# Formal Choreographic Languages

Ivan Lanese

University of Bologna, Italy/INRIA, France

Joint work with Franco Barbanera (University of Catania, Italy)
and Emilio Tuosto (Gran Sasso Science Institute, Italy)

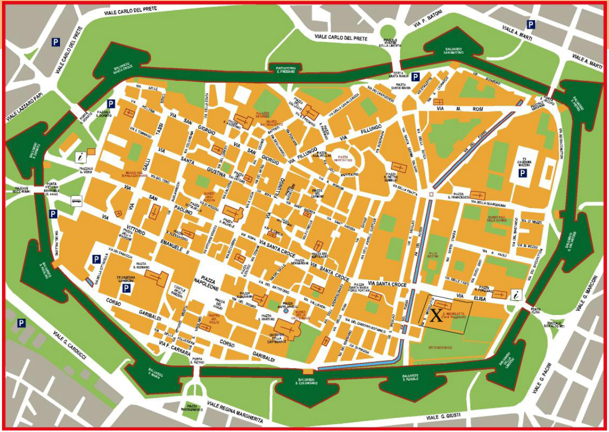
# Outline

Choreographic formalisms

Choreographic languages

Properties

Conclusion



# Choreographic formalisms: why?

Programming multi-party message-passing systems is difficult and error-prone due to issues such as deadlocks and races.

A number of approaches propose solutions based on the concept of a choreographic description:

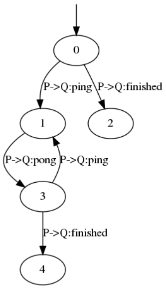
- a blueprint of the message passing behavior of the system
- good as specification of the overall behavior
- specification of local participants can be generated
- some good properties may hold by construction (deadlock freedom, ...)

# Choreographic formalisms: how?

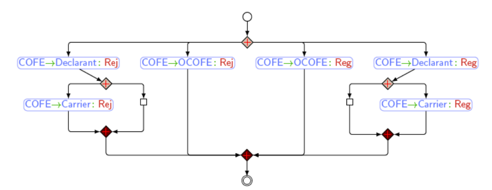
Behavior defined by composing interactions A→B:m

A→B:m: participant A sends a message m to participant B, and B receives it

Allowed sequences of interactions generated in many ways: process algebras (multiparty session types), automata, graphs, programs, ...




```
global protocol OnlineWallet
(role Wallet, role Customer, role Vendor) {
  rec AuthLoop {
    login(account: int) from Customer to Wallet;
    pin(pin: int) from Customer to Wallet;
    choice at Wallet {
      login_ok() from Wallet to Customer;
      login_ok() from Wallet to Vendor;
```




Various constraints posed by the syntax of the formalism, or added on top of it to ensure properties of interest

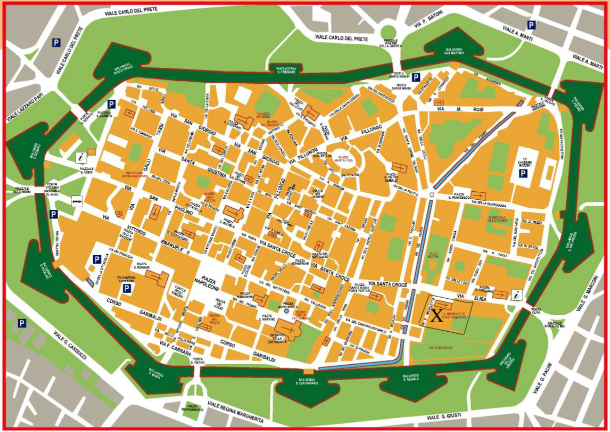
# Outline

Choreographic formalisms

Choreographic languages

Properties

Conclusion



# Choreographic languages: why and how?

Our question: what can be done working at the level of languages of interactions only?

- abstracting away how they are generated

**Definition (Global language)**

Languages of finite and infinite words on the alphabet:

$$\Sigma_{\mathrm{int}} = \{\, \mathsf{A}{\rightarrow}\mathsf{B}{:}\mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P}, \mathsf{m} \in \mathfrak{M} \,\}$$

# Choreographic languages: why and how?

Our question: what can be done working at the level of languages of interactions only?

