Multiparty Session Types and their Applications to Large Distributed Systems

Nobuko Yoshida and Raymond Hu

Imperial College London

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 (GL), Scholz (HW) (1.53M)
- Programme Grant EPSRC *From Data Types to Session Types: A Basis for Concurrency and Distribution*, with Wadler (ED) and Gay (GL) (3.9M)
- EPSRC *Conversation-Based Governance for Distributed Systems by Multiparty Session Types* (1.5M)
- EU FP7 FETOpenX UpScale with de Boer (CWI), Clark, Wrigstad (Uppsala) Johnsen (Oslo) and Drossopoulou
- VMware Dynamic Assurance based on Multiparty Session Types
- Cognizant EPSRC Knowledge Transfer Secondments
- EPSRC Imperial Doctoral Prize Fellowship

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

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?

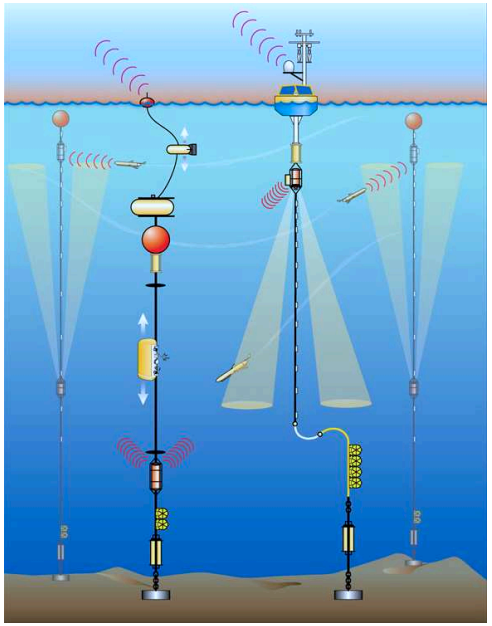
Communication is Ubiquitous

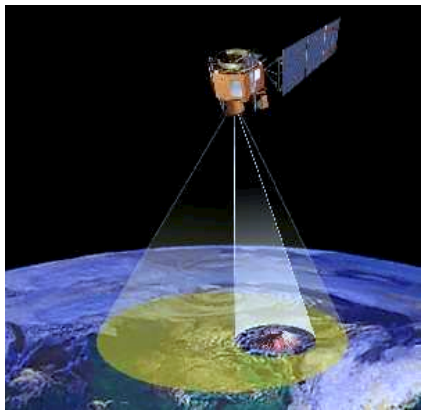
- Internet, the WWW, Cloud Computing, the next-generation manycore chips, message-passing parallel computations, large-scale cyberinfrastructure for e-Science.
- The way to organise software is increasingly based on communications.
- Applications need *structured* series of communications.

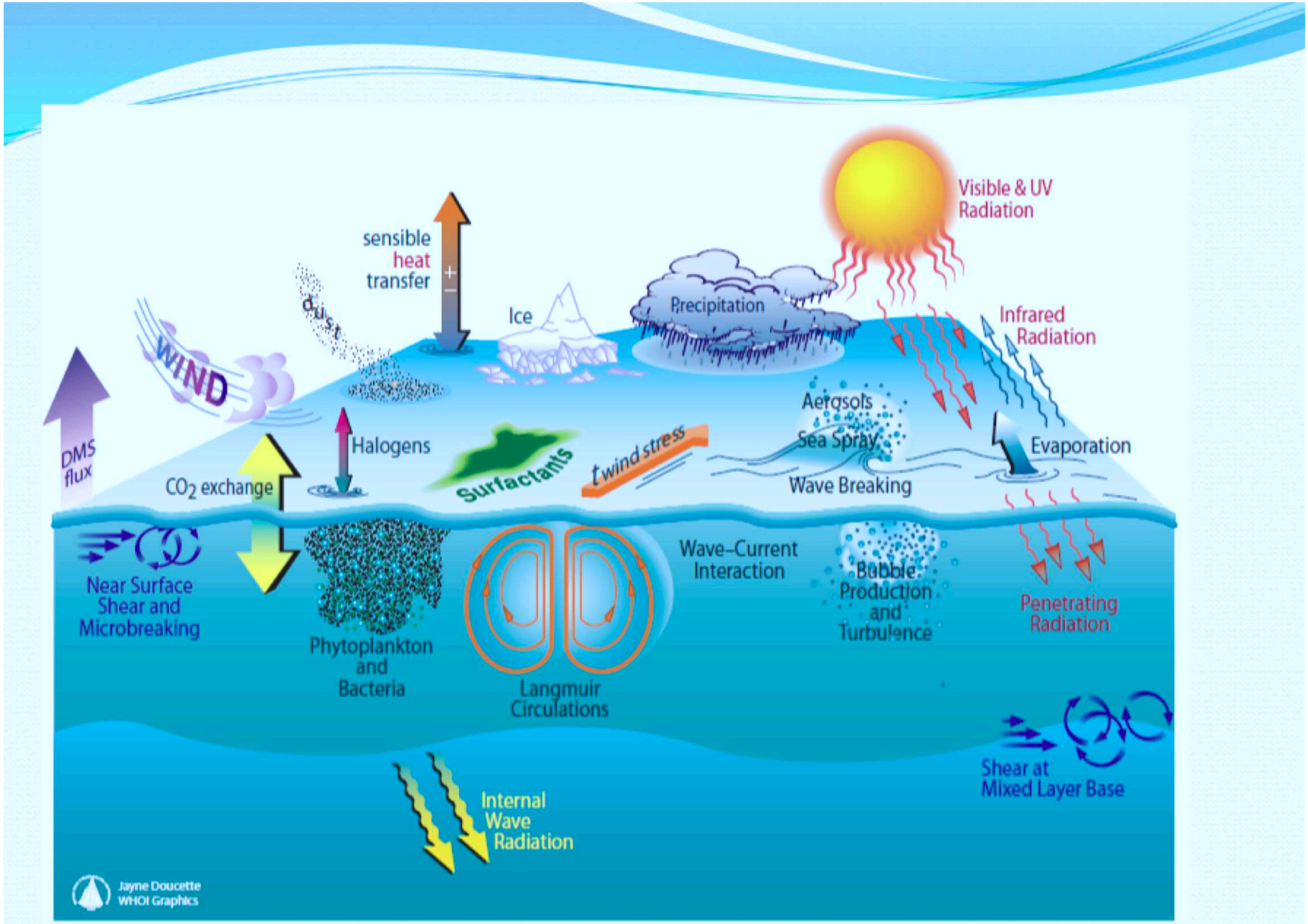
- Question $\Longrightarrow$ Multiparty session type theory
  - How to formally abstract/specify/implement/control communications?

Ocean Observatories Initiative

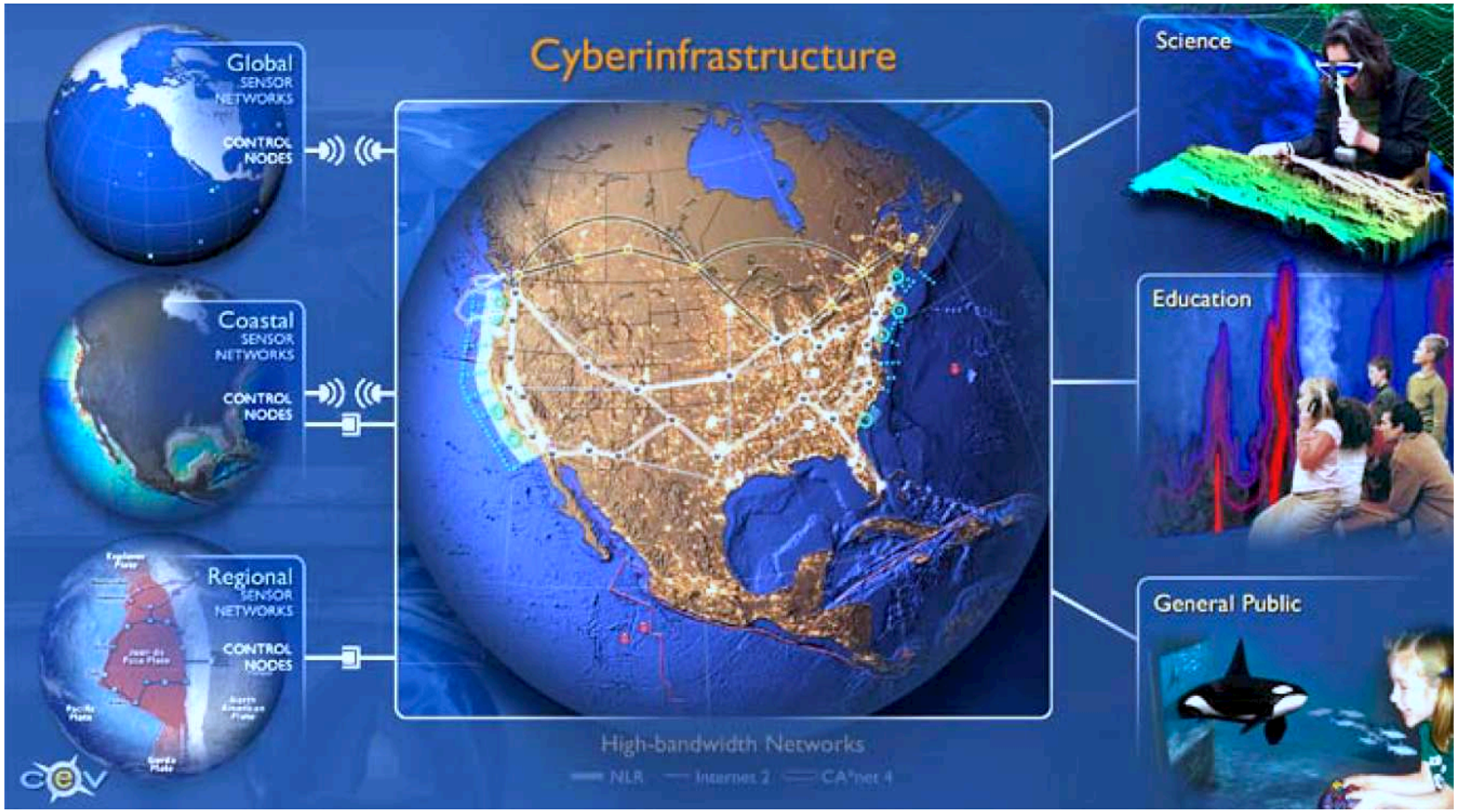
- A NSF project (400M$, 5 Years) to build a cyberinfrastructure for observing oceans around US and beyond.
- Real-time sensor data constantly coming from both off-shore and on-shore (e.g. buoys, submarines, under-water cameras, satellites), transmitted via high-speed networks.

