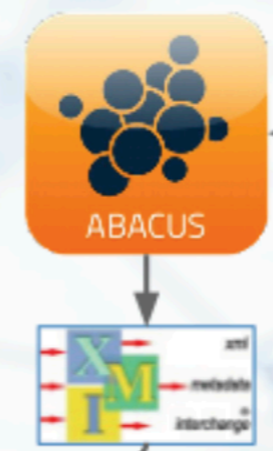
1.	INTRODUCTION	14
2.	THE SMTP MODEL	14
3.	THE SMTP PROCEDURE	14
3.1.	Mail	14
3.2.	Forwarding	14
3.3.	Verifying and Expanding	14
3.4.	Sending and Mailing	14
3.5.	Opening and Closing	14
3.6.	Relaying	14
3.7.	Domains	14
3.8.	Changing Roles	14
4.	THE SMTP SPECIFICATIONS	14
4.1.	SMTP Commands	14
4.1.1.	Command Semantics	14
4.1.2.	Command Syntax	14
4.2.	SMTP Replies	14
4.2.1.	Reply Codes by Function Group	14
4.2.2.	Reply Codes in Numeric Order	14
4.3.	Sequencing of Commands and Replies	14
4.4.	State Diagrams	14
4.5.	Details	14
4.5.1.	Minimum Implementation	14
4.5.2.	Transparency	14
4.5.3.	Sizes	14

□



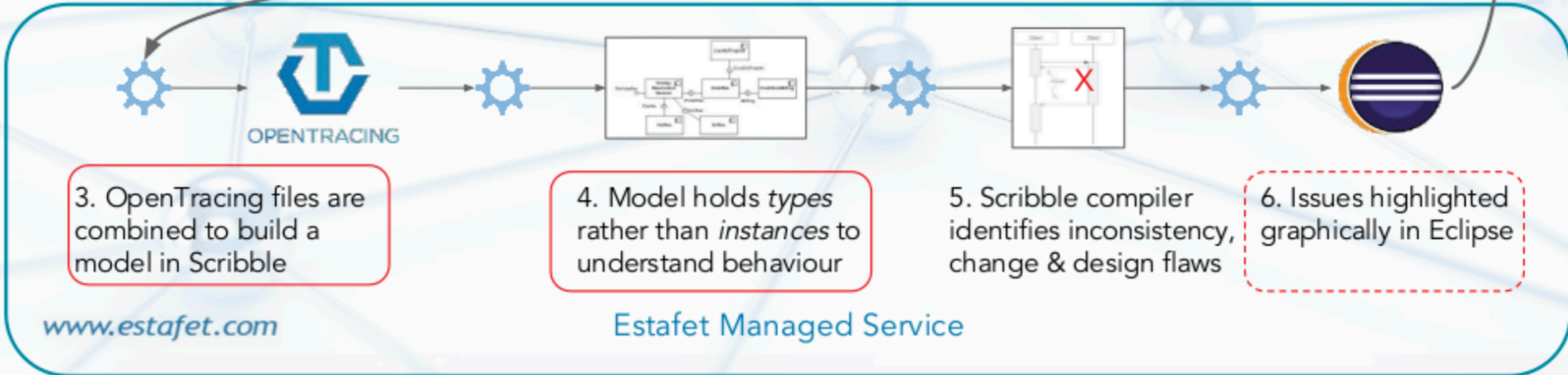
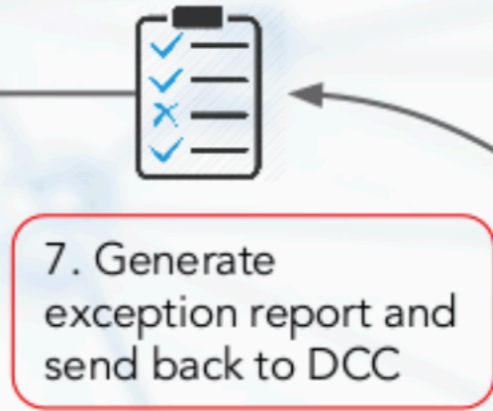
```
.send(Smtplib.S, new DataLine("Session  
.send(Smtplib.S, new EndOfData())  
.receive(Smtplib.S, Smtplib._250, new Buf  
.S  
● send(S role, Mail m) : Smtplib_C_11 - Smtplib_C_10  
● send(S role, Quit m) : EndSocket - Smtplib_C_10
```

Scribble – Proving a distributed design



1. All design work takes place in ABACUS, DCC's enterprise architecture tool. This can export standard XMI files (an open standard for UML5)