- abstracting away how they are generated

**Definition (Global language)**

Languages of finite and infinite words on the alphabet:

$$\Sigma_{\text{int}} = \{\, \mathsf{A} \to \mathsf{B} : \mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P}, \mathsf{m} \in \mathfrak{M} \,\}$$

We use languages also to describe single participants:

**Definition (Local language)**

Languages of finite and infinite words on the alphabet:

$$\Sigma_{\text{act}} = \{\, \mathsf{A}\,\mathsf{B}!\mathsf{m}, \mathsf{A}\,\mathsf{B}?\mathsf{m} \mid \mathsf{A} \neq \mathsf{B} \in \mathfrak{P}, \mathsf{m} \in \mathfrak{M} \,\}$$

We consider only prefix-closed languages.

# An advantage

Many formalisms are restricted to regular languages (e.g., automata).

We have no such constraint.

**Example (A context-free language)**

$$
\begin{array}{rcll}
S & ::= & S' \cdot \mathsf{S}{\to}\mathsf{D}{:}\mathsf{s} \cdot \mathsf{S}{\to}\mathsf{H}{:}\mathsf{s} & \\
S' & ::= & \mathsf{S}{\to}\mathsf{D}{:}\mathsf{a} \cdot \mathsf{D}{\to}\mathsf{S}{:}\mathsf{t} \cdot \mathsf{S}{\to}\mathsf{H}{:}\mathsf{t} \cdot S' \cdot \mathsf{S}{\to}\mathsf{H}{:}\mathsf{r} \cdot \mathsf{H}{\to}\mathsf{S}{:}\mathsf{r} \cdot \mathsf{S}{\to}\mathsf{D}{:}\mathsf{d} \cdot S' & \\
 & \mid & \mathsf{S}{\to}\mathsf{D}{:}\mathsf{a} \cdot \mathsf{D}{\to}\mathsf{S}{:}\mathsf{t} \cdot \mathsf{S}{\to}\mathsf{D}{:}\mathsf{d} \cdot S' & \mid \ \varepsilon
\end{array}
$$

Dispatcher D sends tasks t to server S. The server can either send the resulting data d to dispathcer, or send tasks to some helper H and resume r them later on.

Not regular: same structure as balanced parenthesis.

We can prove properties about it!

# Projection

The projecton on $\mathsf{C}$ of an interaction $\mathsf{A}{\rightarrow}\mathsf{B}{:}\mathsf{m}$ is defined as:

$$(\mathsf{A}{\rightarrow}\mathsf{B}{:}\mathsf{m})\downarrow_{\mathsf{C}} = \begin{cases} \mathsf{A}\,\mathsf{B}!\mathsf{m} \text{ if } \mathsf{C} = \mathsf{A} \\ \mathsf{A}\,\mathsf{B}?\mathsf{m} \text{ if } \mathsf{C} = \mathsf{B} \\ \varepsilon \text{ if } \mathsf{C} \neq \mathsf{A}, \mathsf{B} \end{cases}$$

The definition extends homomorphically to words and languages.

The projection $\mathcal{L}\downarrow$ of a g-language $\mathcal{L}$ is the communicating system $(\mathcal{L}\downarrow_{\mathsf{A}})_{\mathsf{A}\in\mathsf{ptp}(\mathcal{L})}$.

# Projection

The projecton on $\mathsf{C}$ of an interaction $\mathsf{A}\to\mathsf{B}:\mathsf{m}$ is defined as:

$$(\mathsf{A}\to\mathsf{B}:\mathsf{m})\downarrow_{\mathsf{C}} = \begin{cases} \mathsf{A}\,\mathsf{B}!\mathsf{m} \text{ if } \mathsf{C} = \mathsf{A} \\ \mathsf{A}\,\mathsf{B}?\mathsf{m} \text{ if } \mathsf{C} = \mathsf{B} \\ \varepsilon \text{ if } \mathsf{C} \neq \mathsf{A}, \mathsf{B} \end{cases}$$

The definition extends homomorphically to words and languages.

The projection $\mathcal{L}\downarrow$ of a g-language $\mathcal{L}$ is the communicating system $(\mathcal{L}\downarrow_{\mathsf{A}})_{\mathsf{A}\in \mathsf{ptp}(\mathcal{L})}$.