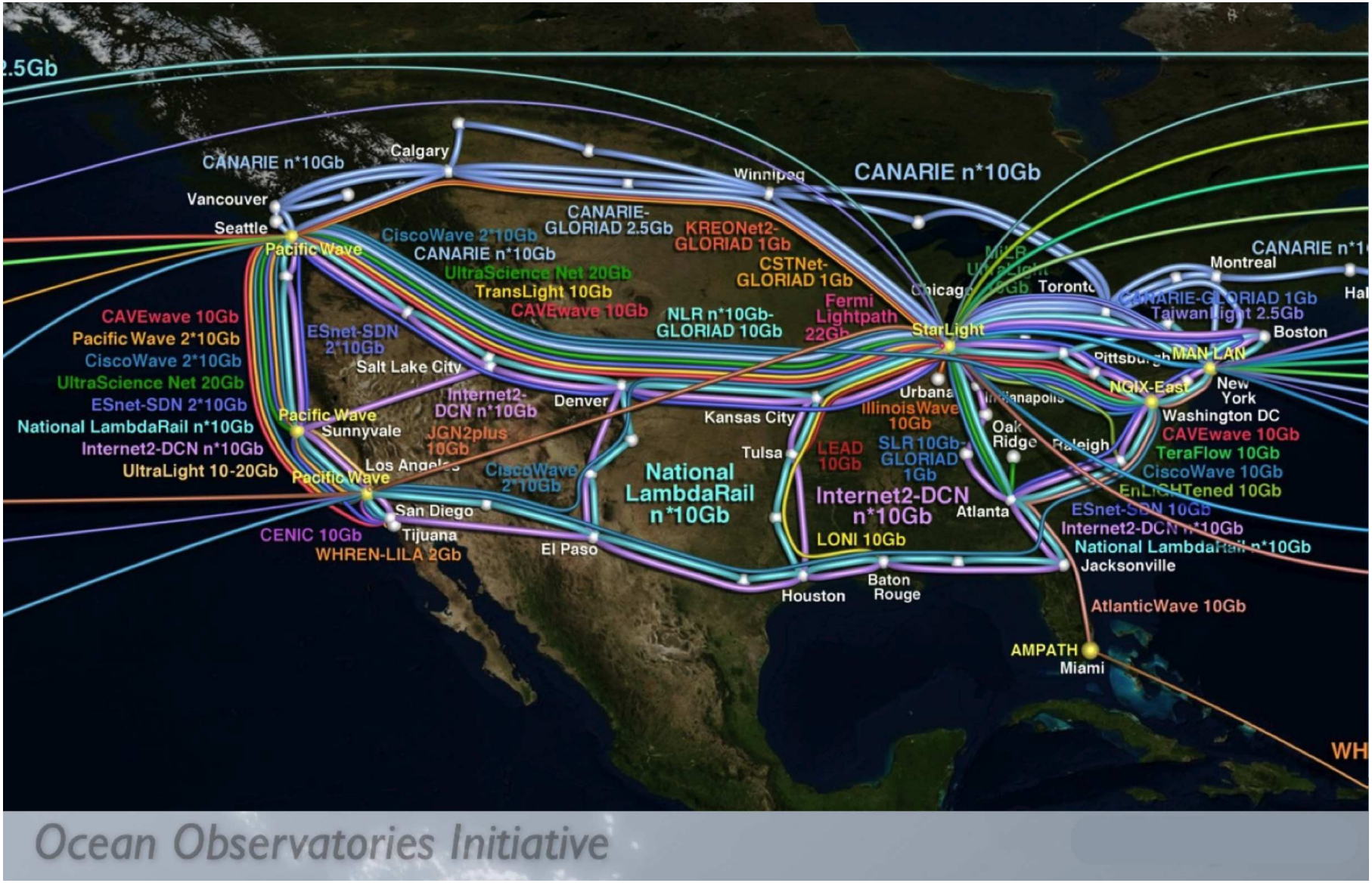





Visible & UV Radiation
sensible heat transfer
+
−
Ice
Precipitation
Infrared Radiation
dust
WIND
DMS flux
Halogens
Aerosols
Sea Spray
Evaporation
CO_2 exchange
Surfactants
τ wind stress
Wave Breaking
Near Surface Shear and Microbreaking
Phytoplankton and Bacteria
Wave–Current Interaction
Bubble Production and Turbulence
Penetrating Radiation
Langmuir Circulations
Shear at Mixed Layer Base
Internal Wave Radiation
Jayne Doucette
WHOI Graphics

Ocean Observatories Initiative





Ocean Observatories Initiative

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?

