VerChor

A Framework for Verifying Choreographies

P. Poizat

Université Paris Ouest - LIP6

joint work with M. Güdemann, G. Salaün, and A. Dumont

FASE 2013 - Rome - March, 20th, 2013

Context

➜ composition of services / business processes

- **choreography: global perspective**
  specifies interactions among **roles**

- **peers: local perspective**
  implement roles (1-1)
  set of peers = distributed system

- **communication model**
  synchronous or asynchronous (buffers)

Context

➜ composition of services / business processes

top-down

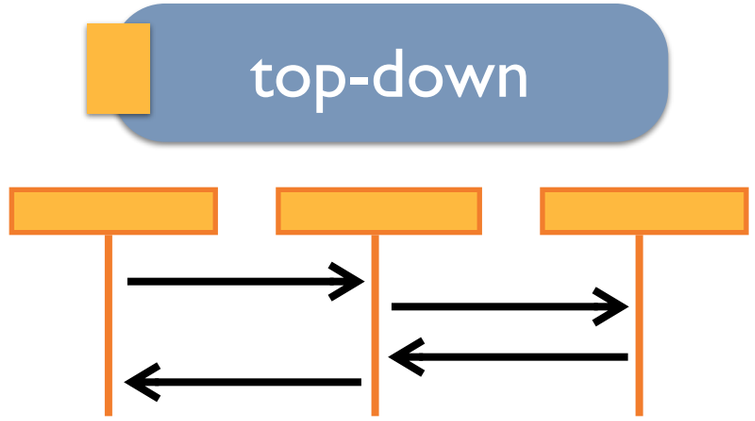
- **choreography: global perspective**
  specifies interactions among **roles**

- **peers: local perspective**
  implement roles (1-1)
  set of peers = distributed system

- **communication model**
  synchronous or asynchronous (buffers)

Context

→ composition of services / business processes



- **choreography: global perspective**
  specifies interactions among **roles**

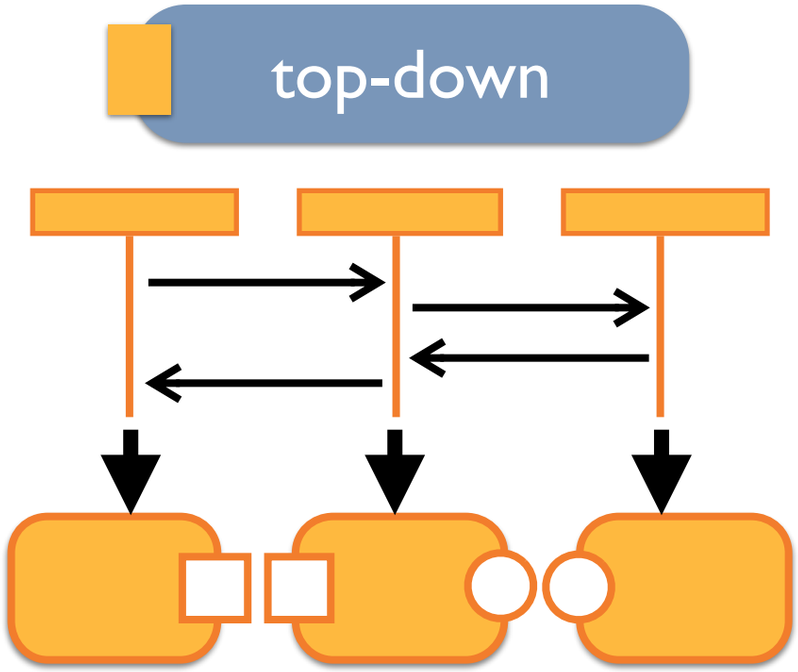
- **peers: local perspective**
  implement roles (1-1)
  set of peers = distributed system

- **communication model**
  synchronous or asynchronous (buffers)

Context

➜ composition of services / business processes

- **choreography: global perspective**
  specifies interactions among **roles**



- **peers: local perspective**
  implement roles (1-1)
  set of peers = distributed system

- **communication model**
  synchronous or asynchronous (buffers)

