2008 Sixth IEEE International Conference on Software Engineering and Formal Methods

Bridging the gap between Interaction- and Process-Oriented Choreographies*

Ivan Lanese Claudio Guidi Fabrizio Montesi Gianluigi Zavattaro
Computer Science Department, University of Bologna
Mura A. Zamboni, 7, 40127 Bologna, Italy
{*lanese,cguidi,fmontesi,zavattar*}*@cs.unibo.it*

Abstract

In service oriented computing, choreography languages are used to specify multi-party service compositions. Two main approaches have been followed: the interaction-oriented approach of WS-CDL and the process-oriented approach of BPEL4Chor. We investigate the relationship between them. In particular, we consider several interpretations for interaction-oriented choreographies spanning from synchronous to asynchronous communication. Under each of these interpretations we characterize the class of interaction-oriented choreographies which have a process-oriented counterpart, and we formalize the notion of equivalence between the initial interaction-oriented choreography and the corresponding process-oriented one.

1 Introduction

Choreography languages are attracting a lot of attention within the Service Oriented Computing (SOC) research community. They are intended as notations for representing multi-party service compositions, that is, descriptions of the global behavior of service-based applications in which several services reciprocally communicate in order to complete a predefined task.

Despite the recognized need for a standard choreography language to be used by service application developers, two main approaches are currently followed. On the one hand, W3C is developing the Web Services Choreography Description Language WS-CDL [Wor]. On the other hand, the research community around the Web Service Business Process Execution Languages WS-BPEL [OAS] is investigating BPEL4Chor [DKLW07].

*Research partially funded by EU Integrated Project SENSORIA, contract n. 016004.

In WS-CDL, the basic activities in a service choreography are interactions, that is, the atomic execution of a send and a receive operations performed by two communicating partners. For this reason, we say that WS-CDL follows an *interaction-oriented* approach. On the contrary, in BPEL4Chor the business process of each partner involved in a choreography is specified using an abstract version of BPEL, in which the basic activities are either invoke or receive operations performed by the specified partner. A choreography is the parallel composition of the independently specified business processes. For this reason, we say that BPEL4Chor follows a *process-oriented* approach.

Even if the two approaches are both intended as solutions to the same problem, their relationship is not trivial, and depends on choices about the kind of communication (synchronous or asynchronous) and the local events that are globally observed (either send or receive, or both). In the literature the different alternatives have never been systematically compared.

Our comparison starts with the observation that the interaction-oriented approach supports a more abstract (global) vision of a choreography, in which the send and the receive events of a communication are considered as an atomic entity. On the contrary, the process-oriented approach keeps the more concrete vision of the two distinct send and receive events performed by two separate processes. For instance, consider a trivial interaction-oriented choreography (IOC) specification

$$a \xrightarrow{o} b; c \xrightarrow{o'} d$$

describing an interaction on the operation o between the roles a and b, followed by an interaction on the operation o' between the roles c and d. The naturally correspondent process-oriented choreography (POC) is:

$$(\overline{o})_a \parallel (o)_b \parallel (\overline{o'})_c \parallel (o')_d$$

where we indicate for each role the invoke and the receive operations to be executed.

978-0-7695-3437-4/08 $25.00 © 2008 IEEE
DOI 10.1109/SEFM.2008.11

The two above choreographies could give rise to different behaviors. For instance, in the POC, the communication between c and d could happen before the interaction between a and b. On the contrary, if we consider $a = c$, we obtain as correspondent POC:

$$(\overline{o}; \overline{o'})_a \parallel (o)_b \parallel (o')_d$$

This last POC better matches the initial IOC. In fact, if we consider the order in which the interactions are started (i.e., the send event), the interaction on the operation o surely occurs before the one on o'. On the contrary, if we consider the order in which the interactions are completed (i.e., the receive event), in the case of asynchronous communication the interaction on the operation o could be completed after the one on o'.

The above example shows that the relationship between the interaction- and the process-oriented approaches is strongly influenced by the kind of communication (synchronous or asynchronous) and the events that are observed in the POC (send, receive, or both). In this paper, we consider the relationship between the two approaches for choreography specification under synchronous communication, asynchronous communication, and, in the latter case, we consider the possibility to observe either send or receive, or both events. For each interpretation we obtain the following: (i) the precise characterization of the IOCs which have a direct POC counterpart and (ii) the formalization of the corresponding notion of equivalence between the initial IOC and the corresponding POC.