2. XMI is converted into OpenTracing format for consumption by managed service



3. OpenTracing files are combined to build a model in Scribble

4. Model holds *types* rather than *instances* to understand behaviour

5. Scribble compiler identifies inconsistency, change & design flaws

6. Issues highlighted graphically in Eclipse

Relationship with Communicating AUTOMATA

ESOP 12

Characterisation

ICALP 13

Synthesis

CONCUR 15

Timed

Automata

POPL 15

Graphical

Synthesis

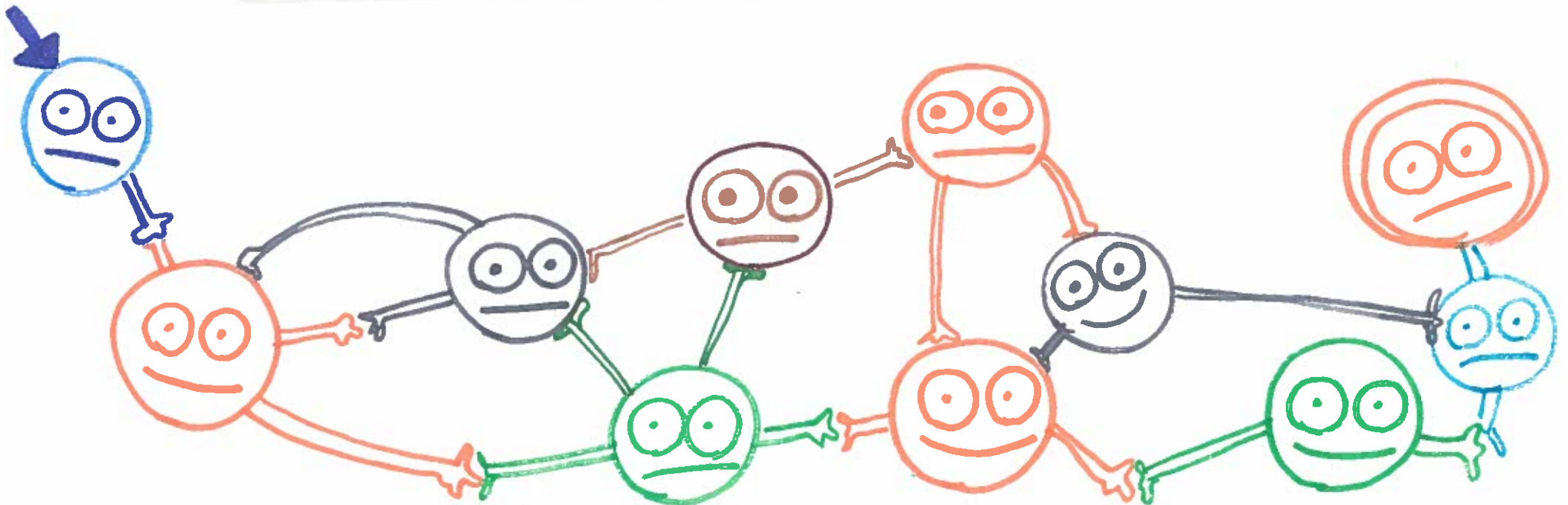
TACAS 16

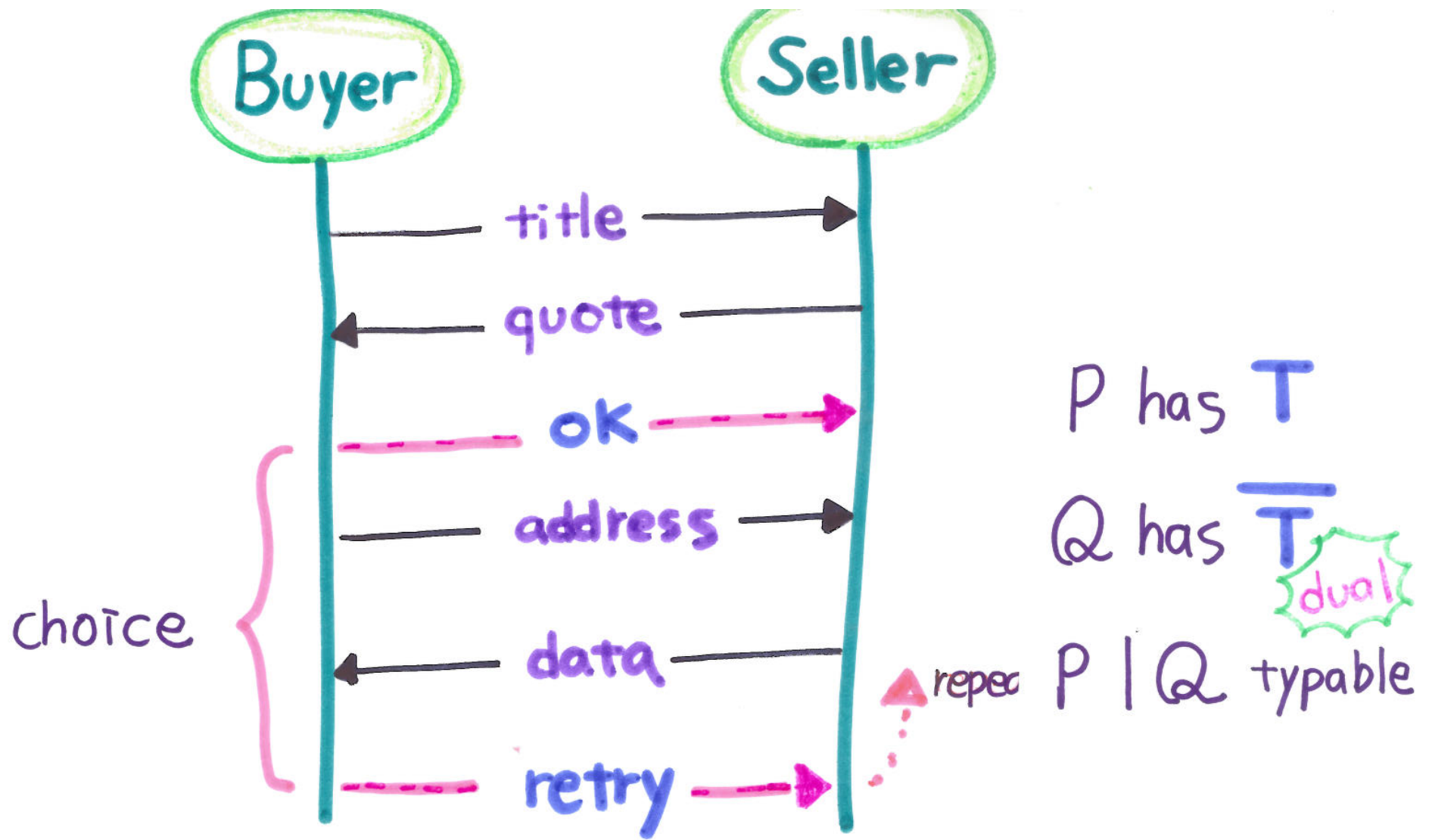
Subtyping

+ Model-checking

FOSSACS 17

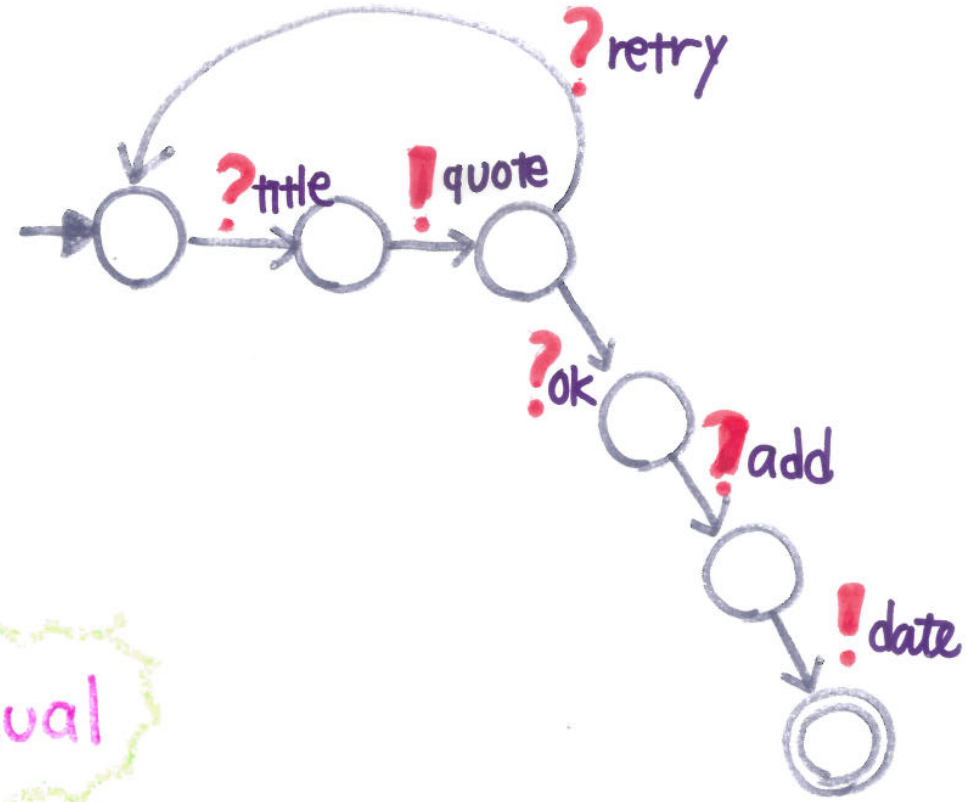
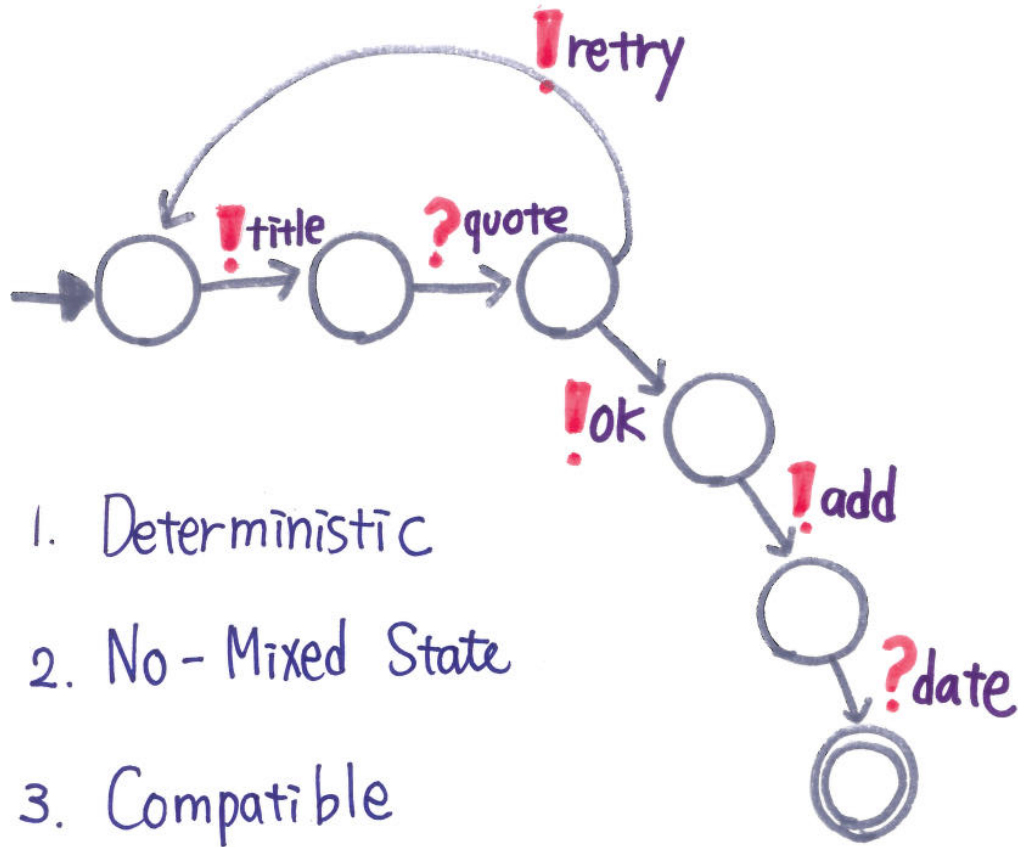
Undecidability