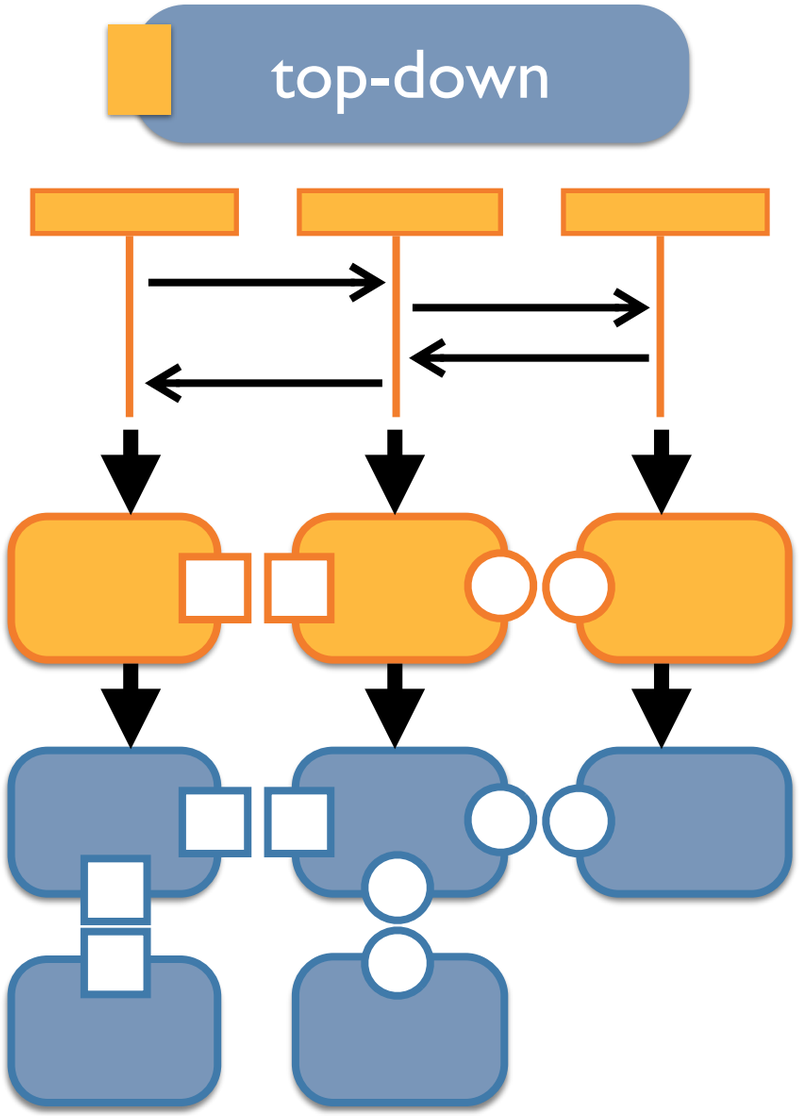
Context

➜ composition of services / business processes

- **choreography: global perspective**
  specifies interactions among **roles**

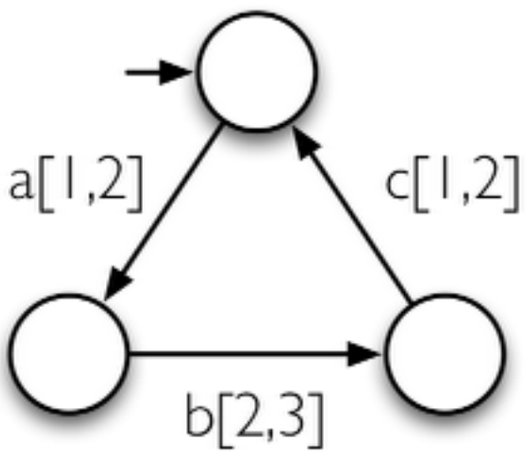
- **peers: local perspective**
  implement roles (1-1)
  set of peers = distributed system

- **communication model**
  synchronous or asynchronous (buffers)