2 Calculi

In this section we define two basic choreography languages, an Interaction-Oriented Choreography language (IOC) and a Process-Oriented Choreography language (POC).

2.1 Interaction-Oriented Choreography

The syntax of IOC, where we use a, b, ... to range over roles and o to range over operations, is:

$$\mathcal{I} ::= a \xrightarrow{o} b \mid \mathbf{1} \mid \mathbf{0} \mid \mathcal{I};\mathcal{I}' \mid \mathcal{I} \parallel \mathcal{I}' \mid \mathcal{I} + \mathcal{I}'$$

The basic construct is the interaction between two distinct roles a and b on operation o, denoted by $a \xrightarrow{o} b$. In addition there are the empty IOC $\mathbf{1}$, the terminated IOC $\mathbf{0}$, sequential and parallel composition and nondeterministic choice. For instance $(a \xrightarrow{o} b \parallel a \xrightarrow{o'} c); b \xrightarrow{o''} c$ specifies that $a \xrightarrow{o} b$ and $a \xrightarrow{o'} c$ can be performed in any order, and after both of them have been completed then $b \xrightarrow{o''} c$ can be executed.

(Interaction) $\quad a \xrightarrow{o} b \xrightarrow{a \xrightarrow{o} b} \mathbf{1}$

(Sequence) $\quad \dfrac{\mathcal{I} \xrightarrow{\sigma} \mathcal{I}' \quad \sigma \neq \surd}{\mathcal{I};\mathcal{J} \xrightarrow{\sigma} \mathcal{I}';\mathcal{J}}$

(End) $\quad \mathbf{1} \xrightarrow{\surd} \mathbf{0}$

(Parallel) $\quad \dfrac{\mathcal{I} \xrightarrow{\sigma} \mathcal{I}' \quad \sigma \neq \surd}{\mathcal{I} \parallel \mathcal{J} \xrightarrow{\sigma} \mathcal{I}' \parallel \mathcal{J}}$

(Choice) $\quad \dfrac{\mathcal{I} \xrightarrow{\sigma} \mathcal{I}'}{\mathcal{I} + \mathcal{J} \xrightarrow{\sigma} \mathcal{I}'}$

(Seq-end) $\quad \dfrac{\mathcal{I} \xrightarrow{\surd} \mathcal{I}' \quad \mathcal{J} \xrightarrow{\sigma} \mathcal{J}'}{\mathcal{I};\mathcal{J} \xrightarrow{\sigma} \mathcal{J}'}$

(Par-end) $\quad \dfrac{\mathcal{I} \xrightarrow{\surd} \mathcal{I}' \quad \mathcal{J} \xrightarrow{\surd} \mathcal{J}'}{\mathcal{I} \parallel \mathcal{J} \xrightarrow{\surd} \mathcal{I}' \parallel \mathcal{J}'}$

Table 1. IOC semantics

We give an LTS semantics to IOCs. The rules are in Table 1, and are standard (see, e.g., [BZ07]). We use σ to range over labels. Symmetric rules for parallel composition and choice have been omitted.

We define the function roles($\mathcal{I}$) that given an IOC $\mathcal{I}$ computes the set of roles in it as:
roles($a \xrightarrow{o} b$) = $\{a, b\}$ $\qquad$ roles($\mathbf{1}$) = roles($\mathbf{0}$) = $\emptyset$
roles($\mathcal{I};\mathcal{I}'$) = roles($\mathcal{I} \parallel \mathcal{I}'$) = roles($\mathcal{I}$) $\cup$ roles($\mathcal{I}'$)
roles($\mathcal{I} + \mathcal{I}'$) = roles($\mathcal{I}$) $\cup$ roles($\mathcal{I}'$)

2.2 Process-Oriented Choreography

POC describes processes, ranged over by P, and systems, ranged over by $\mathcal{S}$.

$$\begin{array}{lcl} P & ::= & o \mid \overline{o} \mid \mathbf{1} \mid \mathbf{0} \mid P;P' \mid P \mid P' \mid P + P' \\ \mathcal{S} & ::= & (P)_a \mid \mathcal{S} \parallel \mathcal{S}' \end{array}$$

Processes include input o and output $\overline{o}$ on operation o, the empty and the terminated process, sequential and parallel composition and nondeterministic choice. The runtime syntax includes also messages $\langle o \rangle$. Systems are parallel compositions of roles. Each role includes a unique role name and a process.

We define two semantics for POC: synchronous and asynchronous. In the synchronous semantics input and output interact atomically, while in the asynchronous one the output creates a message that can later interact with the corresponding input.

