M & M's

Contributions to
SESSION TYPES

Theory and
PRACTICE

Mariangela
Mario

NOBUKO YOSHIDA
Two Extensions of Curry’s Type Inference System

by F. Cardone and M. Coppo
ARIANGIOLA

photo by J.W. Klop 1978

ETAPS '02 ???

NY

MD
MARIANGiola

photo by J.W. Klop 1978

ETAPS’02 ???

Mini Skirt

NY

MD
Session Types in a Nutshell

**SESSION** = STRUCTURED SEQUENCE OF COMMUNICATION

```plaintext
send(int).send(int).receive(bool)
```

“…Session Types **structure** a **series of interactions** in a simple and concise syntax and ensure **type safe communication**.”
A Protocol

- Protocol: Buyer-Seller
- Description: Alice buying a book

 send(string).receive(int).⊕{ok: send(string).receive(date), quit:end}
receive(string).send(int).&{ok: receive(string).send(date), quit: end}
Are we compatible?

\texttt{send(int).send(int).receive(bool)}

\texttt{receive(int).receive(int).send(bool)}

\textit{It is all about duality!}
Are we compatible?

\texttt{receive(int).send(int).receive(bool)}

MUTE

\texttt{receive(int).receive(int).send(bool)}
What is a **type safe communication**?

- **Communication safety**
  - No communication mismatch

- **Session Fidelity**
  - Communications follow the desired protocol

- **Progress**
  - No deadlock/stuck in a session
Session Types and Objects

Shophia Drossopoulou

Mariangiola

Dimitris

NY
Session Types and Objects

TGC 2005

Shophia Drossopoulou

Mariangiola

Dimitris

ECOOP 2006
Session Types and Objects

TGC 2005

Mariangiola

Shophia Drossopoulou

ECOOP 2006

Dimitris

NY
Session Types and Objects

First Asynchronous Session

Shophia Drossopoulou

Mariangiola

Dimitris

FMOODS 07
Session-based Distributed Programming in Java

Raymond Hu, Nobuko Yoshida

Kohei Honda

Imperial College London

Queen Mary University of London
Implementing Customer(4)

protocol p {
    begin.
    ![
        ![String>.
        ?(int)
    ]*
    !{
        ACCEPT: { 
            ![Address>.
            ?(Date)
        },
        REJECT: { }
    }
}

SJSocket s = SJSocket.create(p, ...);

s.request();
s.outwhile(...) {
    s.send("PARIS/EUROSTAR");
    cost = s.receive();
}
if (...) {
    s.outbranch(ACCEPT) {
        s.send(...);
        date = s.receive();
    }
} else {
    s.outbranch(REJECT) { }
}
Dialogue between Industry and Academia

Binary Session Types [PARL’94, ESOP’98]

Milner, Honda and Yoshida joined W3C WS-CDL (2002)

Formalisation of W3C WS-CDL [ESOP’07]

Scribble at π⁴ Technology
CDL Equivalent

• Basic example:

```java
package HelloWorld {
    roleType YouRole, WorldRole;
    participantType You{YouRole}, World{WorldRole};
    relationshipType YouWorldRel between YouRole and WorldRole;
    channelType WorldChannelType with roleType WorldRole;

    choreography Main {
        WorldChannelType worldChannel;

        interaction operation=hello from=YouRole to=WorldRole
            relationship=YouWorldRel channel=worldChannel {
                request messageType=Hello;
            }
    }
}
```

Dr Gary Brown (Pi4 Tech) in 2007
Scribble Protocol

- "Scribbling is necessary for architects, either physical or computing, since all great ideas of architectural construction come from that unconscious moment, when you do not realise what it is, when there is no concrete shape, only a whisper which is not a whisper, an image which is not an image, somehow it starts to urge you in your mind, in so small a voice but how persistent it is, at that point you start scribbling" - Kohei Honda 2007

- Basic example:

```cpp
protocol HelloWorld {
    role You, World;
    Hello from You to World;
}
```
Dialogue between Industry and Academia

Binary Session Types [PARL’94, ESOP’98]

↓

Milner, Honda and Yoshida joined W3C WS-CDL (2002)

↓

Formalisation of W3C WS-CDL [ESOP’07]

↓

Scribble at \(\text{π}^4\) Technology

↓

Multiparty Session Types [POPL’08]
Dialogue between Industry and Academia

Binary Session Types [PARL’94, ESOP’98]

вшись

Milner, Honda and Yoshida joined W3C WS-CDL (2002)

 весну

Formalisation of W3C WS-CDL [ESOP’07]

 весну

Scribble at π4 Technology

 весну

Multiparty Session Types [POPL’08]
Session Types Overview

- Global session type
  \[ G = A \rightarrow B : \langle U_1 \rangle. B \rightarrow C : \langle U_2 \rangle. C \rightarrow A : \langle U_3 \rangle \]

- Local session type
  - Slice of global protocol relevant to one role
  - Mechanically derived from a global protocol
  \[ T_A = !\langle B, U_1 \rangle. ?\langle C, U_3 \rangle \]

