Relating BIP and Reo

K. Dokter[1]   S.-S.T.Q. Jongmans[1]   F. Arbab[1]   S. Bliudze[2]

[1]Centrum Wiskunde & Informatica,
Amsterdam, Netherlands

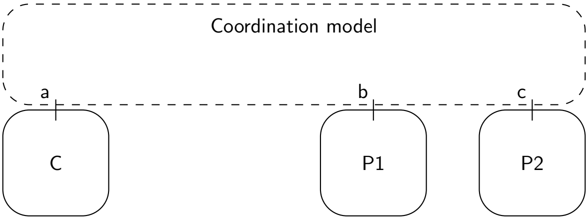
[2]École Polytechnique Fédérale de Lausanne,
Lausanne, Switzerland

ICE 2015

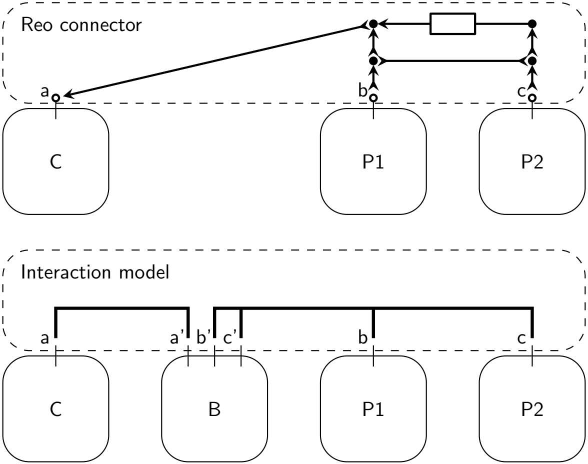
Simple producer-consumer example

```
public static void Producer1(Port b) {
    b.put("Hello ");
}

public static void Producer2(Port c) {
    c.put("ICE!");
}

public static void Consumer(Port a) {
    String s1 = a.get();
    String s2 = a.get();
    System.out.println(s1 + s2);
}
```

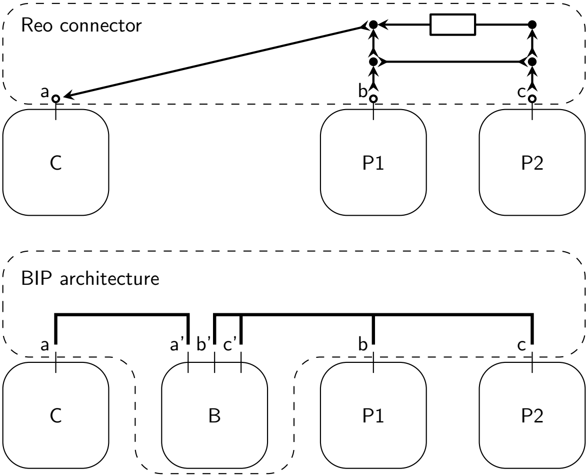
Simple producer-consumer example



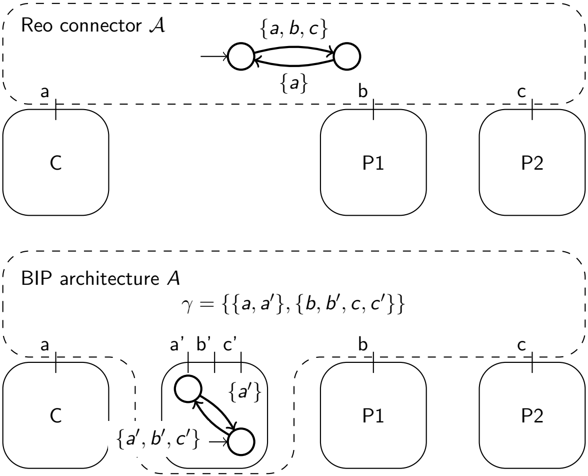
Simple producer-consumer example in Reo and BIP



Simple producer-consumer example in Reo and BIP



Simple producer-consumer example in Reo and BIP



Formal models of Reo and BIP

Definition

A port automaton is a tuple $\mathcal{A} = (Q, \mathcal{N}, \to, q_0)$, where

1. Q is a set of states;
2. $\mathcal{N}$ is a finite set of ports;
3. $\to \subseteq Q \times 2^{\mathcal{N}} \times Q$ is a transition relation;
4. $q_0 \in Q$ is the initial state.