nt! Title ; ? Quote ; ! { ok: ! Add ; ? Date, **retry** : t }

nt? Title ; ! Quote ; ? { ok: ? Add ; ! Date, **retry** : t }

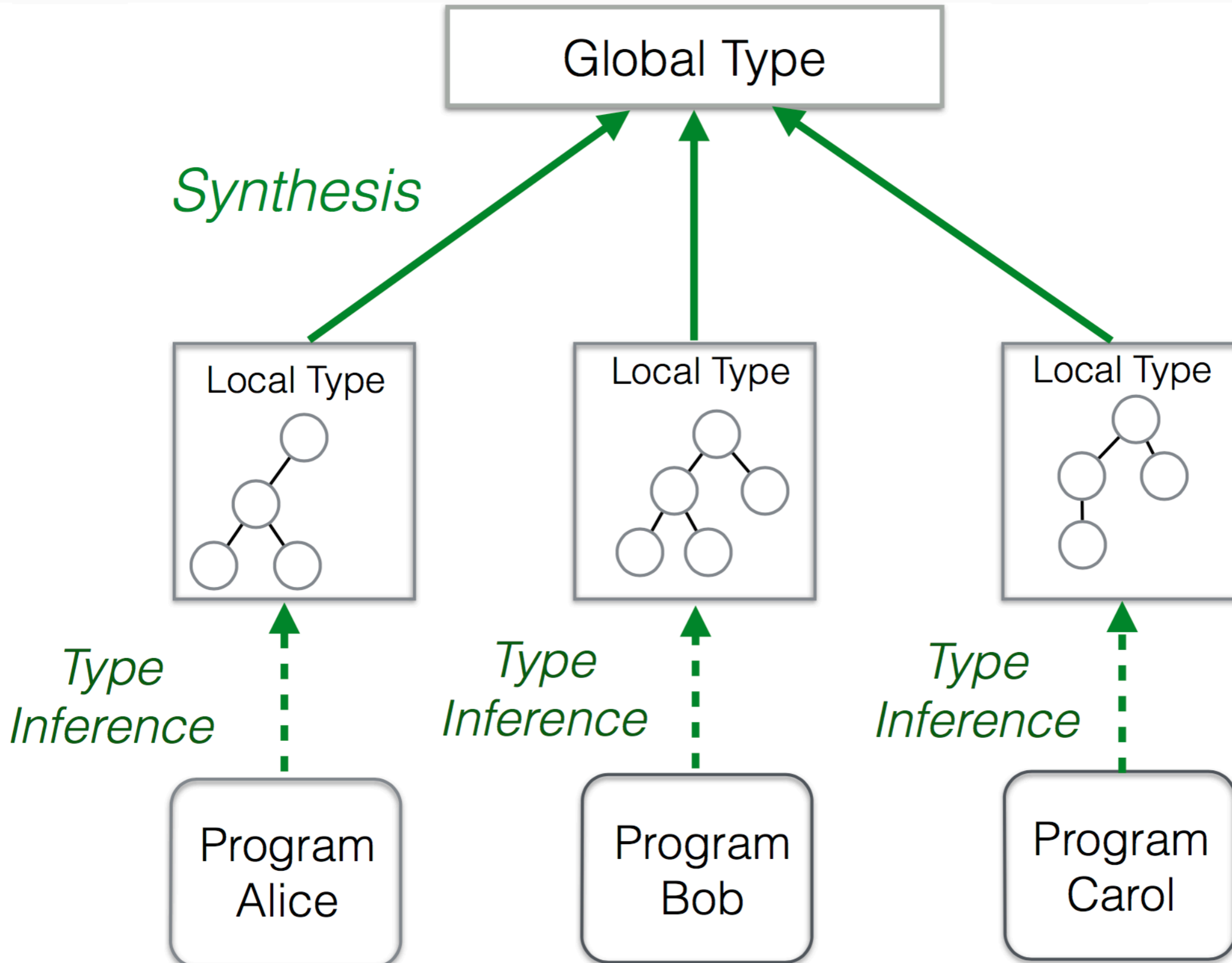


dual

[Gouda et al 1986] Two compatible machines without mixed states which are deterministic satisfy deadlock-freedom.

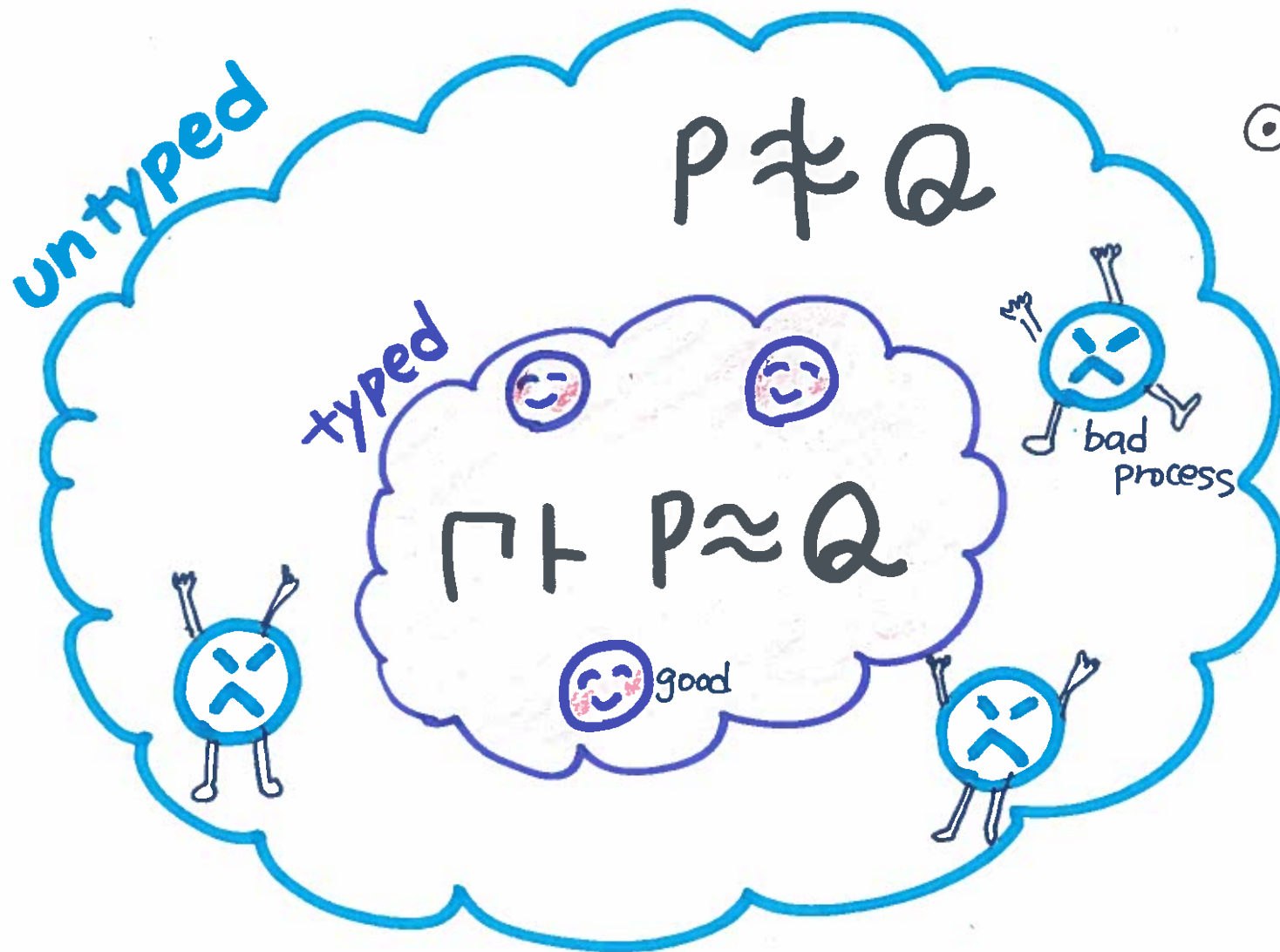
Synthesis

[ICALP'13, POPL'15, CONCUR'15, TACAS'16, CC'16]



Typed Semantics in π 1991 \rightarrow

IO-subtyping, Linear types, Secure Information Flow, ...