**Example**

$\mathcal{L} = \{\, \mathsf{C}\to\mathsf{A}:\mathsf{m}\cdot\mathsf{A}\to\mathsf{B}:\mathsf{m},\ \mathsf{C}\to\mathsf{B}:\mathsf{m}\cdot\mathsf{A}\to\mathsf{B}:\mathsf{m},\ \mathsf{C}\to\mathsf{A}:\mathsf{m}\cdot\mathsf{C}\to\mathsf{B}:\mathsf{m} \,\}$ closed under prefix

$\mathcal{L}\downarrow_{\mathsf{A}} = \{\, \varepsilon,\ \mathsf{C}\,\mathsf{A}?\mathsf{m},\ \mathsf{C}\,\mathsf{A}?\mathsf{m}\cdot\mathsf{A}\,\mathsf{B}!\mathsf{m},\ \mathsf{A}\,\mathsf{B}!\mathsf{m} \,\}$

$\mathcal{L}\downarrow_{\mathsf{B}} = \{\, \varepsilon,\ \mathsf{A}\,\mathsf{B}?\mathsf{m},\ \mathsf{C}\,\mathsf{B}?\mathsf{m},\ \mathsf{C}\,\mathsf{B}?\mathsf{m}\cdot\mathsf{A}\,\mathsf{B}?\mathsf{m} \,\}$

$\mathcal{L}\downarrow_{\mathsf{C}} = \{\, \varepsilon,\ \mathsf{C}\,\mathsf{A}!\mathsf{m},\ \mathsf{C}\,\mathsf{B}!\mathsf{m},\ \mathsf{C}\,\mathsf{A}!\mathsf{m}\cdot\mathsf{C}\,\mathsf{B}!\mathsf{m} \,\}$

# Semantics

The (synchronous) semantics $[\![S]\!]$ of a communicating system $S$ is the language of words $w$ such as $w{\downarrow_{\mathsf{A}}} \in S(\mathsf{A})$ for each $\mathsf{A}$

**Example**

$\mathcal{L}{\downarrow_{\mathsf{A}}} = \{\, \varepsilon,\ \mathsf{C\,A?m},\ \mathsf{C\,A?m} \cdot \mathsf{A\,B!m},\ \mathsf{A\,B!m}\,\}$
$\mathcal{L}{\downarrow_{\mathsf{B}}} = \{\, \varepsilon,\ \mathsf{A\,B?m},\ \mathsf{C\,B?m},\ \mathsf{C\,B?m} \cdot \mathsf{A\,B?m}\,\}$
$\mathcal{L}{\downarrow_{\mathsf{C}}} = \{\, \varepsilon,\ \mathsf{C\,A!m},\ \mathsf{C\,B!m},\ \mathsf{C\,A!m} \cdot \mathsf{C\,B!m}\,\}$

$[\![\mathcal{L}{\downarrow}]\!] = \{\, \mathsf{C{\to}A{:}m} \cdot \mathsf{A{\to}B{:}m},\ \mathsf{C{\to}B{:}m} \cdot \mathsf{A{\to}B{:}m},\ \boxed{\mathsf{C{\to}A{:}m} \cdot \mathsf{C{\to}B{:}m} \cdot \mathsf{A{\to}B{:}m}}\,\}$ closed under prefix

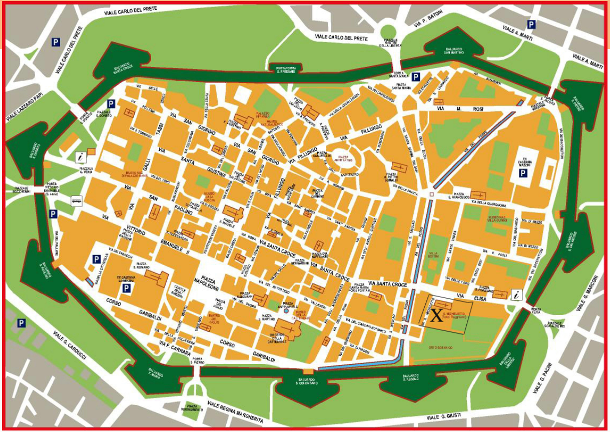
# Outline

Choreographic formalisms

Choreographic languages

Properties

Conclusion



# Correctness & completeness

**Definition**

A system $S$ is *correct w.r.t. a g-language* $\mathcal{L}$ if $[\![S]\!] \subseteq \mathcal{L}$
*complete* if $[\![S]\!] \supseteq \mathcal{L}$

$\mathcal{L}\downarrow$ is always complete w.r.t. $\mathcal{L}$.

However, it may not be correct.

