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 \( A \rightarrow B: m \)

\( A \rightarrow B: m \): 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, ...

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

\[ \Sigma_{\text{int}} = \{ \text{A} \rightarrow \text{B}: m \mid \text{A} \neq \text{B} \in \mathcal{P}, m \in \mathcal{M} \} \]
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:

\[ \Sigma_{\text{int}} = \{ A \rightarrow B : m \mid A \neq B \in \mathcal{P}, m \in M \} \]

We use languages also to describe single participants:

**Definition (Local language)**
Languages of finite and infinite words on the alphabet:

\[ \Sigma_{\text{act}} = \{ A B! m, A B? m \mid A \neq B \in \mathcal{P}, m \in M \} \]

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)**

<table>
<thead>
<tr>
<th>S</th>
<th>::=</th>
<th>S' · S → D → s · S → H · s</th>
</tr>
</thead>
<tbody>
<tr>
<td>S'</td>
<td>::=</td>
<td>S → D → a · D → S → t · S → H · t · S → H · r · H → S → r · S → D → d · S'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S → D → a · D → S → t · S → D → d · S'</td>
</tr>
</tbody>
</table>

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

Not regular: same structure as balanced parenthesis.

We can prove properties about it!
The projecton on $C$ of an interaction $A \rightarrow B : m$ is defined as:

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

The definition extends homomorphically to words and languages.

The projection $\mathcal{L} \downarrow$ of a g-language $\mathcal{L}$ is the communicating system $(\mathcal{L} \downarrow A)_{A \in \text{ptp}(\mathcal{L})}$. 
The projecton on \( C \) of an interaction \( A \rightarrow B: m \) is defined as:

\[
(A \rightarrow B: m) \downarrow_C = \begin{cases} 
A \land m & \text{if } C = A \\
A \lor m & \text{if } C = B \\
\varepsilon & \text{if } C \neq A, B
\end{cases}
\]

The definition extends homomorphically to words and languages.

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

**Example**

\( \mathcal{L} = \{ C \rightarrow A: m \cdot A \rightarrow B: m, C \rightarrow B: m \cdot A \rightarrow B: m, C \rightarrow A: m \cdot C \rightarrow B: m \} \) closed under prefix

\( \mathcal{L} \downarrow_A = \{ \varepsilon, CA?m, CA?m \cdot A \land m, A \land m \} \)

\( \mathcal{L} \downarrow_B = \{ \varepsilon, AB?m, CB?m, CB?m \cdot AB?m \} \)

\( \mathcal{L} \downarrow_C = \{ \varepsilon, CA\land m, CB\land m, CA\land m \cdot CB\land m \} \)
The (synchronous) semantics $[S]$ of a communicating system $S$ is the language of words $w$ such as $w \downarrow_A \in S(A)$ for each $A$

**Example**

$L \downarrow_A = \{ \varepsilon, CA?m, CA?m \cdot AB!m, AB!m \}$

$L \downarrow_B = \{ \varepsilon, AB?m, CB?m, CB?m \cdot AB?m \}$

$L \downarrow_C = \{ \varepsilon, CA!m, CB!m, CA!m \cdot CB!m \}$

$[L \downarrow] = \{ C \rightarrow A:m \cdot A \rightarrow B:m, C \rightarrow B:m \cdot A \rightarrow B:m, C \rightarrow A:m \cdot C \rightarrow B:m \cdot A \rightarrow B:m \}$ 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 $\llbracket S \rrbracket \subseteq \mathcal{L}$

**complete** if $\llbracket S \rrbracket \supseteq \mathcal{L}$

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

However, it may not be correct.
Characterizing correctness

Definition (Closure under unknown information)

cui(\mathcal{L}) \text{ if for all } w_1 \cdot A \rightarrow B: m, w_2 \cdot A \rightarrow B: m, w \in \mathcal{L} \text{\ implies} \\
w \downarrow_A = w_1 \downarrow_A \text{ and } w \downarrow_B = w_2 \downarrow_B \text{\ imply } w \cdot A \rightarrow B: m \in \mathcal{L}
Characterizing correctness