- ⊙ Correctness of Encoding
- ⊙ Limit environment \vdash
 \Rightarrow Equate more processes
- ⊙ Compositional

GLOBALLY GOVERNED SESSION SEMANTICS

CONCUR 15
LMCS



Dimitrios
Kouzapas
Glasgow



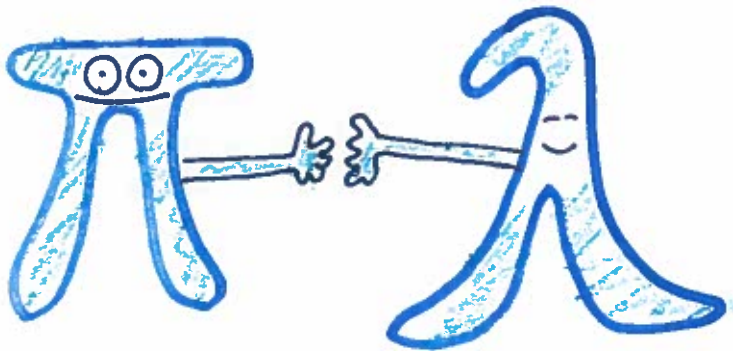
Nobuko
Yoshida

Imperial
College London



HOT and

TYPES



CONCUR 16 Characteristic
Bisimulations

ESOP 16 Expressiveness

ACTA INFORMATICA

on Polymorphic
and Sessions
Functions

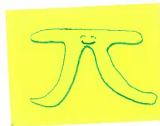
a Tale of Two
Encodings

Bernardo Toninho @Nova

Nobuko Yoshida @Imperial

on Polymorphic
and Sessions

Function 



a Tale of Two

Fully
Abstract

Encodings

Bernardo Toninho @ Nova

Nobuko Yoshida @ Imperial

on Polymorphic Sessions and Functions



a Tale of Two Fully Abstract Encodings

Bernardo Toninho @ Nova

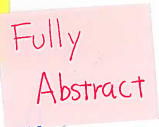
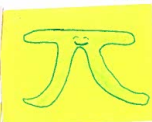
Nobuko Yoshida @ Imperial