Definition

A BIP architecture is a tuple $A = (\{B_1, \ldots, B_n\}, P, \gamma)$, where

1. $\mathcal{B} = \{B_1, \ldots, B_n\}$ is a set of coordinating components;
2. P is a set of ports;
3. $\gamma \subseteq 2^P$ is the interaction model.

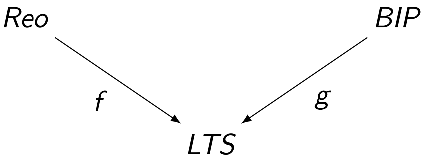
such that $P_{B_i} \cap P_{B_j} = \emptyset$, for all $1 \leq i < j \leq n$.

Interpretation of Reo and BIP

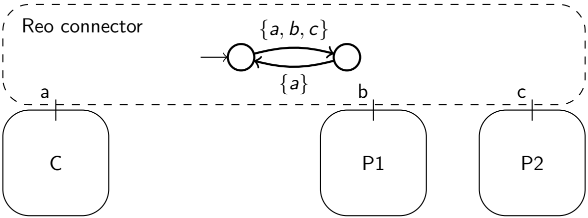
Let

- Reo be the class of all port automata; and
- BIP be the class of all BIP architectures; and
- LTS be the class of all labeled transition systems with labels in 2^P, for some finite set of ports P.

We need to find the interpretations f and g in



Interpretation of Reo connectors



The Reo connector restricts possible synchronizations of ports in its domain (independent of the components connected to them).

$$f : Reo \rightarrow LTS \qquad f(A) = A$$

Interpretation of BIP architectures

Definition (Architecture application)

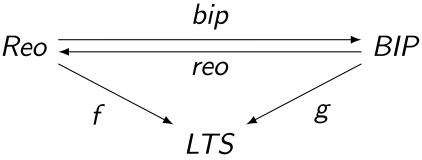
If $A = (\mathcal{B}, P, \gamma)$ is a BIP architecture and $\mathcal{C}$ a set of components, such that $S = (\mathcal{B} \cup \mathcal{C}, P \cup P_{\mathcal{C}}, \gamma)$ is a BIP architecture without dangling ports, then the application $A(\mathcal{C})$ of A to $\mathcal{C}$ is the labeled transition system over $2^{P \cup P_{\mathcal{C}}}$ that describes the behaviour of whole system S.

$$g : BIP \to LTS \qquad g(A) = \exists P_{\mathcal{C}} : A(\{D_A\})$$

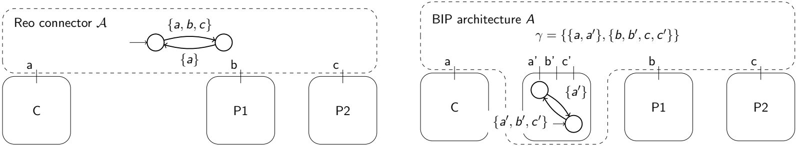
where D_A is a dummy component for A: a single state transition systems with a transition for every non-empty subset of dangling ports of A; and $\exists P_{\mathcal{C}}$ hides ports from coordinating components.

Our task

Let Reo be the class of all port automata; let BIP be the class of all BIP architectures; and let LTS be the class of all labeled transition systems with labels in 2^P, for some finite set of ports P.



Translating Reo to BIP

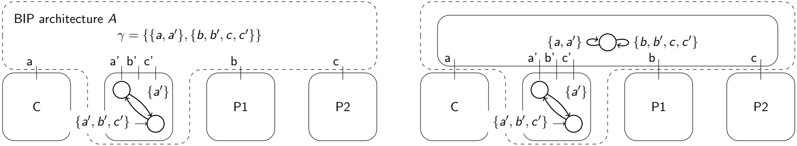


Let $\mathcal{A}$ be a port automaton over ports $\mathcal{N}$.

- Let $\overline{\mathcal{A}}$ be $\mathcal{A}$ with p replaced by p', for all ports p of $\mathcal{A}$.
- Let γ be the closure under set union of $\{\emptyset\} \cup \{\{p, p'\} \mid p \in \mathcal{N}\}$.