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

  $\Longrightarrow$ Can describe communication protocols as *types*

  $\Longrightarrow$ 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:

```
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]

⇓

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]

⇓

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]

⇓

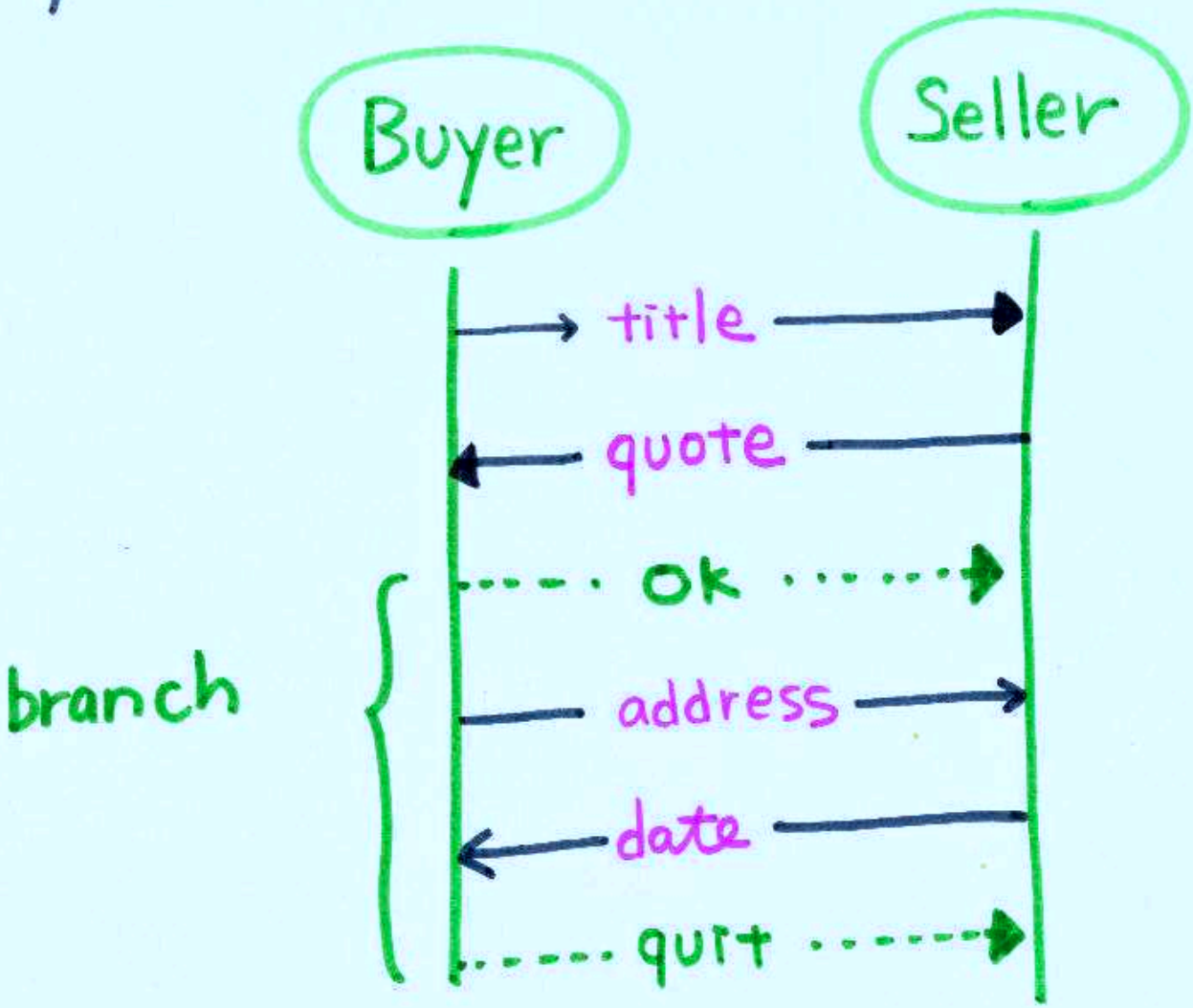
Scribble at π4 Technology

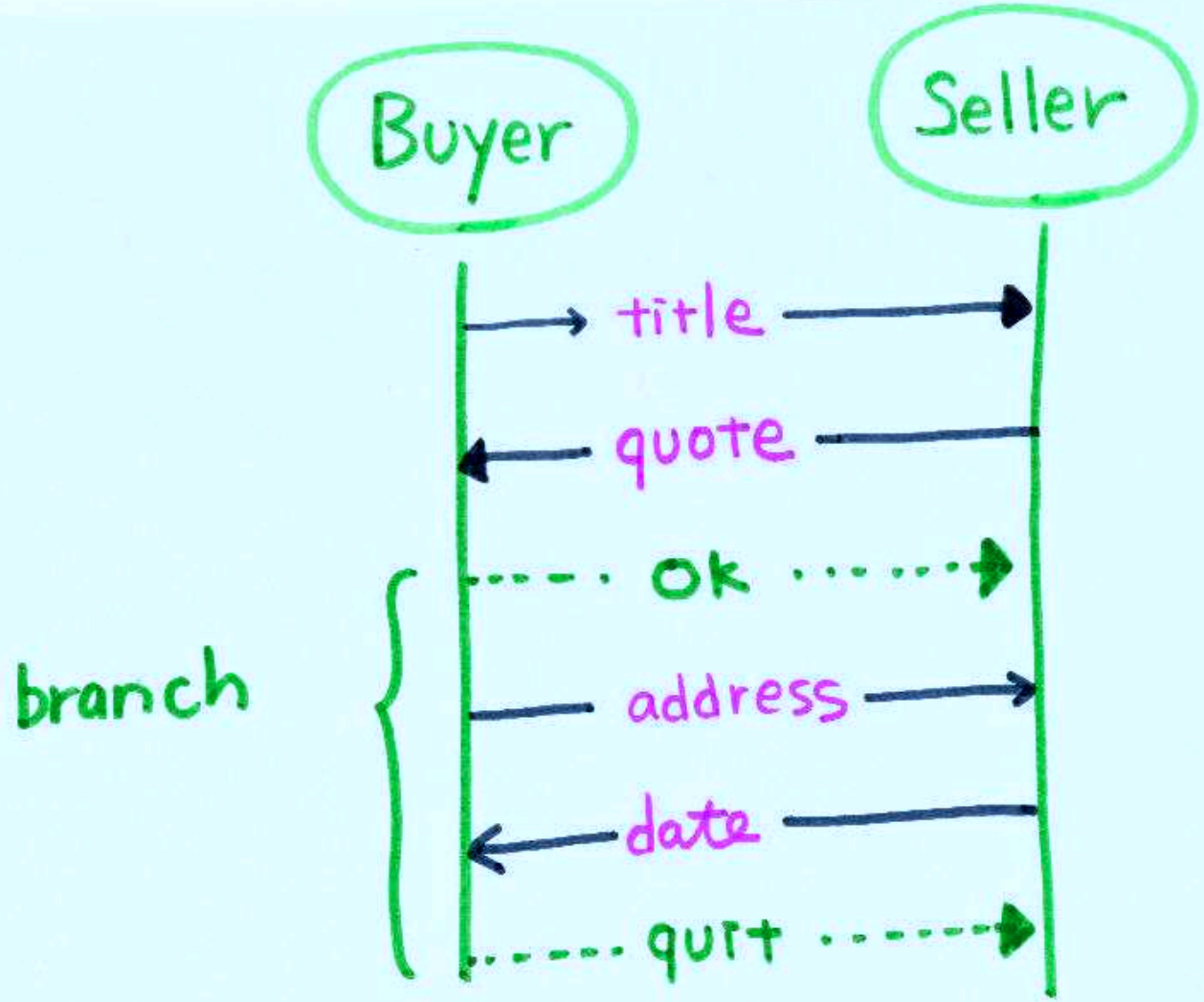
⇓

Multiparty Session Types [POPL'08]

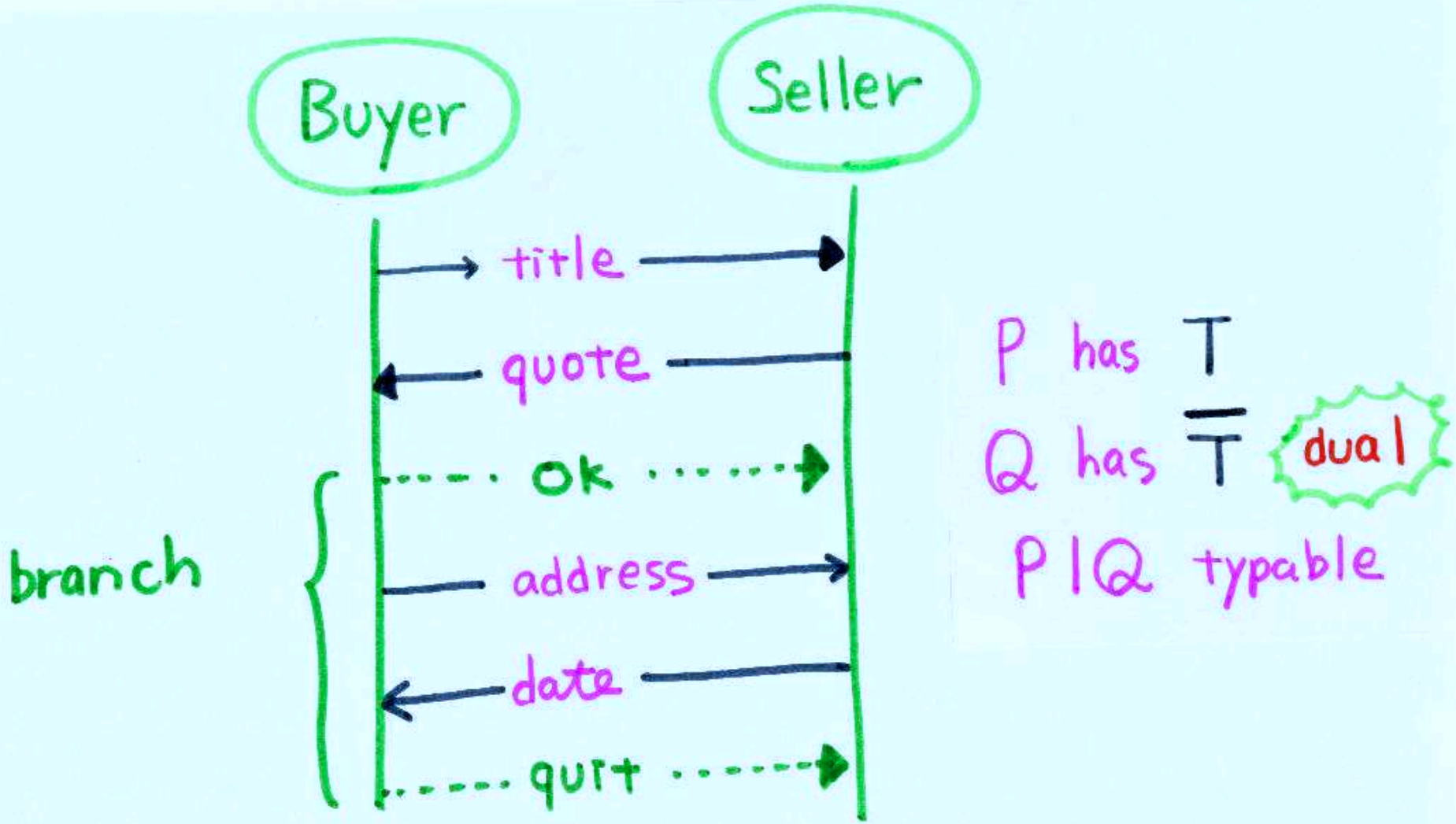
⇓

Binary Session Types : Buyer-Seller Protocol





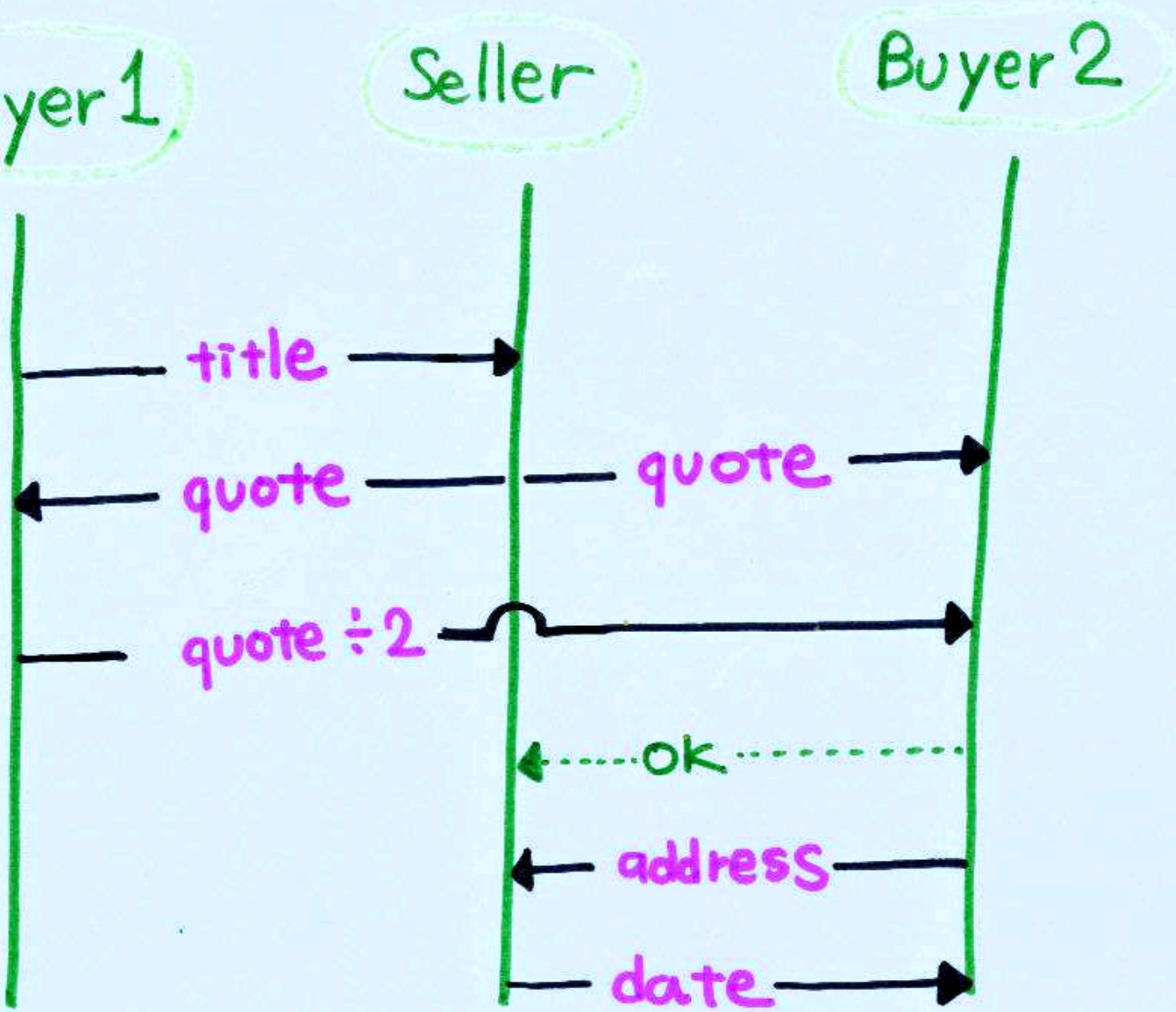