- Process language
  - Execution model of I/O actions by session participants
  - Mechanically derived from a global protocol
  \[ P_A = a[A](x). x!\langle B, u_1 \rangle. x?(C, y) \]

- (Static) type checking for communication safety and progress
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.
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;
    }
}
Multiparty Session Types (1)

POPL ’08
Alice → Bob: $K \langle \text{Nat} \rangle$
Bob → Alice: $K' \langle \text{Int} \rangle$

CONCUR ’08
Alice → Bob: $\langle \text{Nat} \rangle$
Bob → Alice: $\langle \text{Int} \rangle$

Much Simpler

30 Minutes

TCP Connection

A

B

C

Connection
Multiparty Session Types (1)

POPL ‘08
Alice → Bob: $K \langle \text{Nat} \rangle$
Bob → Alice: $K' \langle \text{Int} \rangle$

CONCUR ‘08
Alice → Bob: $\langle \text{Nat} \rangle$
Bob → Alice: $\langle \text{Int} \rangle$

Much Simpler

30 Minutes

TCP Connection

No channel!
Multiparty Session Types (1)

**POPL’08**

Alice → Bob: \( K \langle \text{Nat} \rangle \)

Bob → Alice: \( K' \langle \text{Int} \rangle \)

**CONCUR’08**

Alice → Bob: \( \langle \text{Nat} \rangle \)

Bob → Alice: \( \langle \text{Int} \rangle \)

Much Simpler

No channel!

My 4th 180+

TCP Connection

30 Minutes
CONCUR '08

Global Progress of MPST

\[ \downarrow \]

NY

Dezani

\[ \downarrow \]

Bettini

Coppo

D'Antoni

Lucia

COORDINATION '13

Inference of Global Progress in MPST

\[ \downarrow \]

MSCS

\[ \downarrow \]

Gentle Introduction of MPST

Padovani
Us ∈ Mobility Research Group

MobilityReadingGroup

π-calculus, Session Types research at Imperial College

NEWS

Our recent work Fencing off Go: Liveness and Safety for Channel-based Programming was summarised on The Morning Paper blog.

2 Feb 2017
Weizhen passed her viva today, congratulations Dr. Yang!

24 Jan 2017
Mariangiola Dezani-Ciancaglini, a long-term collaborator with our group working on Session Types turns 70 today, more details here.

23 Dec 2016
Rumyana passed her viva today.

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/
Ocean Observatories Initiative

- A NSF project (400M$, 5 Years) to build a cyberinfrastructure for observing oceans around US and beyond.
- Real-time sensor data constantly coming from both off-shore and on-shore (e.g. buoys, submarines, under-water cameras, satellites), transmitted via high-speed networks.
Ocean Observatories Initiative
https://confluence.oceanobservatories.org/display/syseng/CIAD+COI+OV+Negotiate+Protocol
type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;

global protocol Negotiate(role Consumer as C, role Producer as P) {

}
type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;

global protocol Negotiate(role Consumer as C, role Producer as P) {
    propose(SAP) from C to P;

    choice at P {
        accept() from P to C;
        confirm() from C to P;
    } or {
        reject() from P to C;
    } or {
        propose(SAP) from P to C;
    }
}
OOI agent negotiation 4/5

type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;

global protocol Negotiate(role Consumer as C, role Producer as P) {
    propose(SAP) from C to P;

    choice at P {
        accept() from P to C;
        confirm() from C to P;
    } or {
        reject() from P to C;
    } or {
        propose(SAP) from P to C;
        choice at C {
            accept() from C to P;
            confirm() from P to C;
        } or {
            reject() from C to P;
        } or {
            propose(SAP) from C to P;
        }
    }
}
OII agent negotiation 5/5 (recursion)

type <yml> "SAPDoc1" from "SAPDoc1.yml" as SAP;

global protocol Negotiate(rrole Consumer as C, role Producer as P) {
    propose(SAP) from C to P;
    rec X {
        choice at P {
            accept() from P to C;
            confirm() from C to P;
        } or {
            reject() from P to C;
        } or {
            propose(SAP) from P to C;
            choice at C {
                accept() from C to P;
                confirm() from P to C;
            } or {
                reject() from C to P;
            } or {
                propose(SAP) from C to P;
                continue X;
            }
        }
    }
}

15 / 42
Local protocol projection (Negotiation Consumer)

// Global
propose(SAP) from C to P;
rec START {
  choice at P {
    accept() from P to C;
    confirm() from C to P;
  } or {
    reject() from P to C;
  } or {
    propose(SAP) from P to C;
    choice at C {
      accept() from C to P;
      confirm() from P to C;
    } or {
      reject() from C to P;
    } or {
      propose(SAP) from C to P;
      continue START;
    }
  }
}

// Projection for Consumer
propose(SAP) to P;
rec START {
  choice at P {
    accept() from P;
    confirm() to P;
  } or {
    reject() from P;
  } or {
    propose(SAP) from P;
    choice at C {
      accept() to P;
      confirm() from P;
    } or {
      reject() to P;
    } or {
      propose(SAP) to P;
      continue START;
    }
  }
}
FSM generation (Negotiation Consumer)
Welcome to Release 2 of the Ocean Observatories Initiative Observatory (OOI). You already have access to many OOI features and real-time data. Just click on something that looks interesting on this page to start using the OOI as our guest.