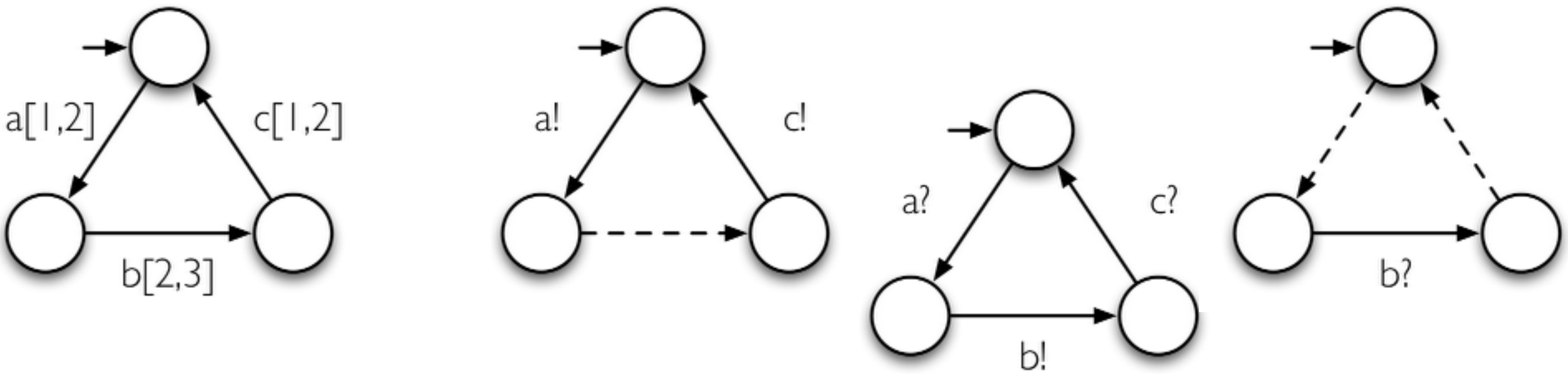
Issue

- do (projected) peers implement the choreography?



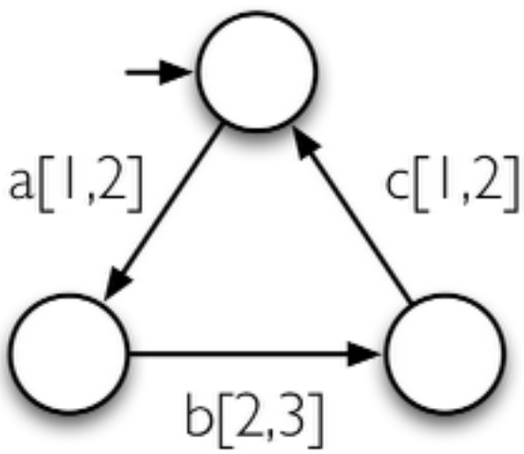
Issue

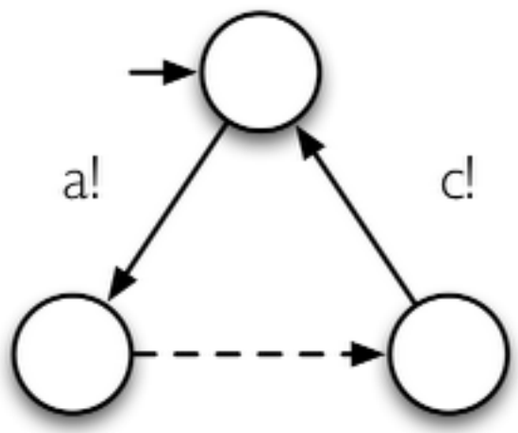
- do (projected) peers implement the choreography?

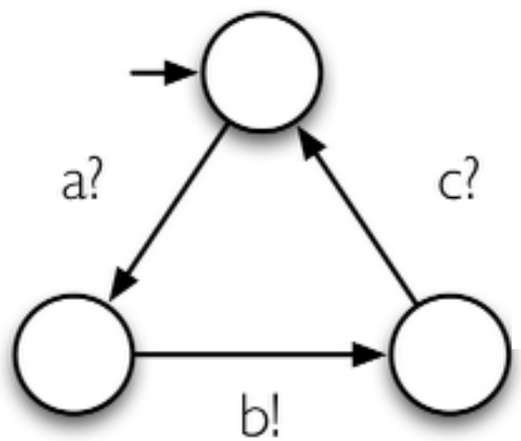


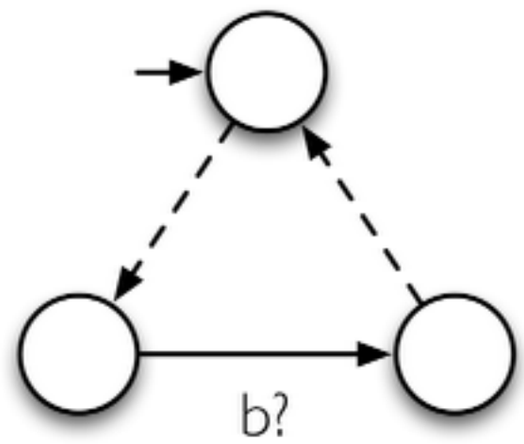
Issue

- do (projected) peers implement the choreography?

