Multiparty Session Types and their Applications to Large Distributed Systems

Nobuko Yoshida
Imperial College London
In collaboration with:

Matthew Arrott (OOI)
Gary Brown (Red Hat)
Stephen Henrie (OOI)
Bippin Makoond (Cognizant)
Michael Meisinger (OOI)
Matthew Rawlings (ISO TC68 WG4/5)
Alexis Richardson (RabbitMQ/VMware)
Steve Ross-Talbot (Cognizant)

and all our academic colleagues
Laura Bocchi, Tzu-Chun Chen, Romain Demangeon, Pierre-Malo Denielóu, Luca Fossati, Dimitrios Kouzapas, Rumyana Neykova, Nicholas Ng, Weizhen Yang
Outline

➤ Background

➤ Multiparty Session Types [POPL’08]

➤ Scribble and Applications to a Large-scale Cyberinfrastructure

➤ Monitoring Theory

➤ Summary
Communication is Ubiquitous

- Internet, the WWW, Cloud Computing, the next-generation manycore chips, message-passing parallel computations, large-scale cyberinfrastructure for e-Science.
- The way to organise software is increasingly based on communications.
- Applications need *structured* series of communications.

**Question**

- How to formally abstract/specify/implement/control communications?
Communication is Ubiquitous

- Internet, the WWW, Cloud Computing, the next-generation manycore chips, message-passing parallel computations, large-scale cyberinfrastructure for e-Science.
- The way to organise software is increasingly based on communications.
- Applications need *structured* series of communications.

Question

- How to formally abstract/specify/implement/control communications?
Communication is Ubiquitous

➤ Internet, the WWW, Cloud Computing, the next-generation manycore chips, message-passing parallel computations, large-scale cyberinfrastructure for e-Science.

➤ The way to organise software is increasingly based on communications.

➤ Applications need *structured* series of communications.

➤ Question ➞ Multiparty session type theory

➤ How to formally abstract/specify/implement/control communications?
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.
Challenges

➤ The need to specify, catalogue, program, implement and manage *multiparty message passing protocols*.

➤ Communication assurance

➤ Correct message ordering and synchronisation
➤ Deadlock-freedom, progress and liveness
➤ Dynamic message monitoring and recovery
➤ Logical constraints on message values

➤ Shared and used over a long-term period (e.g. 30 years in OOI).
Why Multiparty Session Types?

 организм (2002): *Types are the leaven of computer programming; they make it digestible.*

 ➞ Can describe communication protocols as *types*
 ➞ Can be materialised as *new communications programming languages* and *tool chains*.

 ➰ *Scalable* automatic verifications (deadlock-freedom, safety and liveness) without *state-space explosion problems* (*polynomial time complexity*).