For personalized services, such as setting up notifications and preserving settings for your next visit, create a free account by clicking on "Create Account" at the top of the page.

DATA LEGEND
- Temperature
- Salinity
- Oxygen
- Density
- Currents
- Sea Surface Height (SSH)
- Chlorophyll
- Turbidity
- pCO2
- Autonomy
- Other

RECENT UPDATES
- **01 h Oregon Coast North Salinity** 2012-01-10 23:55:55 Type Event Description goes here Note goes here
- **02 h Oregon Coast South 100m therm** 2012-01-10 23:55:55 Type Event Description goes here Note goes here
- **03 h Oregon Coast South 100m pH** 2012-01-10 23:55:55 Type Event Description goes here Note goes here
- **04 h Oregon Coast North Salinity** 2012-01-10 23:55:55 Type Event Description goes here Note goes here

National Science Foundation working with Consortium for Ocean Leadership

Funding for the Ocean Observatories Initiative is provided by the National Science Foundation through a Cooperative Agreement with the Consortium for Ocean Leadership. The OOI Program Implementing Organizations are funded through sub-awards from the Consortium for Ocean Leadership.
DOZEN MONITORING and ADAPTATIONS

by Coppo & Dezani

based on Session Types

[PPDP'14] [PPDP'14] [SOCA'15] [JLAMP'16]
[PLACES'16] [FAC'16]

etc etc
Dynamic Monitoring

[RV’13, COORDINATION’14, FMSD’15, LMCS’17, CC’17]
Type Checking
[OOPSLA’15, ECOOP’16, ECOOP’17, COORDINATION’17]

Diagram:
- Global Type
  - Projection
    - Local Type
      - Type Checking
        - Program Alice
      - Type Checking
        - Program Bob
      - Type Checking
        - Program Carol
Code Generation
[CC’15, FASE’16, FASE’17]
Synthesis

[ICALP’13, POPL’15, CONCUR’15, TACAS’16, CC’16]
Session Type based Tools

OIO Governance

ZDLC: Process Modeling

Actor Verification

MPI code generations
Interactions with Industries

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

Adam Bowen @adambowen · 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.

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)
Behavioural Type-Based Static Verification Framework for

Julian Lange  Nicholas Ng  Bernardo Toninho  Nobuko Yoshida
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)".
Java API Generation [FASE’16]
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.
7. Generate exception report and send back to DCC.
LIVE Researchers

Future

70 years old

Past

20 years old

Information Flow and Security
[CONCUR'10]...

Reversible Computation
[CONCUR'17]...

Preciseness
[PPDP'14] [LMCS]...
Background: session subtyping

Types and Subtypes for Client-Server Interactions

Simon Gay and Malcolm Hole

(ESOP’99)
Background: session subtyping

Types and Subtypes for Client-Server Interactions

Simon Gay and Malcolm Hole

(ESOP’99)

Global Principal Typing in Partially Commutative Asynchronous Sessions

Dimitris Mostrous¹, Nobuko Yoshida¹, and Kohei Honda²

(ESOP’09)
Background: session subtyping

Types and Subtypes for Client-Server Interactions
Simon Gay and Malcolm Hole
(ESOP’99)

Global Principal Typing in Partially Commutative Asynchronous Sessions
Dimitris Mostrous\(^1\), Nobuko Yoshida\(^1\), and Kohei Honda\(^2\)
(ESOP’09)

On the Preciseness of Subtyping in Session Types
Tzu-Chun Chen
Università di Torino
chen@di.unito.it

Mariangiola Dezani-Ciancaglini
Università di Torino
dezani@di.unito.it

Nobuko Yoshida
Imperial College London
yoshida@doc.ic.ac.uk
(PPDP’14)
Background: session subtyping

Types and Subtypes for Client-Server Interactions

Simon Gay and Malcolm Hole

(ESOP’99)

Global Principal Typing in Partially Commutative Asynchronous Sessions

Dimitris Mostrous¹, Nobuko Yoshida¹, and Kohei Honda²

(ESOP’09)

*Other completeness results*  Subtyping of recursive types requires algorithms for checking subtype relations, as discussed in [32, Chapter 21]. These algorithms need to be proved sound and complete with respect to the definition of the corresponding subtyping, as done for example in [7, 12, 33]. Algorithms for checking the synchronous and asynchronous subtypings of the present paper can be easily designed.
LIVE Researchers

- Information Flow and Security [CONCUR'10]...
- Reversible Computation [CONCUR'17]...
- Preciseness [PPDP'14] [LMCS]...

120 years old (Future)
70 years old
20 years old (Past)
HAPPY 70

Simona
Mariaugiola
Mario