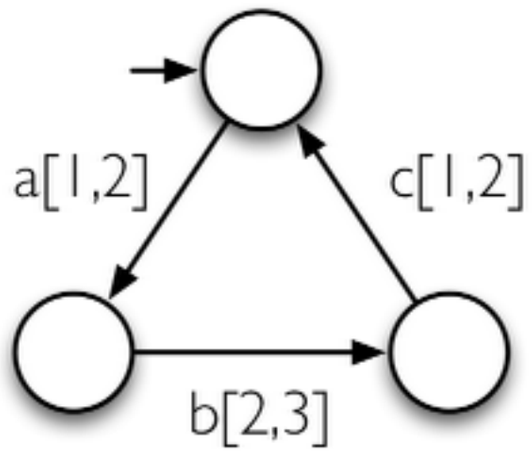






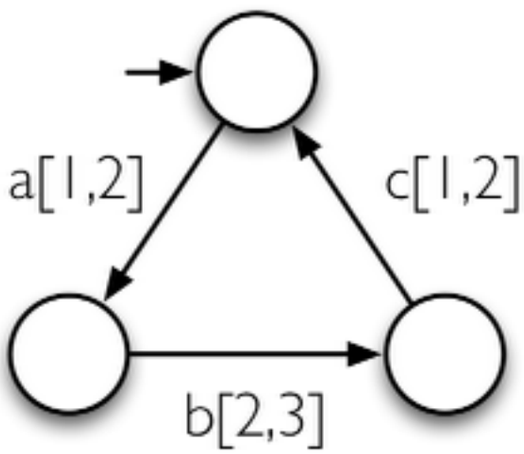


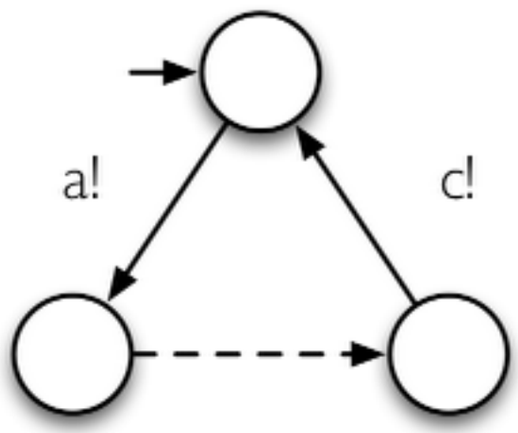
- if **synchronous** communication: **yes**

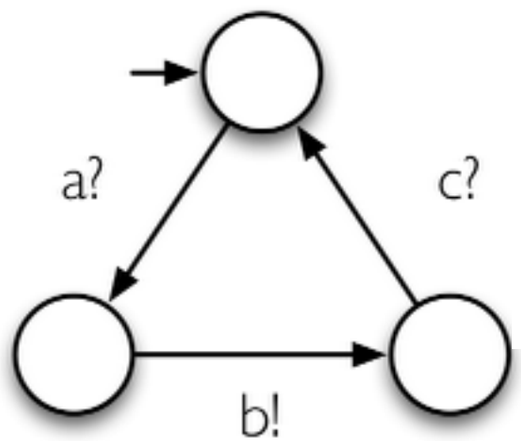


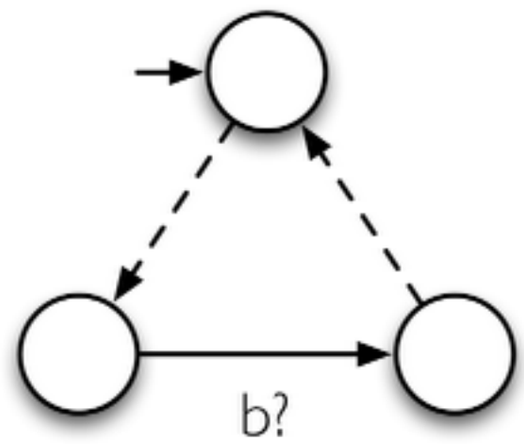
Issue

- do (projected) peers implement the choreography?