 ➰ Extendable to *logical verifications* and flexible *dynamic monitoring*. 
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 \( \pi^4 \) 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;
            }
    }
}
```
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:

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

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

\[ \downarrow \]

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

\[ \downarrow \]

Formalisation of W3C WS-CDL [ESOP’07]

\[ \downarrow \]

Scribble at π^4 Technology

\[ \downarrow \]

Multiparty Session Types [POPL’08]
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

⇒

Multiparty Session Types [POPL’08]
Binary Session Types: Buyer-Seller Protocol

Buyer

-> title

-> quote

-- ok

-> address

-> date

-- quit

Seller

branch
branch

Buyer

- title
- quote
- ok
- address
- date
- quit

Seller

String; Int; \oplus\{ok: String; ?Date; end, quit: end\}
! String ; ? Int ; ⊕ { OK : ! String ; ? Date ; end , quit : end }

dual ? String ; ! Int ; ⊗ { OK : ? String ; ! Date ; end , quit : end }
Multiparty Session Types

Buyer 1 -> Seller
- title
- quote
- quote ÷ 2

Buyer 2
- address
- date

Ok
Multiparty Session Types

Buyer 1

Seller

Buyer 2

- title
- quote
- quote ÷ 2
- ok
- address
- date
Multiparty Session Types

Global Types

Alice $\rightarrow$ Bob: $\langle \text{Nat} \rangle$.
Bob $\rightarrow$ Carol: $\langle \text{Nat} \rangle$.

Local Types

$T_{Bob} = ?\langle Alice, \text{Nat} \rangle; !\langle Carol, \text{Nat} \rangle; \text{end}$

Multiple Languages

$P_{Bob} = s?(Alice, x); s!(Carol, x); 0$

Projection
Figure 5: A coordinated set of autonomous underwater vehicles
Figure 3: Observatory comprised of ships, aircraft and autonomous vehicles linked to assimilation modeling capabilities on shore.
Multiparty Session Type Theory

- Multiparty Asynchronous Session Types [POPL’08]
- Progress
  - Inference of Progress Typing [Coordination’13]
- Asynchronous Optimisations and Resource Analysis
  - Global Principal Typing in Partially Commutative Asynchronous Sessions [ESOP’09]
  - Higher-Order Pi-Calculus [TLCA’07, TLCA’09]
  - Buffered Communication Analysis in Distributed Multiparty Sessions [CONCUR’10]
Logics

Design-by-Contract for Distributed Multiparty Interactions [CONCUR’10]

Specifying Stateful Asynchronous Properties for Distributed Programs [CONCUR’12]

Multiparty, Multi-session Logic [TGC’12]

Extensions of Multiparty Session Types

Multiparty Symmetric Sum Types [Express’10]

Parameterised Multiparty Session Types [FoSSaCs’10, LMCS]

Global Escape in Multiparty Sessions [FSTTCS’10]

Dynamic Multirole Session Types [POPL’11]

Nested Multiparty Sessions [CONCUR’12]
Dynamic Monitoring

Asynchronous Distributed Monitoring for Multiparty Session Enforcement [TGC’11]

Monitoring Networks through Multiparty Sessions [FORTE’13]

Automata Theories

Multiparty Session Automata [ESOP’12]

Synthesis in Communicating Automata [ICALP’13]

Typed Behavioural Theories

On Asynchronous Eventful Session Semantics [FORTE’11]


Governed Session Semantics [CONCUR’13]

Choreography Languages

Compositional Choreographies [CONCUR’13]
Language and Implementations

➤ Carrying out large-scale experiences with OOI, VMWare, Red Hat, Congnizant, UNIFI, TrustCare

➤ JBoss SCRIBBLE [ICDCIT’10, COB’12] and SAVARA projects

➤ High-performance computing

Session Java [ECOOP’08, ECOOP’10, Coordination’11]

⇒ Multiparty Session C [TOOLS’12][Hearts’12][EuroMPI’12]

➤ Multiparty session languages Ocaml, Java, C, Python, Scala, Jolie

➤ Trustworthy Pervasive Healthcare Services via Multiparty Session Types [FHIES’12]

➤ SPY: Local Verification of Global Protocols [RV’13]

➤ Practical interruptible conversations: Distributed dynamic verification with session types and Python [RV’13]
Session Type Projects

- **COST Action** *Behavioural Types for Reliable Large-Scale Software Systems*, over 60 academic members in 17 countries

- **SADEA** EPSRC *Exploiting Parallelism through Type Transformations for Hybrid Manycore Systems*, with Vanderbauwhede, Scholz, Gay and Luk

- **Programme Grant** *From Data Types to Session Types: A Basis for Concurrency and Distribution*, with Wadler and Gay

- EPSRC *Conversation-Based Governance for Distributed Systems by Multiparty Session Types*

- **NSF** Ocean Observatories Initiative

- **VMware** Dynamic Assurance based on Multiparty Session Types

- **Cognizant** EPSRC Knowledge Transfer Secondments
Session Type Reading List

[ESOP’98] Honda, Vasconcelos and Kubo, Language Primitives and Type Disciplines for Structured Communication-based Programming,

[SecRet’06] Yoshida and Vasconcelos, Language Primitives and Type Disciplines for Structured Communication-based Programming Revisited, ENTCS.

[ECOOP’08] Hu, Yoshida and Honda, Session-Based Distributed Programming in Java

[POPL’08] Carbone, Yoshida and Honda, Multiparty Asynchronous Session Types

[WS-FM’09] Dezani-Ciancaglini and de’Liguoro, Sessions and Session Types

[TOOLS’12] Ng, Yoshida and Honda, Multiparty Session C

[CONCUR’10] Caires and Pfenning, Session Types as Intuitionistic Linear Propositions; [ICFP’12] Walker, as Classical Linear Propositions.

[OOI] Video by John Orcutt, Professor of Geophysics, UCSD, Ocean Observing: Oceanography in the 21st Century
A rare cluster of qualities

From the team of OOI CI:

*Kohei has lead us deep into the nature of communication and processing. His esthetics, precision and enthusiasm for our mutual pursuit of formal Session (Conversation) Types and specifically for our OOI collaboration to realize this vision in very concrete terms were, as penned by Henry James, lessons in seeing the nuances of both beauty and craft, through a rare cluster of qualities - curiosity, patience and perception; all at the perfect pitch of passion and expression.*