# Characterizing correctness

**Definition (Closure under unknown information)**

$\mathsf{cui}(\mathcal{L})$ if for all $w_1 \cdot \mathsf{A} \to \mathsf{B} : \mathsf{m}, w_2 \cdot \mathsf{A} \to \mathsf{B} : \mathsf{m}, w \in \mathcal{L}$
$w \downarrow_{\mathsf{A}} = w_1 \downarrow_{\mathsf{A}}$ and $w \downarrow_{\mathsf{B}} = w_2 \downarrow_{\mathsf{B}}$
imply $w \cdot \mathsf{A} \to \mathsf{B} : \mathsf{m} \in \mathcal{L}$

# Characterizing correctness

**Definition (Closure under unknown information)**

$\mathsf{cui}(\mathcal{L})$ if for all $w_1 \cdot \mathsf{A} \to \mathsf{B} : \mathsf{m}$, $w_2 \cdot \mathsf{A} \to \mathsf{B} : \mathsf{m}$, $w \in \mathcal{L}$
$w \downarrow_{\mathsf{A}} = w_1 \downarrow_{\mathsf{A}}$ and $w \downarrow_{\mathsf{B}} = w_2 \downarrow_{\mathsf{B}}$
imply $w \cdot \mathsf{A} \to \mathsf{B} : \mathsf{m} \in \mathcal{L}$

**Theorem**

*If $\mathcal{L}\downarrow$ is correct w.r.t. $\mathcal{L}$ then $\mathsf{cui}(\mathcal{L})$ holds. If $\mathcal{L}$ is a standard or continuous and $\mathsf{cui}(\mathcal{L})$ then $\mathcal{L}\downarrow$ is correct w.r.t. $\mathcal{L}$.*

# Characterizing correctness

**Definition (Closure under unknown information)**

$\mathsf{cui}(\mathcal{L})$ if for all $w_1 \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, $w_2 \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m}$, $w \in \mathcal{L}$
$w \downarrow_{\mathsf{A}} = w_1 \downarrow_{\mathsf{A}}$ and $w \downarrow_{\mathsf{B}} = w_2 \downarrow_{\mathsf{B}}$
imply $w \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \in \mathcal{L}$

**Theorem**

*If $\mathcal{L}\!\downarrow$ is correct w.r.t. $\mathcal{L}$ then $\mathsf{cui}(\mathcal{L})$ holds. If $\mathcal{L}$ is a standard or continuous and $\mathsf{cui}(\mathcal{L})$ then $\mathcal{L}\!\downarrow$ is correct w.r.t. $\mathcal{L}$.*

**Example**

$\mathcal{L} = \{\, \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m},\ \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m} \cdot \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m} \,\}$ closed under prefix

Not $\mathsf{cui}(\mathcal{L})$.

$w_1 = \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m} \qquad w_2 = \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}$
$w = \mathsf{C}{\to}\mathsf{A}{:}\mathsf{m} \cdot \mathsf{C}{\to}\mathsf{B}{:}\mathsf{m}$

# Is this enough? NO!

Languages which are correct and complete may generate “bad” systems, e.g., systems that deadlock.

**Deadlock-freedom (DF)**: for each participant A, A has no pending actions in maximal computations.

**Example**

$\mathcal{L} = \{ w = \mathsf{A}{\to}\mathsf{C}{:}\mathsf{l} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \cdot \mathsf{A}{\to}\mathsf{C}{:}\mathsf{m},\ w' = \mathsf{A}{\to}\mathsf{C}{:}\mathsf{r} \cdot \mathsf{A}{\to}\mathsf{B}{:}\mathsf{m} \cdot \mathsf{B}{\to}\mathsf{C}{:}\mathsf{m} \}$
closed under prefix

$\mathsf{cui}(\mathcal{L})$: A and C know which word has been taken.

$\mathcal{L}{\downarrow}$ is not deadlock-free since $w$ is a deadlock:

- finite maximal word in $\mathcal{L}$, but
- $w{\downarrow_{\mathsf{B}}} = \mathsf{A}\,\mathsf{B}?\mathsf{m}$ is not maximal in $\mathcal{L}{\downarrow_{\mathsf{B}}}$ because $w'{\downarrow_{\mathsf{B}}} = \mathsf{A}\,\mathsf{B}?\mathsf{m} \cdot \mathsf{B}\,\mathsf{C}!\mathsf{m} \in \mathcal{L}{\downarrow_{\mathsf{B}}}$

After $\mathsf{A}\,\mathsf{B}?\mathsf{m}$ participant B does not know whether the protocol has finished or not

# Properties of interest

Beyond deadlock freedom, we may want communication properties to hold.