! String ; ? Int ; ⊕ { ok : ! String ; ? Date ; end , quit : end }

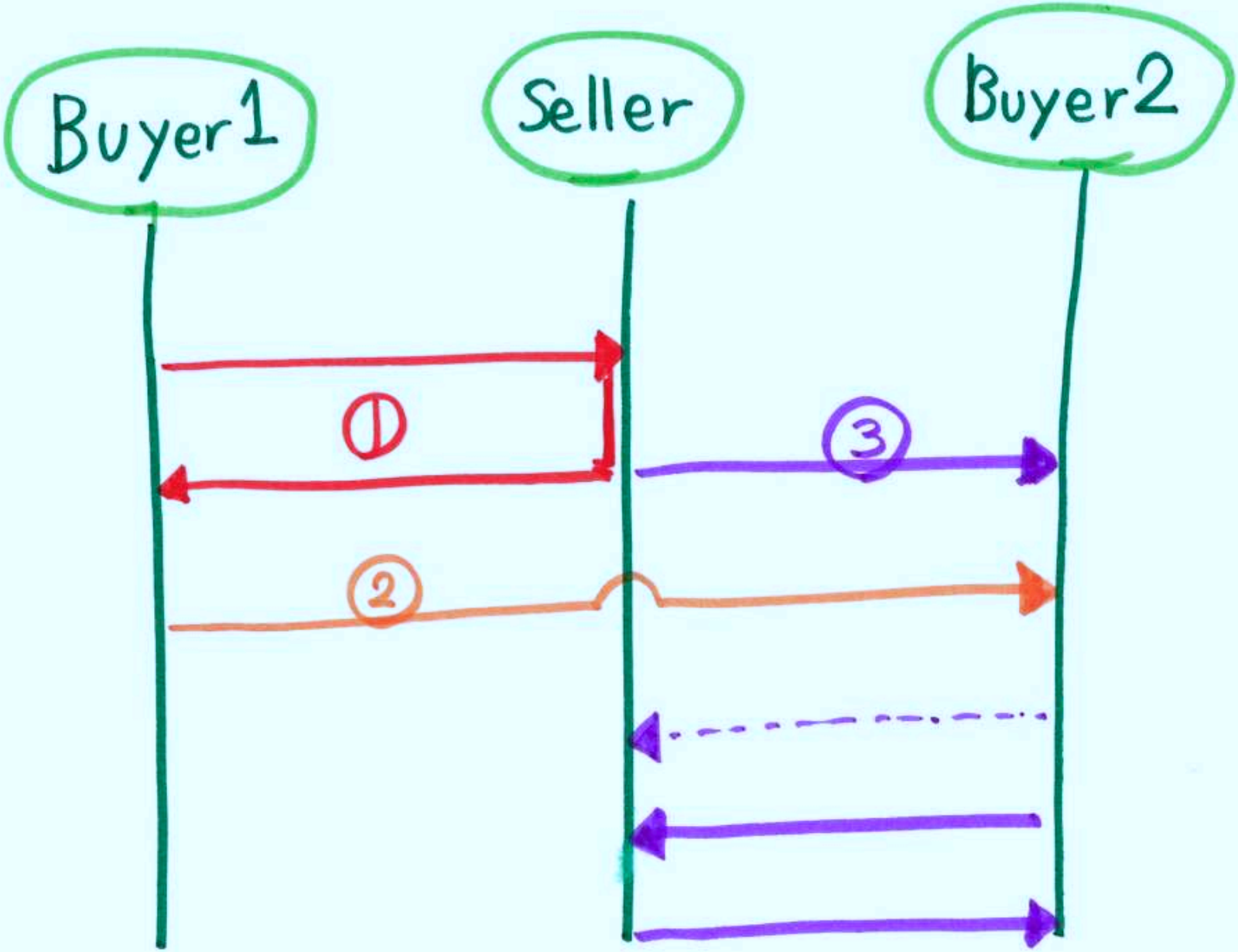


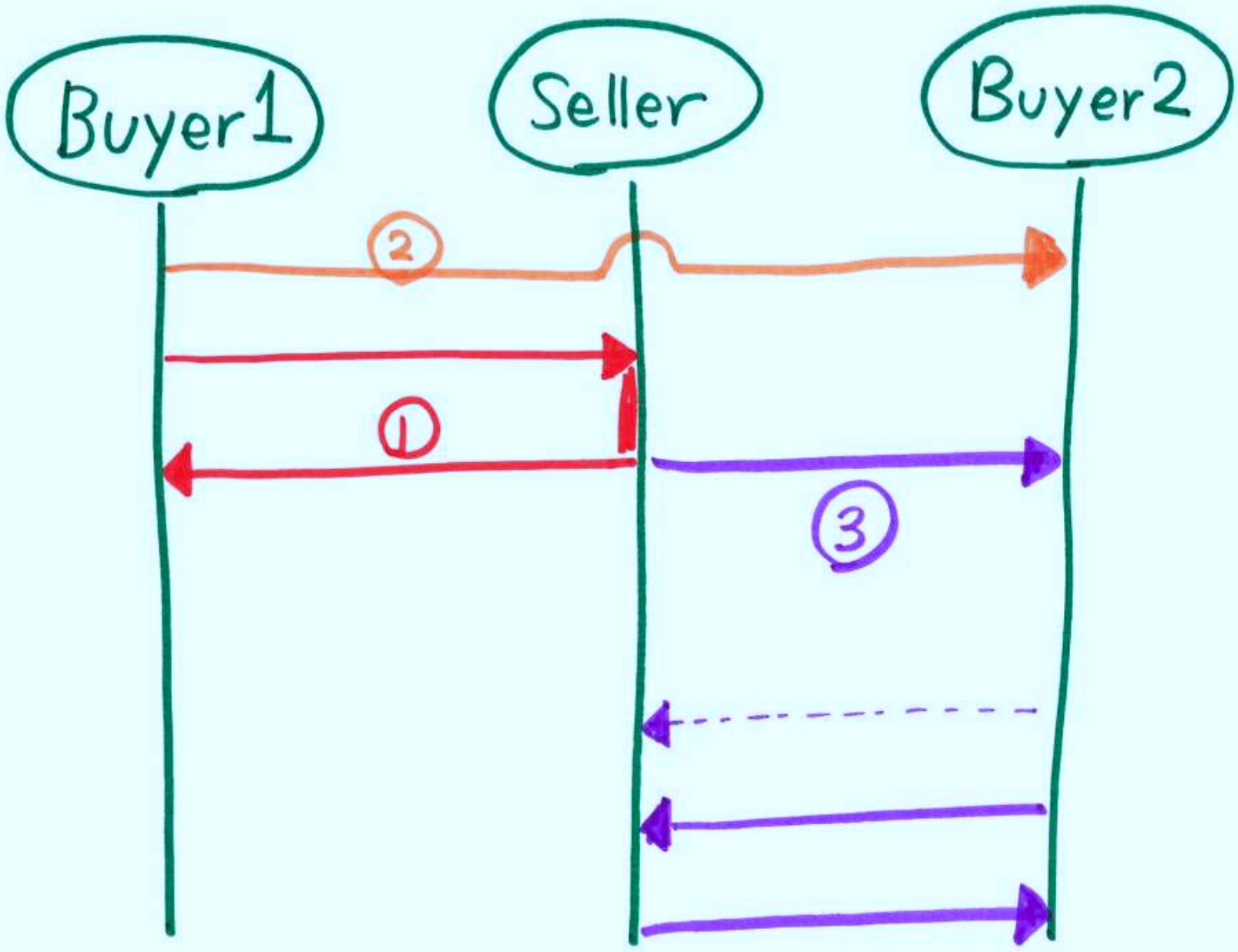
! String ; ? Int ; ⊕ { ok : ! String ; ? Date ; end , quit : end }

dual ? String ; ! Int ; & { ok : ? String ; ! Date ; end , quit : end }

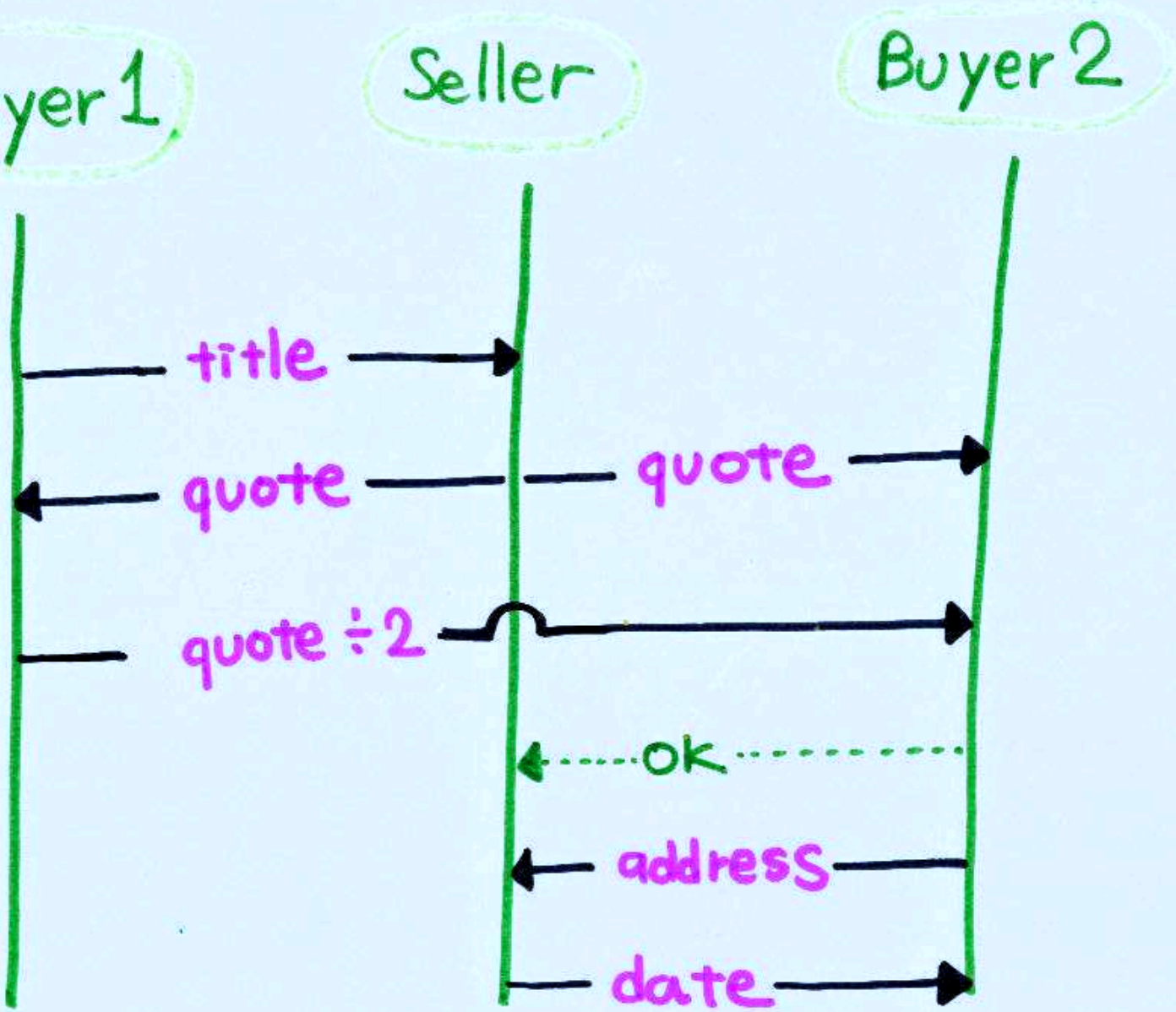
Multiparty Session Types



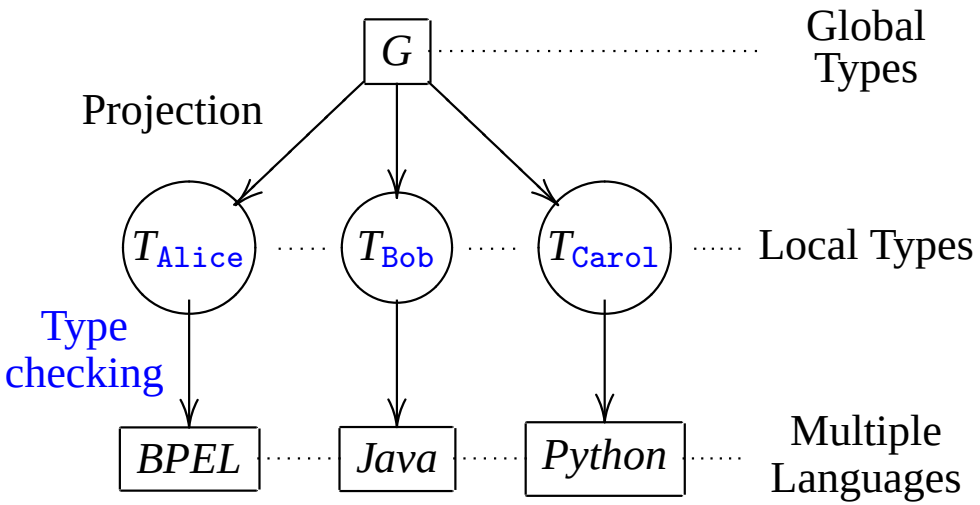

Buyer1
Seller
Buyer2
①
②
③


Buyer1
Seller
Buyer2
2
1
3

Multiparty Session Types



Multiparty Session Types

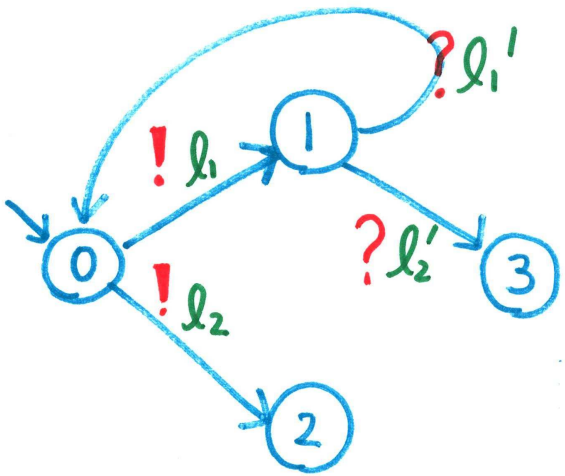


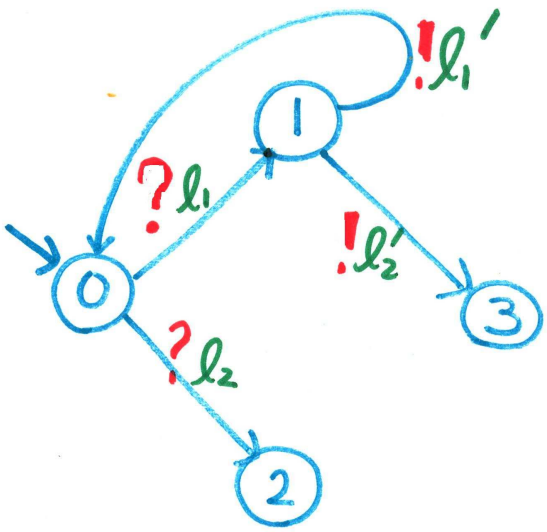
$\texttt{Alice} \to \texttt{Bob} : \langle \mathsf{Nat} \rangle .$