**Definition (Closure under unknown information)**

\( \text{cui}(\mathcal{L}) \) if for all \( w_1 \cdot A \rightarrow B : m, w_2 \cdot A \rightarrow B : m, w \in \mathcal{L} \)

\( w \downarrow_A = w_1 \downarrow_A \) and \( w \downarrow_B = w_2 \downarrow_B \)

imply \( w \cdot A \rightarrow B : m \in \mathcal{L} \)

**Theorem**

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

then \( \mathcal{L} \downarrow \) is correct w.r.t. \( \mathcal{L} \).
Characterizing correctness

<table>
<thead>
<tr>
<th>Definition (Closure under unknown information)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{cui}(\mathcal{L}) ) if for all ( w_1 \cdot A \rightarrow B : m, w_2 \cdot A \rightarrow B : m, w \in \mathcal{L} ) |</td>
</tr>
<tr>
<td>( w \downarrow_A = w_1 \downarrow_A ) and ( w \downarrow_B = w_2 \downarrow_B ) |</td>
</tr>
<tr>
<td>imply ( w \cdot A \rightarrow B : m \in \mathcal{L} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ( \mathcal{L} \downarrow ) is correct w.r.t. ( \mathcal{L} ) then ( \text{cui}(\mathcal{L}) ) holds. If ( \mathcal{L} ) is a standard or continuous and ( \text{cui}(\mathcal{L}) ) then ( \mathcal{L} \downarrow ) is correct w.r.t. ( \mathcal{L} ).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{L} = { C \rightarrow A : m \cdot A \rightarrow B : m, C \rightarrow B : m \cdot A \rightarrow B : m, C \rightarrow A : m \cdot C \rightarrow B : m } ) closed under prefix |</td>
</tr>
<tr>
<td>Not ( \text{cui}(\mathcal{L}) ). |</td>
</tr>
<tr>
<td>( w_1 = C \rightarrow A : m \</td>
</tr>
<tr>
<td>( w_2 = C \rightarrow B : m \</td>
</tr>
<tr>
<td>( w = C \rightarrow A : m \cdot C \rightarrow B : m \</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L} = { w = A \rightarrow C : l \cdot A \rightarrow B : m \cdot A \rightarrow C : m, \ w' = A \rightarrow C : r \cdot A \rightarrow B : m \cdot B \rightarrow C : m }$</td>
</tr>
<tr>
<td>closed under prefix</td>
</tr>
<tr>
<td>$\text{cui}(\mathcal{L})$: $A$ and $C$ know which word has been taken.</td>
</tr>
<tr>
<td>$\mathcal{L}_{down}$ is not deadlock-free since $w$ is a deadlock:</td>
</tr>
<tr>
<td>• finite maximal word in $\mathcal{L}$, but</td>
</tr>
<tr>
<td>• $w_{down_B} = A B ? m$ is not maximal in $\mathcal{L}<em>{down_B}$ because $w'</em>{down_B} = A B ? m \cdot B C ! m \in \mathcal{L}_{down_B}$</td>
</tr>
<tr>
<td>After $A B ? m$ participant $B$ does not know whether the protocol has finished or not</td>
</tr>
</tbody>
</table>
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, $DF \land SF \iff SLF$ and $SLF \Rightarrow LF$.

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

Moreover, $DF \land SF \iff SLF$ and $SLF \Rightarrow LF$. 
Ensuring communication properties

Definition (Branch-awareness)

A g-language $L$ on $P$ is *branch-aware* if for each $X \in P$ and for each pair of maximal words $w_1$ and $w_2$ in $L$ if $w_1 \downarrow_X \neq w_2 \downarrow_X$ then $w_1 \downarrow_X \not\prec w_2 \downarrow_X$ and $w_2 \downarrow_X \not\prec w_1 \downarrow_X$.

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 $X \in \mathcal{P}$ and for each pair of maximal words $w_1$ and $w_2$ in $\mathcal{L}$ if $w_1 \downarrow_X \neq w_2 \downarrow_X$ then $w_1 \downarrow_X \not\prec w_2 \downarrow_X$ and $w_2 \downarrow_X \not\prec w_1 \downarrow_X$.

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.
Outline

Choreographic formalisms

Choreographic languages

Properties

Conclusion
• 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?