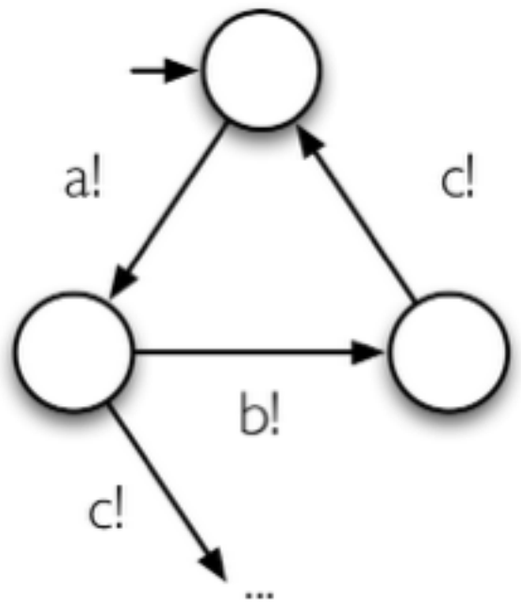






- if **synchronous** communication: **yes**

- if **asynchronous** communication: **no** further, **unbounded** system



Synchronizability and Realizability

- let C be a choreography
  S_C be the system made up of n peers P1, ..., Pn obtained from C
  synch(S_C): S_C with **synchronous** communication
  asynch(S_C,n): S_C with **n-bounded asynchronous** communication

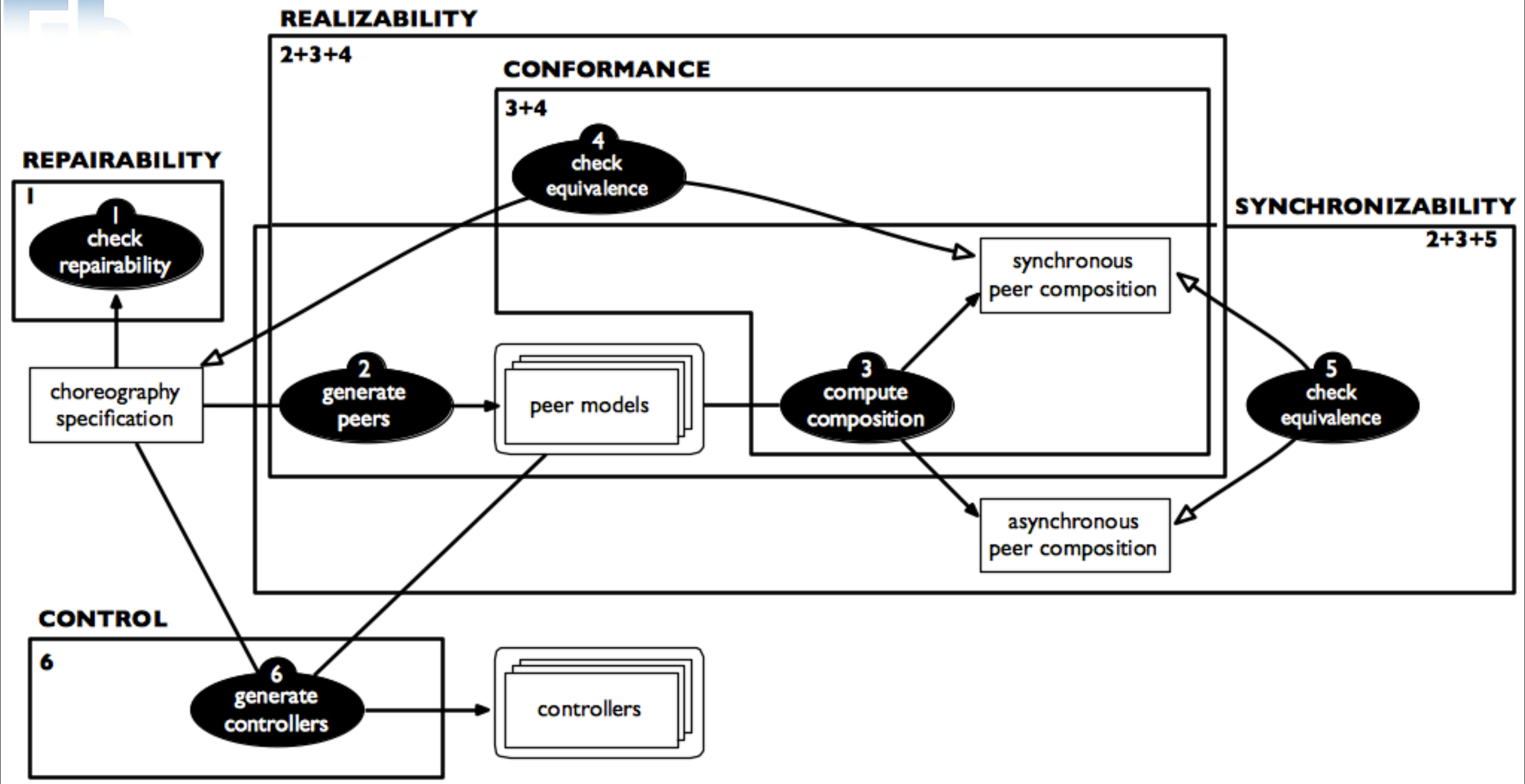
following (Basu et.al., POPL 2012) :