$\texttt{Bob} \to \texttt{Carol} : \langle \mathsf{Nat} \rangle .\mathsf{end}$

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

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

$T_1 = \mu X.\, \oplus\{ \ell_1\langle \text{Nat} \rangle.\ \&\{ \ell_1'\langle \text{Int} \rangle.\, X,\ \ell_2'\langle \text{Nat} \rangle.\, \text{End} \}$
$\ell_2\langle \text{Int} \rangle.\, \text{End} \}$



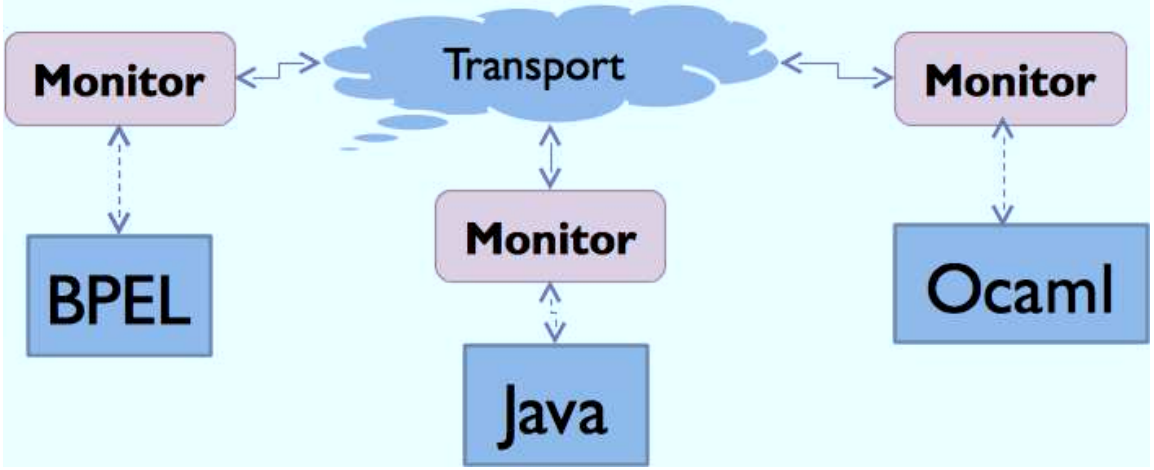


Dual

Binary session types correspond to two compatible, deterministic CFSMs with non-mixed states [Gouda et al 86]

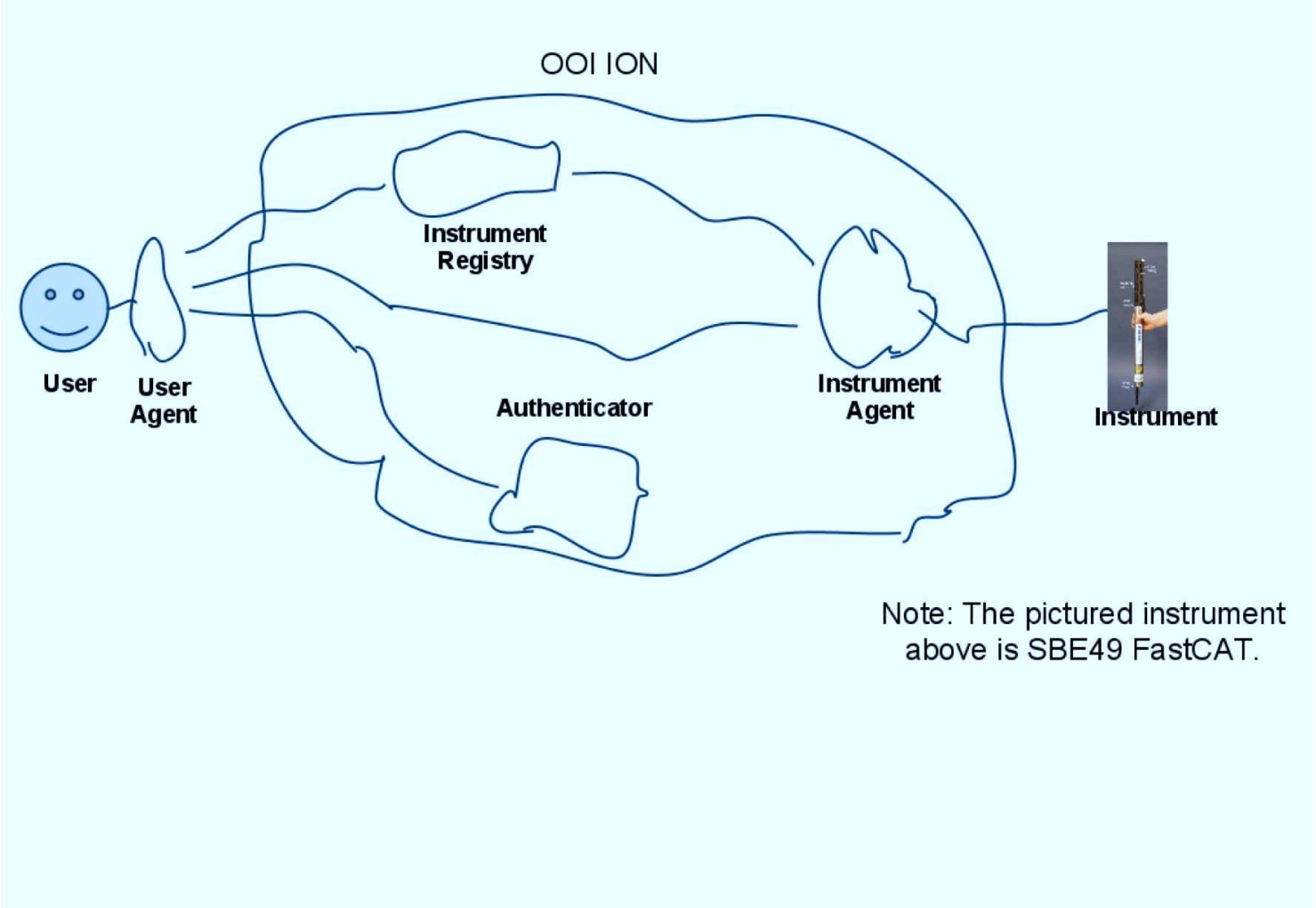
$\Longrightarrow$ Multiparty session types and CFSMs [ESOP'12,ICALP'13].