Then, $bip(\mathcal{A}) = (\{\overline{\mathcal{A}}\}, \mathcal{N} \cup \mathcal{N}', \gamma)$, with $\mathcal{N}' = \{p' \mid p \in \mathcal{N}\}$.

Translating BIP to Reo

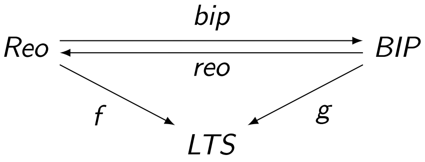


Let $A = (\mathcal{B}, P, \gamma)$ be a BIP architecture.

- Transform the interaction model into a port automaton $\mathcal{A}_\gamma$.
- Compose $\mathcal{A}_\gamma$ with the coordinating components $B_i \in \mathcal{B}$, using composition ($\bowtie$) of port automata.
- Hide all ports of the coordinating components $B_i \in \mathcal{B}$.

Let $reo(A)$ be the resulting port automaton.

Main result



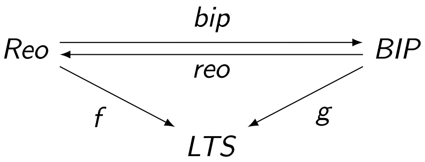
Theorem

We have

$$g(bip(\mathcal{A})) \cong f(\mathcal{A}) \quad \text{and} \quad f(reo(A)) \cong g(A),$$

for all port automata $\mathcal{A}$ *with* $q \xrightarrow{\emptyset} q'$ *iff* $q' = q$ *and all BIP architectures* A *with* $q_i \xrightarrow{\emptyset}_i q_i'$ *implies* $q_i' = q_i$*, for all coordinating components* B_i *of* A*.*

Application to model checking



Corollary

bip and reo preserve all properties closed under bisimulation, i.e.,

$$f(\mathcal{A}) \in P \Leftrightarrow g(bip(\mathcal{A})) \in P \quad \text{and} \quad g(A) \in P \Leftrightarrow f(reo(A)) \in P,$$

for all $P \subseteq LTS$ *closed under bisimulation, and all port automata* $\mathcal{A}$ *with* $q \xrightarrow{\emptyset} q'$ *iff* $q' = q$ *and all BIP architectures* A *with* $q_i \xrightarrow{\emptyset}_i q_i'$ *implies* $q_i' = q_i$*, for all coordinating components* B_i *of* A*.*

Data sensitive models

- Reo introduces data by adding guards g to the label of a transition $q \xrightarrow{N} q'$. These guards relate the data observed at the ports in N.
- BIP introduces data by extending each interaction $\{a_1, \ldots, a_n\} \in \gamma$ with
  - a guard expression, that checks whether the data presented at the a_i is eligible for synchronization;
  - an upward data transfer function, that calculates a vector x from the data obtained from the ports a_i;
  - a downward data transfer function, that outputs data at the a_i, based upon the value of x.

  Labels of coordinating components (which are just interactions) are also enriched with data.

Compositionality

Theorem

We have

$$g(bip(\mathcal{A}_1 \bowtie \mathcal{A}_2)) \cong g(reo(A_1) \bowtie reo(A_2))$$

and

$$f(reo(A_1 \oplus A_2)) \cong f(reo(A_1) \bowtie reo(A_2))$$

for all BIP architectures $A_i = (\mathcal{C}_i, P_i, \gamma_i)$, *with*

1. $q \xrightarrow{\emptyset} q' \Rightarrow q' = q$, *for all coordinating components* $B \in \mathcal{C}_i$;
2. $P_{\mathcal{C}_1} \cap P_2 = P_{\mathcal{C}_2} \cap P_1 = \emptyset$; *and*
3. $\emptyset \in \gamma_1 \cap \gamma_2$.

and for all port automata $\mathcal{A}_i = (Q_i, \mathcal{N}_i, \rightarrow_i, q_{0,i})$, *with* $q_i \xrightarrow{\emptyset}_i q_i' \Leftrightarrow q_i' = q_i$, *for all* $i = 1, 2$;

What can we learn?

Commonalities:

- exogeneous coordination;
- multi-party synchronization.

Differences:

- stateless vs. stateful coordination;
- composition of port automata versus architecture application and composition of BIP architectures;
- independent progress of silent transitions.