- **synchronizable(C): synch(S_C) behaves as asynch(S_C,1)**

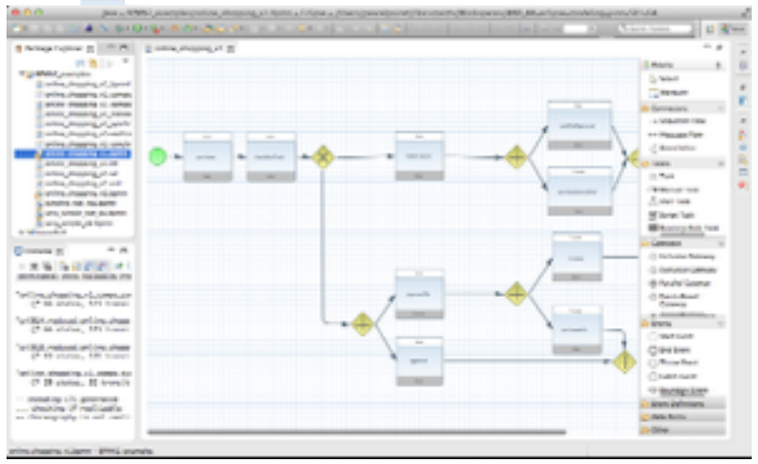
  synchronizability involves equivalence checking finite systems only
  but important result for infinite systems
  if synchronizable(C) then asynch(S_C,n+1) behaves as asynch(S_C,n)

- **realizable(C): synchronizable(C) and synch(S_C) behaves as C**

Properties: Overview



The VerChor Platform



choreography design

BPMN
WS-CDL
Chor
...

model transformation

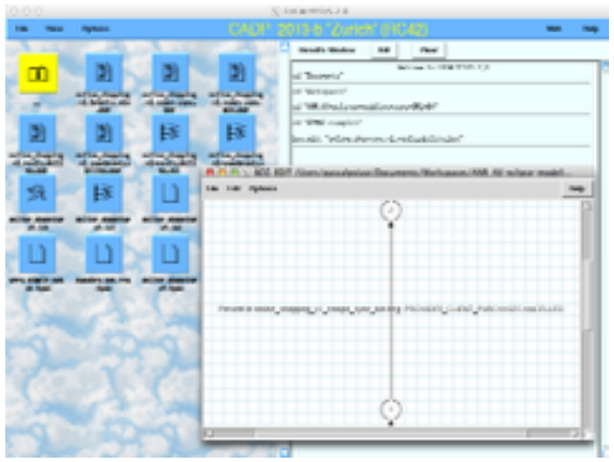
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<choreography xmlns="http://convecs.inria.fr">
    <choreoID>online_shopping_v1</choreoID>
    <participants>
        <peer>
            <peerID>Client</peerID>
        </peer>
        <peer>
            <peerID>Seller</peerID>
        </peer>
        <peer>
            <peerID>Bank</peerID>
        </peer>
        <peer>
            <peerID>Provider</peerID>
        </peer>
    </participants>
    <alphabet>
        <message>
            <msgID>ChoreographyTask_10</msgID>
            <sender>Seller</sender>
            <receiver>Client</receiver>
            <messageContent>totalValue</messageContent>
        </message>
```

intermediary format (CIF)

python scripts

Texte

diagnostic



CADP toolbox

script execution

```
module online_shopping_v1 with "get" is

% CAESAR_OPEN_OPTIONS="-silent -warning"
% CAESAR_OPTIONS="-more cat"                                    ed,Pr

% DEFAULT_PROCESS_FILE=online_shopping_v1.lnt

"online_shopping_v1_bpmnlts_min.bcg" = safety reduction of tau*.

"online_shopping_v1_compo_sync.bcg" = root leaf branching reduct
par Seller_Client_totalValue,Provider_Seller_purchaseInfo,Seller_
 peer_Seller [Seller_Client_totalValue,Provider_Client_purchaseC
||
par Provider_Client_purchaseCancelled,Provider_Client_invoice in
 peer_Client [Seller_Client_totalValue,Provider_Client_purchaseC
||
par Bank_Provider_paymentOk in
 peer_Provider [Seller_Client_totalValue,Provider_Client_purchas
||
```

formal models (LNT)
verification scripts (SVL)