Dynamic Message Monitoring

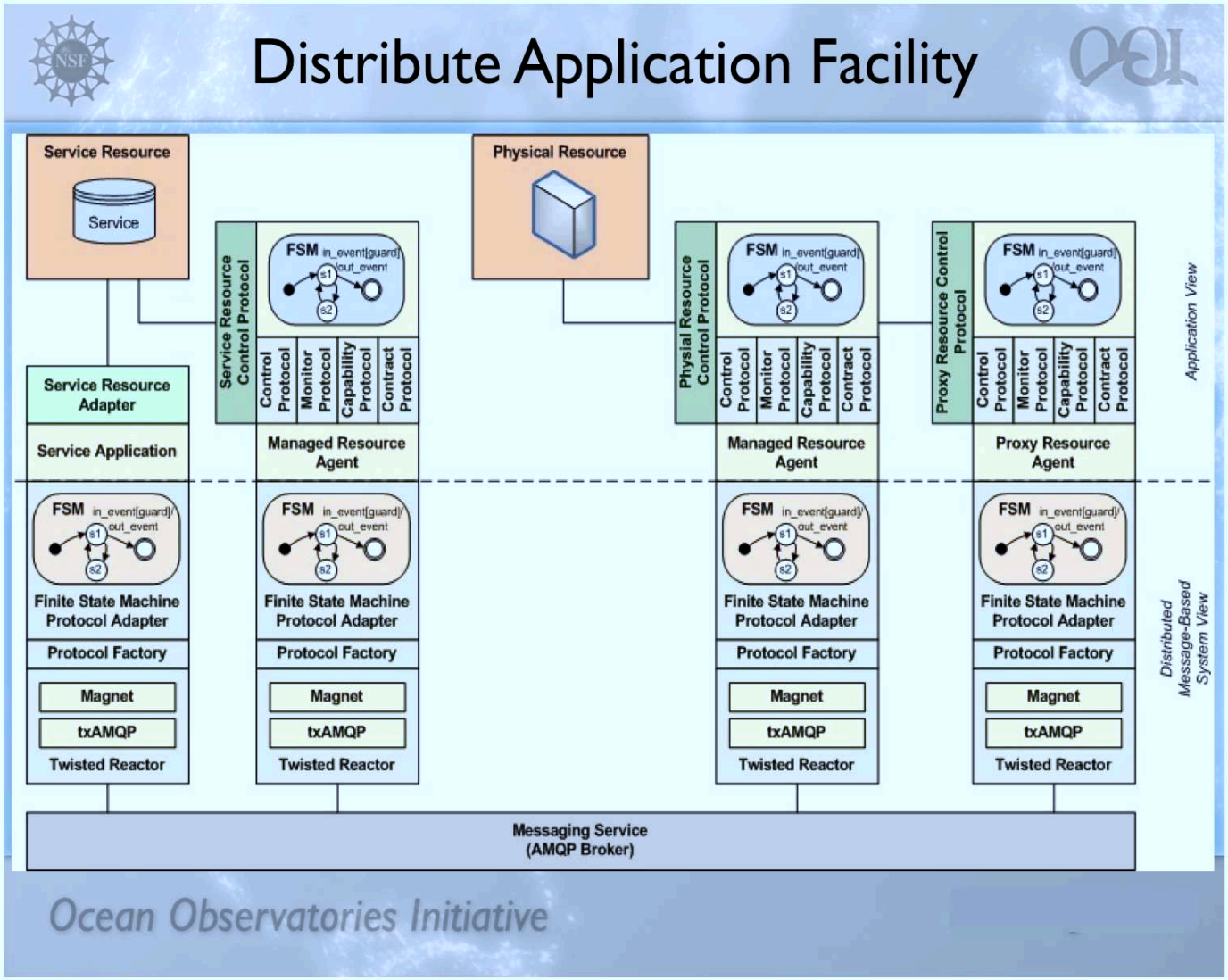


- Others' code may be unreliable, specifications can change.
- Use CFSMs generated from local types as *monitors*, checking incoming and outgoing messages in *linear time*, managing global behaviour.
- Theories of dynamic monitoring and logics [CONCUR'10,TGC'11,TGC'12,CONCUR'12,FMOODS'13].

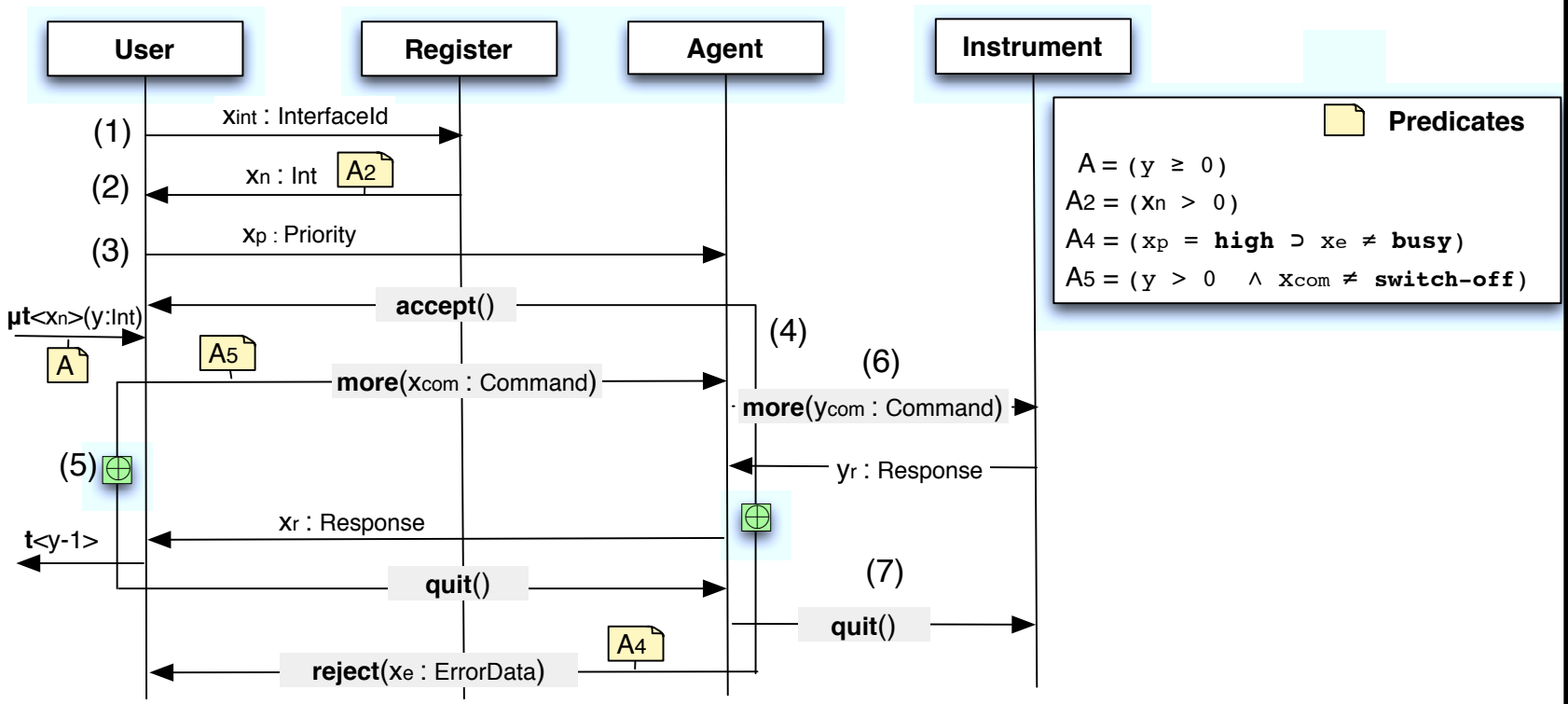
Use Case: Command Instrument



Distribute Application Facility



Command Instrument Specification



Use Case

Global View:

$$
\begin{aligned}
G_{IC} \;=\; & \texttt{User} \to \texttt{Register} : (x_{i}nt : InterfaceId) \\
& \texttt{Register} \to \texttt{User} : (x_n : Int) \\
& \texttt{User} \to \texttt{Agent} : (x_p : Priority) \\
& \texttt{Agent} \to \texttt{User} : \{\texttt{accept}().G_{acc},\ \texttt{reject}(x_E : \texttt{ErrData})\}
\end{aligned}
$$

$$
\begin{aligned}
G_{acc} \;=\; & \mu \mathrm{t}\langle x_n \rangle (y)\{y \geq 0\}. \\
& \texttt{User} \to \texttt{Agent} : \{\texttt{more}(x_{com} : \texttt{Command}).G_{com},\ \texttt{quit}()\} \\
G_{com} \;=\; & \texttt{Agent} \to \texttt{Instrument} : (y_{com} : \texttt{Command}). \\
& \texttt{Instrument} \to \texttt{Agent} : (y_r : \texttt{Response}). \\
& \texttt{Agent} \to \texttt{User} : (x_r : \texttt{Response}).\ \mathrm{t}\langle y - 1 \rangle
\end{aligned}
$$

Scribble:

```
session AccessConversation =
roles user, agent, register

global inv main = {
    InterfaceId(string) from user to register;
    MaxCommands(string) from register to user;
    Priority(string) from user to agent;
    choice from user to agent
     {Accept.
       (more_commands:
        choice from agent to user
         {More.
           (CCommand(string) from user to agent;
            choice from agent to instrument
             {Command.(
               CCommand(string) from agent to instrument;
               Response(string) from instrument to agent)
              | Quit};
             Response(string) from agent to user)
         | Quit};
        more_commands)
      | Reject.(ErrorData(string) from agent to user;)}
}
```

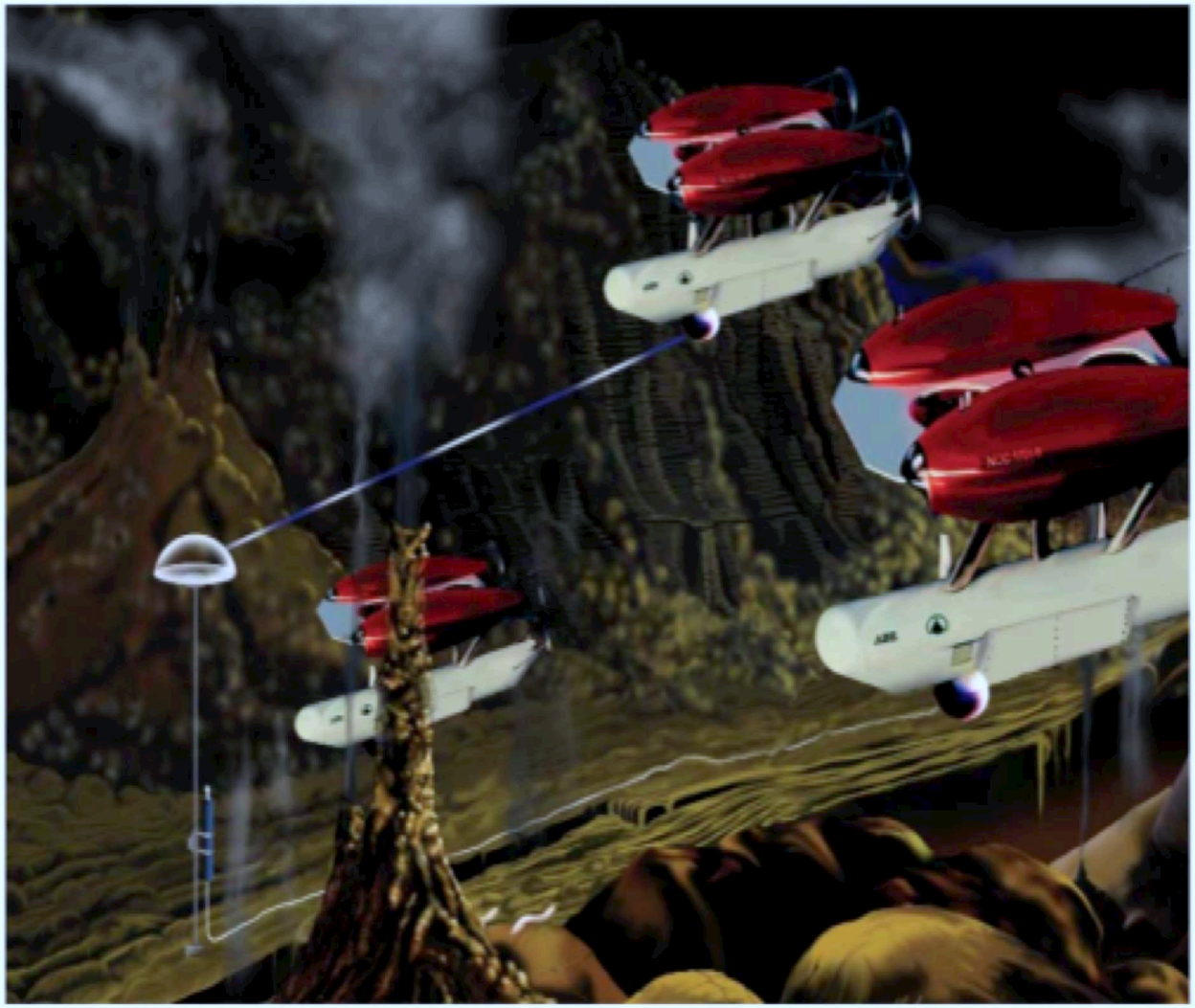


Figure 5: A coordinated set of autonomous underwater vehicles

Figure 3: Observatory comprised of ships, aircraft and autonomous vehicles linked to assimilation modeling capabilities on shore

Ocean Observatory Initiative

CREATE ACCOUNT | SIGN IN

SEARCH

RESOURCES

- All Resources
- Data Products ✓
- Observatories
- Platforms
- Instruments

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.

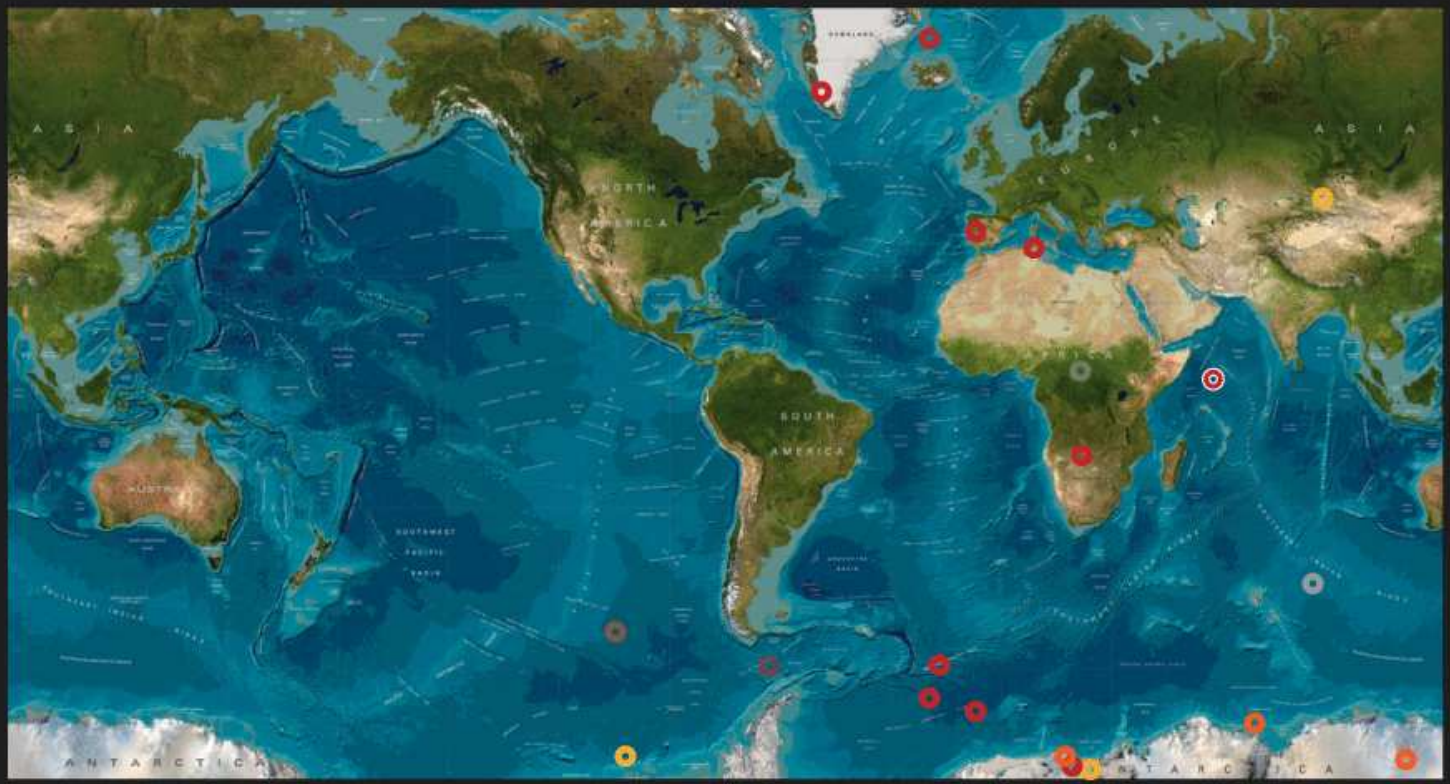
National Science Foundation working with *Consortium for Ocean Leadership*

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.

Location

CURRENT LOCATION

FILTER



DATA LEGEND

- Temperature ✓
- Salinity
- Oxygen ✓
- Density
- Currents
- Sea Surface Height (SSH)
- Chlorophyll
- Turbidity ✓
- pH
- Seismology ✓
- Other ✓

RECENCY

1 Hour
2 hours
3 hours
5 hours
8 hours
12 hours
18 hours
24 hours
48 Hours
72 Hours

RECENT UPDATES

	NAME	DATE	TYPE	EVENT	DESCRIPTION	NOTE
01 m	Oregon Coast North Salinity	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 m	California South 100m pH	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 m	California South salinity	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
03 m	Oregon North Turbidity	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
05 m	Oregon SouthTemperature	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
20 m	Oregon Coast Currents	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 h	California South Seismology	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
01 h	Oregon Coast South 1000m Ox	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
02 h	California Coast Seismology	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here
04 h	California North Seismology	2012-01-10 23:55:55	Type	Event	Description goes here	Note goes here

FACEPAGE | RELATED | COMPOSITE | STATUS

Dashboard

RECENT IMAGES

Glider
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24

Gorgonian Coral
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24

Acoustic Release
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24

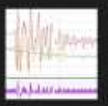
POPULAR RESOURCES

SeaBird CDT
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24

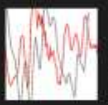
Marine caption
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24

Surface Buoy
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24

UNUSUAL EVENTS



Oregon Coast Wave Heigh
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24



Water Surface Elevation
Last Modified: 2011-06-15
Last Viewed: 2011-12-15
Last Updated: 2011-12-30, 13.24

Multiparty Session Type Theory

- Multiparty Asynchronous Session Types [POPL'08]
- Progress
  - Global Progress in Dynamically Interleaved Multiparty Sessions [CONCUR'08], [Math. Struct. Comp. Sci.]
  - 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] [Math. Struct. Comp. Sci.]
  - 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]

    [Math. Struct. Comp. Sci.]
  - 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, TGC'13] and SAVARA projects
- High-performance computing

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

  $\Longrightarrow$ C and MPI [TOOLS'12][Hearts'12][EuroMPI'12][PDP'14]
- 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 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