on Polymorphic

Sessions

and

Function



a Tale

of

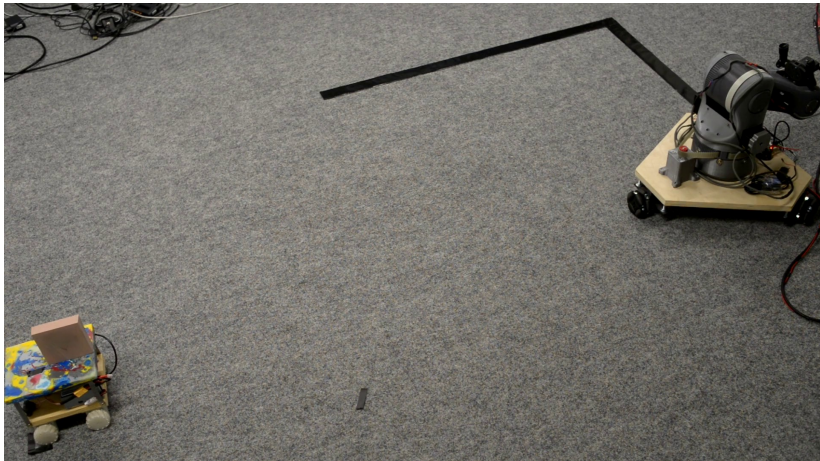
Two

Encodings

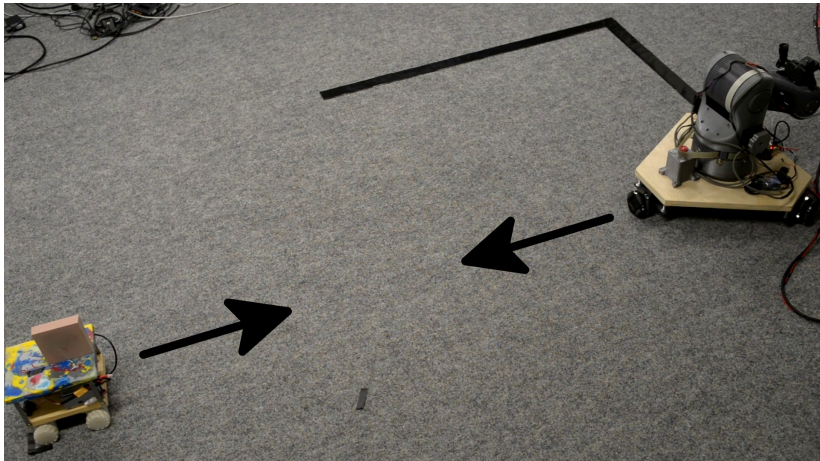
Bernardo Toninho @ Nova

Nobuko Yoshida @ Imperial

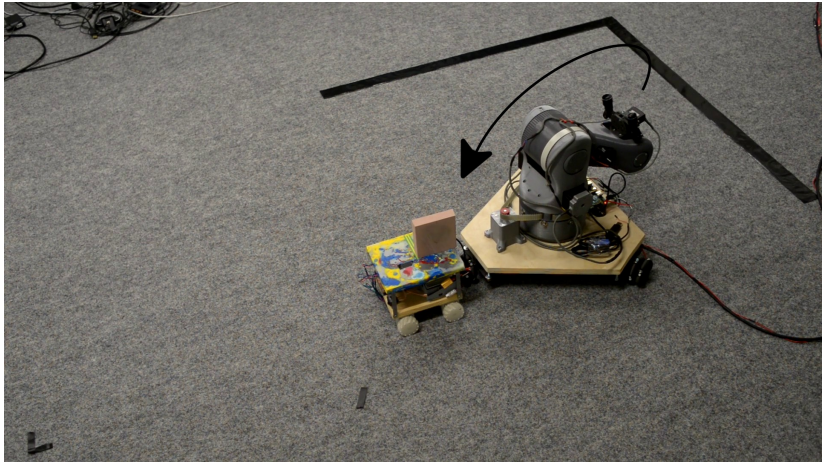
Impacts (V): Session Types for Robotics



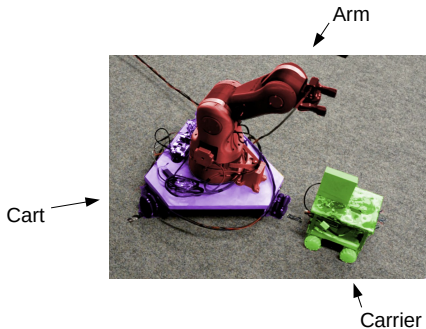
Impacts (V): Session Types for Robotics



Impacts (V): Session Types for Robotics

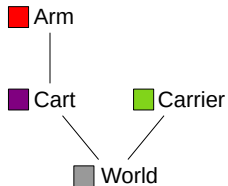
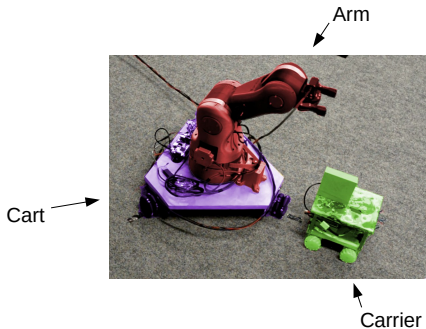


Impacts (V): Session Types for Robotics

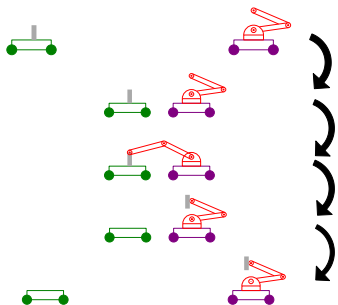


Cart & Arm:
Two robots attached together that
act as “one” robot (communication)

Impacts (V): Session Types for Robotics



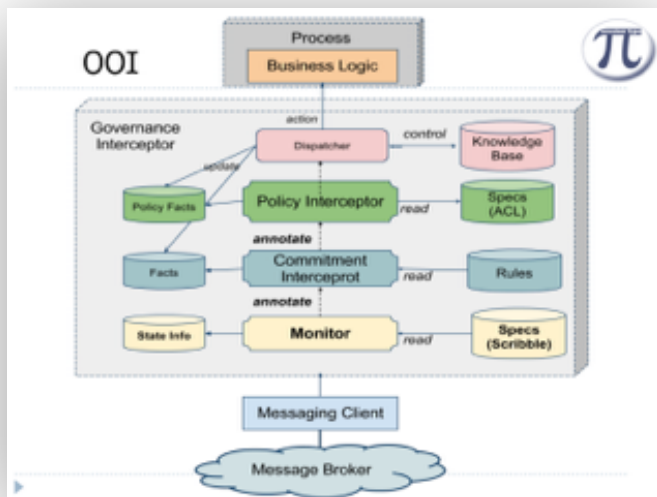
Impacts (V): Session Types for Robotics



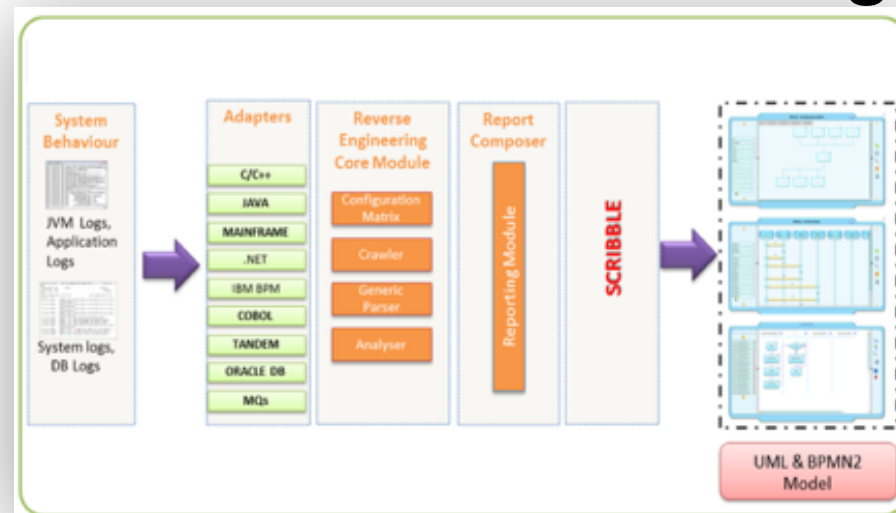
```
Cart → Arm : fold(unit).dt(Cart : idle, Carrier : idle, Arm : fold).  
Arm → Cart : ok(unit).Cart → Carrier : ok(unit).  
dt(Cart : move, Carrier : move, Arm : idle).  
Carrier → Cart : ok(unit).Cart → Arm : grab(unit).  
dt(Cart : idle, Carrier : idle, Arm : grip).  
Arm → Cart : ok(unit).Cart → Carrier : ok(unit).  
dt(Cart : move, Carrier : move, Arm : idle).  
Cart → Arm : done(unit).Cart → Carrier : done(unit).end
```

