VerChor
A Framework for Verifying Choreographies

P. Poizat
Université Paris Ouest - LIP6

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

• **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)
• **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)
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)
• do (projected) peers implement the choreography?
• do (projected) peers implement the choreography?
• do (projected) peers implement the choreography?

• if synchronous communication: yes
• 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 \( P_1, \ldots, P_n \) obtained from \( C \)
  \( \text{synch}(S_C) : S_C \) with \textit{synchronous} communication
  \( \text{asynch}(S_C,n) : S_C \) with \textit{n-bounded asynchronous} communication

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

• \textit{synchronizable}(C): \( \text{synch}(S_C) \) behaves as \( \text{asynch}(S_C,1) \)

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

• \textit{realizable}(C): \textit{synchronizable}(C) and \( \text{synch}(S_C) \) behaves as \( C \)
Properties: Overview

- **Realizability**: 2+3+4
  - **Repairability**
    - 1. Check repairability
  - **Conformance**
    - 4. Check equivalence
    - 2. Generate peers
    - 3. Compute composition
  - **Synchronizability**
    - 5. Check equivalence
    - Synchronous peer composition
    - Asynchronous peer composition

- **Control**: 6. Generate controllers
  - Controllers
The VerChor Platform

choreography design

BPMN
WS-CDL
Chor...

model transformation

intermediary format (CIF)

Texte

python scripts

script execution

formal models (LNT)
verification scripts (SVL)

CADP toolbox

diagnostic