**Harmonicity (HA):** each local sequence of actions can be executed in some computation of the system.

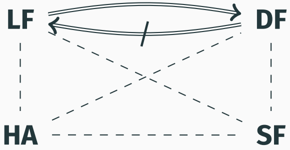
For each A, if A has communications to make on an ongoing computation, then:

**Lock-freedom (LF):** at least one continuation involves A.

**Strong lock-freedom (SLF):** each maximal continuation involves A.

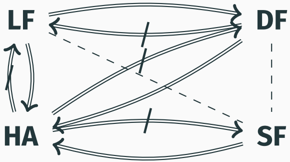
**Starvation-freedom (SF):** each infinite continuation involves A.

# Relations among communication properties




Moreover, DF $\wedge$ SF $\Leftrightarrow$ SLF and SLF $\Rightarrow$ LF.

If systems are projections of g-languages **HA** holds, hence:




Moreover, DF $\wedge$ SF $\Leftrightarrow$ SLF and SLF $\Rightarrow$ LF.

# Ensuring communication properties

**Definition (Branch-awareness)**

A g-language $\mathcal{L}$ on $\mathcal{P}$ is *branch-aware* if for each $\mathsf{X} \in \mathcal{P}$ and for each pair of maximal words $w_1$ and $w_2$ in $\mathcal{L}$ if $w_1{\downarrow_{\mathsf{X}}} \neq w_2{\downarrow_{\mathsf{X}}}$ then $w_1{\downarrow_{\mathsf{X}}} \not< w_2{\downarrow_{\mathsf{X}}}$ and $w_2{\downarrow_{\mathsf{X}}} \not< w_1{\downarrow_{\mathsf{X}}}$.

Neither projection should be a strict prefix of the other.

This was not the case in previous example.

# Ensuring communication properties

**Definition (Branch-awareness)**

A g-language $\mathcal{L}$ on $\mathcal{P}$ is *branch-aware* if for each $\mathsf{X} \in \mathcal{P}$ and for each pair of maximal words $w_1$ and $w_2$ in $\mathcal{L}$ if $w_1{\downarrow_{\mathsf{X}}} \neq w_2{\downarrow_{\mathsf{X}}}$ then $w_1{\downarrow_{\mathsf{X}}} \not< w_2{\downarrow_{\mathsf{X}}}$ and $w_2{\downarrow_{\mathsf{X}}} \not< w_1{\downarrow_{\mathsf{X}}}$.

Neither projection should be a strict prefix of the other.

This was not the case in previous example.

**Proposition (Branch-awareness characterises SLF)**

*A CUI g-language $\mathcal{L}$ is branch-aware iff $\mathcal{L}{\downarrow}$ is strongly lock-free.*

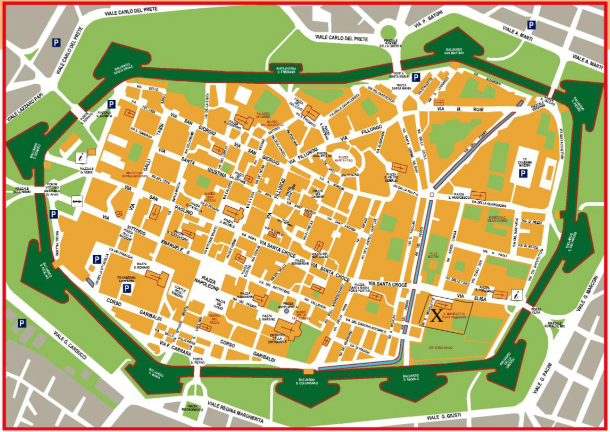
All the other properties follow.

Choreographic formalisms

Choreographic languages

Properties

Conclusion



# Summary

- We have presented a general framework able to capture many choreographic formalisms
- In the paper we detail the cases of
  - global types (from P. Severi and M. Dezani-Ciancaglini. Observational equivalence for multiparty sessions. Fundam. Informaticae (2019))
  - choreography automata (our COORDINATION 2020)
- Correctness ensured by a closure property, instead of by forbidding computations
- We separate conditions for correctness and conditions for behavioral properties
- Our approach can prove properties of non-regular languages

# Future work

- Fitting in our framework other models from the literature
- Extending the framework to cope with asynchronous communication
- Drop prefix closure

# Thanks!

# Questions?