Session Type based Tools

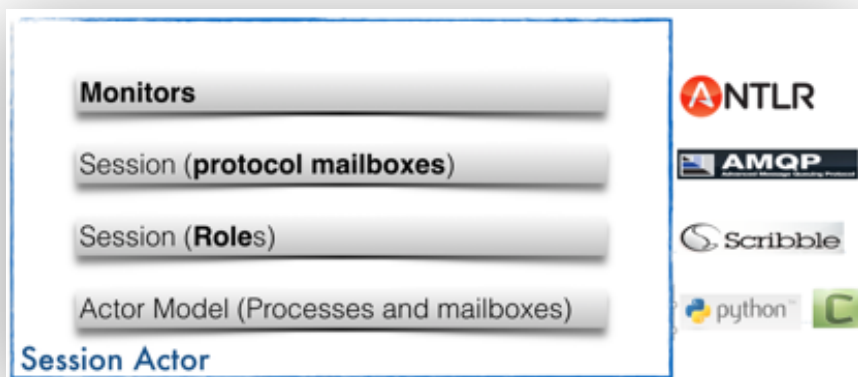
OOI Governance



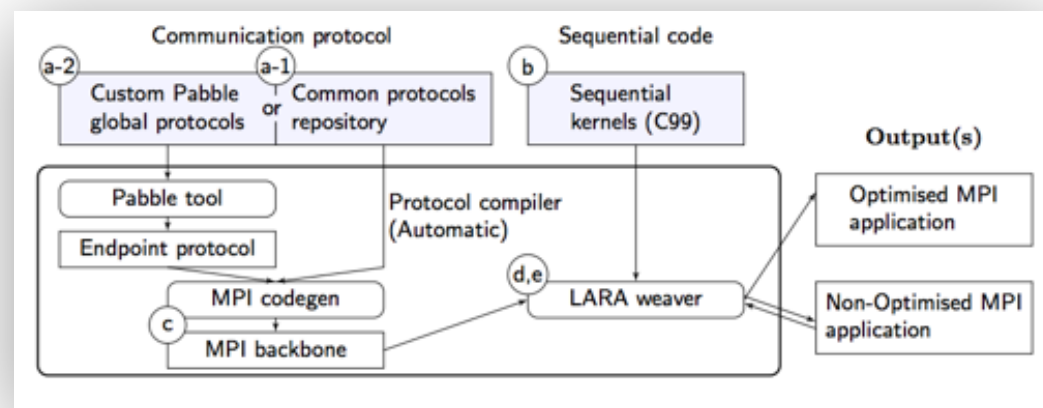
ZDLC: Process Modeling



Actor Verification



MPI code generations



Public Engagements/Lectures/Keynote Talks

▶ Lectures

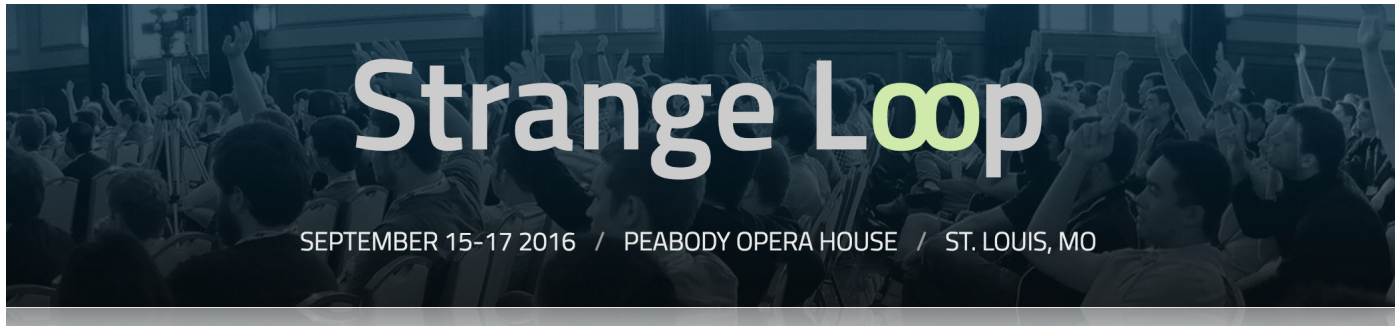
- ▶ **Courses:** Imperial College London, Concurrent Processes (4th MEng, MSc), Keio University (since 2016), Taipei University (2018), L'Aquila (2014)
- ▶ **Tutorials/Schools:** BeheAPI (2019), CGO'18 (Venice), FM'16 (Cyprus), Saint-Malo (France) 2016, Bertinoro (Italy) 2015, and BETTY (Cyprus 2016/Lovran 2014), and POPL'14.

- ▶ **Industrial Seminars and Conferences:** Many, e.g. Newton IT seminar (42 companies attended, London).

- ▶ **STEM:** ACM PLMW Co-Chair (POPL 2019, 2020), London Hoppers, etc.

- ▶ **Invited Keynote Talks:** ICDCIT'20 (India), CONFESTA'18 (Beijing), TLT'17 (Torino), OPT'17 (Venice), STRANGE-LOOP'16 (St. Louis), RTA-TLCA'14 (Venice) and TGC'13 (Buenos Aires)

Interactions with Industries



Nobuko Yoshida
Imperial College, London

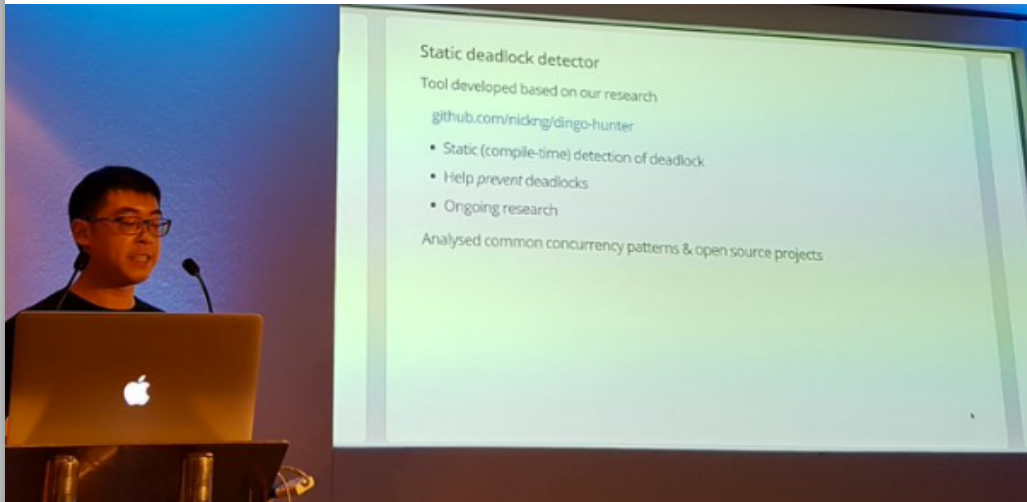


Adam Bowen @adamnbowen · Sep 15

I didn't even know that session types existed an hour ago, but thanks to Nobuko Yoshida's great talk at [#pwlconf](#), I want to learn more.

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

[Click here to add content](#)



.@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... [LEARN MORE](#)

Distributed Systems

vs.

