From Communicating Machines to Graphical Choreographies

Julien Lange

joint work with Emilio Tuosto and Nobuko Yoshida

July 2015
alignment

separation

cohesion

Single bird ≈ local behaviour ≈ CFSM
Flock $\simeq$ global behaviour $\simeq$ choreography
Introduction

- Parts of distributed systems change/evolve, not always in a coordinated way,
- these changes are often *not* documented.
- Service oriented systems are sometimes composed dynamically,
- it is often unclear how complex the overall system has become.
- Cognizant’s Zero Deviation Lifecycle Business Unit.

A *global* point of view of a distributed system is *essential* for top-level management.
- Choreography-driven development, cf. Multiparty Session Types top-down approach (POPL’08 & ESOP’12)
- Not applicable without *a priori knowledge* of a choreography
Choreography-driven development, cf. Multiparty Session Types top-down approach (POPL’08 & ESOP’12)
- Not applicable without \textit{a priori knowledge} of a choreography
- Our goal: from \textit{Communicating Finite-State Machines to Global Graphs}
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Background: CFSMs

“On Communicating Finite-State Machines”, Brand & Zafiropulo (’83)
Global Graphs

Four Player Game

A → C: cwin
C → B: close
B → A: sig
C → A: msg
A → D: free
A → B: bwin
B → C: close
C → D: busy
Global Graphs

Alice

Bob

Carol

Four Player Game

J. Lange, E. Tuosto, N. Yoshida
From Communicating Machines to Graphical Choreographies
CFSMs

Alice

Bob

Carol

Dave

A \to C : cwin
A \to B : bwin
A \to C : close
B \to A : sig
B \to C : close
C \to A : msg
A \to D : free
C \to D : busy
CFSMs

Alice

Bob

Carol

Dave

This work

Deniélov & Yoshida – ESOP’12

J. Lange, E. Tuosto, N. Yoshida
From Communicating Machines to Graphical Choreographies
Objectives

Two main objectives:

- **Sound Condition for Safety:** generalised multiparty compatibility
  
  If \( S = (M_1, \ldots, M_k) \) is *compatible* then \( S \) is “safe”, i.e., every sent message is eventually received and no deadlock.

- **Construction of a Global Graph:**
  
  If \( G \) is the global graph constructed from \( S \), then

\[
S = (M_1, \ldots, M_k) \equiv (G_{1}, \ldots, G_{k})
\]
The Plan

1. Build $\text{TS}(S)$, the transition system of all *synchronous* executions
2. Check for safety on $\text{TS}(S)$ to
   - ensure equivalence between original system and the projections of the choreography,
   - guarantee safety (no deadlock, no orphan message)
3. Build a choreography (global graph) from $\text{TS}(S)$, relying on
   - the theory of regions, and
   - safe Petri nets.
1. Synchronous Transition System of CFSMs
Synchronous Transition System (TS($S$))

Four Player Game (CFSMs)

Synchronous Transition System (TS($S$))
2. **Check for Safety**: *Generalised Multiparty Compatibility* (GMC)

1. **Representability**: ensures that TS contains enough information
   - Essentially: the projection of TS onto a participant must be branching bisimilar to the original machine.

2. **Branching Property**: each branching in TS is “well-formed”
   - it either corresponds to concurrent interactions, or
   - each participant eventually knows which branch was chosen.
3. Build a global graph
We use the work of Cortadella et al. (1998), based on the **theory of regions**, to derive a **safe** and **extended free-choice** Petri net from the Synchronous Transition System.
2: From Petri Net to One-source Petri Net
3: From One-source Petri Net to Joined Petri Net
4 & 5: From Joined Petri Net to Global Graph
Prototype Implementation

https://bitbucket.org/julien-lange/gmc-synthesis

J. Lange, E. Tuosto, N. Yoshida
From Communicating Machines to Graphical Choreographies
Conclusions and Future Work

- Summing up:
  - An effective way of checking properties of CFSMs, and whether one can construct a global graph from them.
  - An algorithm based on the theory of regions.
  - A CFSMs characterisation of well-formed *graphical global types*.
- Tool(s):
  - [http://bitbucket.org/julien-lange/gmc-synthesis](http://bitbucket.org/julien-lange/gmc-synthesis)
  - [http://bitbucket.org/emlio_tuosto/gmc-synthesis-v0.2](http://bitbucket.org/emlio_tuosto/gmc-synthesis-v0.2)

L. Bocchi, J. L., and N. Yoshida: Meeting Deadlines Together. To appear in CONCUR’15
Conclusions and Future Work

- Summing up:
  - An effective way of checking properties of CFSMs, and whether one can construct a global graph from them.
  - An algorithm based on the theory of regions.
  - A CFSMs characterisation of well-formed graphical global types.
- Tool(s):
  - [http://bitbucket.org/julien-lange/gmc-synthesis](http://bitbucket.org/julien-lange/gmc-synthesis)
  - [http://bitbucket.org/emlio_tuosto/gmc-synthesis-v0.2](http://bitbucket.org/emlio_tuosto/gmc-synthesis-v0.2)
- Recent extension to support **Communicating Timed Automata**:
  - To appear in CONCUR’15
Thanks!

Any questions?

https://bitbucket.org/julien-lange/gmc-synthesis