Compositionality

Dr. Roland Kuhn

@rolandkuhn — CTO of Actyx

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)

Publications

- ▶ NY GS: from 4500 to 10140 (MPST 742); 4833 since 2014
- ▶ **75 Conferences Papers**
 - ▶ POPL'15, POPL'16, POPL'17, POPL'19 (A), POPL'19 (B), POPL'19 (C)
 - ▶ ECOOP'19, ECOOP'17, ECOOP'16, OOPSLA'15
 - ▶ ESOP'19, ESOP'18, ESOP'17, ESOP'16, CONCUR'19, CONCUR'15 (A), CONCUR'15 (B), CONCUR'15 (C), CONCUR'14, CONCUR'13 (A), CONCUR'13 (B)
 - ▶ LICS'18, FOSSACS'19, FOSSACS'18, FOSSACS'17, ICALP'13
 - ▶ **Artifacts Evaluations:** All submissions were accepted.
 - ▶ **New Venues:** PLDI'19, ICSE'18, FSE'19, FASE'17, FASE'16, CAV'19, TACAS'16, FPL'16, PPOPP'15, CC'15, CC'16, CC'17, CC'18.
- ▶ **33 Journal Articles** including JACM, ACM TOPLAS, Information and Computations, and Logical Methods of Computations.

Selected Publications [2018-2019]

- [PLDI19] [Alceste Scalas](#), NY, Elias Benussi: [Verifying message-passing programs with dependent behavioural types](#).
- [CAV19] [Julien Lange](#), NY: [Verifying Asynchronous Interactions via Communicating Session Automata](#).
- [CONCUR19] Mario Bravetti, Marco Carbone, [Julien Lange](#), NY, Gianluigi Zavattaro: [A Sound Algorithm for Asynchronous Session Subtyping](#)
- [FSE19] Nicola Atzei, Massimo Bartoletti, Stefano Lande, NY, Roberto Zunino: [Developing secure Bitcoin contracts with BitML](#)
- [ECOOP19] Rupak Majumdar, Marcus Pirron, NY, and Damien Zufferey, Motion Session Types for Robotic Interactions
- [FoSSaCs19] [Simon Castellan](#), NY: [Causality in Linear Logic](#)
- [ESOP19] [Laura Bocchi](#), Maurizio Murgia, Vasco T. Vasconcelos, NY: [Asynchronous Timed Session Types](#)
- [POPL19] [Simon Castellan](#), NY: [Two Sides of the Same Coin: Session Types and Game Semantics](#)
- [POPL19] [David Castro](#), [Raymond Hu](#), Sung-Shik Jongmans, [Nicholas Ng](#), NY: [Distributed Programming Using Role Parametric Session Types in Go](#)
- [POPL19] [Alceste Scalas](#), [Nobuko Yoshida](#): [Less Is More: Multiparty Session Types Revisited](#)
- [POPL19] [Bernardo Toninho](#), NY: [Interconnectability of Session Based Logical Processes](#)
- [ICSE18] [Julien Lange](#), [Nicholas Ng](#), [Bernardo Toninho](#), NY: [A Static Verification Framework for Message Passing in Go using Behavioural Types](#)
- [LICS18] [Romain Demangeon](#), NY: [Causal Computational Complexity of Distributed Processes](#)
- [FoSSaCs18] [Bernardo Toninho](#), [Nobuko Yoshida](#): [Depending On Session Typed Process](#)
- [ESOP18] [Bernardo Toninho](#), NY: [On Polymorphic Sessions And Functions: A Tale of Two \(Fully Abstract\) Encodings](#)
- [CC18] [Rumyana Neykova](#), [Raymond Hu](#), NY, Fahd Abdeljallal: [A Session Type Provider: Compile-time API Generation for Distributed Protocols with Interaction Refinements in F#](#)

Selected Publications [2018-2019]

- [PLDI19] [Alceste Scalas](#), NY, Elias Benussi: [Verifying message-passing programs with dependent behavioural types](#).
- △ [CAV19] [Julien Lange](#), NY: [Verifying Asynchronous Interactions via Communicating Session Automata](#).
- △ [CONCUR19] Mario Bravetti, Marco Carbone, [Julien Lange](#), NY, Gianluigi Zavattaro: [A Sound Algorithm for Asynchronous Session Subtyping](#)
 - [FSE19] [Nicola Atzei](#), [Massimo Bartoletti](#), [Stefano Lande](#), NY, [Roberto Zunino](#): [Developing secure Bitcoin contracts with BitML](#)
 - [ECOOP19] [Rupak Majumdar](#), [Marcus Pirron](#), NY, and [Damien Zufferey](#), [Motion Session Types for Robotic Interactions](#)
- [FoSSaCs19] [Simon Castellan](#), NY: [Causality in Linear Logic](#)
- [ESOP19] [Laura Bocchi](#), [Maurizio Murgia](#), [Vasco T. Vasconcelos](#), NY: [Asynchronous Timed Session Types](#)
- [POPL19] [Simon Castellan](#), NY: [Two Sides of the Same Coin: Session Types and Game Semantics](#)
- [POPL19] [David Castro](#), [Raymond Hu](#), [Sung-Shik Jongmans](#), [Nicholas Ng](#), NY: [Distributed Programming Using Role Parametric Session Types in Go](#)
- [POPL19] [Alceste Scalas](#), [Nobuko Yoshida](#): [Less Is More: Multiparty Session Types Revisited](#)
- [POPL19] [Bernardo Toninho](#), NY: [Interconnectability of Session Based Logical Processes](#)
- [ICSE18] [Julien Lange](#), [Nicholas Ng](#), [Bernardo Toninho](#), NY: [A Static Verification Framework for Message Passing in Go using Behavioural Types](#)
 - [LICS18] [Romain Demangeon](#), NY: [Causal Computational Complexity of Distributed Processes](#)
- [FoSSaCs18] [Bernardo Toninho](#), [Nobuko Yoshida](#): [Depending On Session Typed Process](#)
- [ESOP18] [Bernardo Toninho](#), NY: [On Polymorphic Sessions And Functions: A Tale of Two \(Fully Abstract\) Encodings](#)
 - [CC18] [Ramyana Narayana](#), [Raymond Hu](#), NY, [Fahd Abdeljallal](#): [A Session Type Provider: Compile-time API Generation for Distributed Protocols with Interaction Refinements in F#](#)