The LTS for the asynchronous semantics is in Table 2. We use γ to range over labels. Symmetric rules for parallel composition and choice have been omitted. The unique nonstandard rules are those dealing with asynchronous communication. Msg produces a message $\langle o \rangle$ when a process performs an output operation

$$\text{(IN)}\quad o \xrightarrow{o} \mathbf{1} \qquad \text{(OUT)}\quad \overline{o} \xrightarrow{\overline{o}} \mathbf{1} \qquad \text{(ASYNC-OUT)}\quad \langle o\rangle \xrightarrow{\langle o\rangle} \mathbf{1} \qquad \text{(ONE)}\quad \mathbf{1} \xrightarrow{\surd} \mathbf{0}$$

$$\text{(SEQUENCE)}\ \frac{P \xrightarrow{\gamma} P' \quad \gamma \neq \surd}{P;Q \xrightarrow{\gamma} P';Q} \qquad \text{(INNER PARALLEL)}\ \frac{P \xrightarrow{\gamma} P' \quad \gamma \neq \surd}{P \mid Q \xrightarrow{\gamma} P' \mid Q}$$

$$\text{(CHOICE)}\ \frac{P \xrightarrow{\gamma} P'}{P + Q \xrightarrow{\gamma} P'} \qquad \text{(SEQ-END)}\ \frac{P \xrightarrow{\surd} P' \quad Q \xrightarrow{\gamma} Q'}{P;Q \xrightarrow{\gamma} Q'}$$

$$\text{(INNER PAR-END)}\ \frac{P \xrightarrow{\surd} P' \quad Q \xrightarrow{\surd} Q'}{P \mid Q \xrightarrow{\surd} P' \mid Q'}$$

$$\text{(INNER)}\ \frac{P \xrightarrow{\gamma} P' \quad \gamma \neq \overline{o}, \surd}{(P)_a \xrightarrow{\gamma : a} (P')_a} \qquad \text{(MSG)}\ \frac{P \xrightarrow{\overline{o}} P'}{(P)_a \xrightarrow{\overline{o} : a} (P' \mid \langle o\rangle)_a}$$

$$\text{(SYNCHRO)}\ \frac{\mathcal{S} \xrightarrow{\langle o\rangle : a} \mathcal{S}' \quad \mathcal{S}'' \xrightarrow{o:b} \mathcal{S}'''}{\mathcal{S} \parallel \mathcal{S}'' \xrightarrow{a \xrightarrow{o} b} \mathcal{S}' \parallel \mathcal{S}'''} \qquad \text{(EXT PARALLEL)}\ \frac{\mathcal{S} \xrightarrow{\gamma} \mathcal{S}'}{\mathcal{S} \parallel \mathcal{S}'' \xrightarrow{\gamma} \mathcal{S}' \parallel \mathcal{S}''}$$

Table 2. POC asynchronous semantics

on o. This message can be subsequently consumed by a reader. The synchronous semantics $\rightarrow_s$ differs from the asynchronous one since rules OUT, ASYNC-OUT and MSG are deleted and the rule below is added:

$$\text{(SYNC-OUT)}\quad \overline{o} \xrightarrow{\langle o\rangle}_s \mathbf{1}$$

3 From IOC to POC

In this section we show how to relate the Interaction-Oriented and the Process-Oriented description of a choreography. In particular, given an IOC $\mathcal{I}$ we want to define a system $\mathcal{S}$ implementing it. The idea is to project the IOC on the different roles, and build the system $\mathcal{S}$ as parallel composition of these projections.

Definition 3.1 (Projection function). *Given an IOC $\mathcal{I}$ and a role a, the projection $\mathsf{proj}(\mathcal{I}, a)$ of $\mathcal{I}$ on role a is defined by structural induction on $\mathcal{I}$:*

$$\begin{array}{rcl} \mathsf{proj}(a \xrightarrow{o} b, a) & = & \overline{o} \qquad \mathsf{proj}(a \xrightarrow{o} b, b) \;=\; o \\ \mathsf{proj}(a \xrightarrow{o} b, c) & = & \mathbf{1} \textit{ if } c \neq a, b \\ \mathsf{proj}(\mathbf{1}, a) & = & \mathbf{1} \qquad \mathsf{proj}(\mathbf{0}, a) \;=\; \mathbf{0} \\ \mathsf{proj}(\mathcal{I}; \mathcal{I}', a) & = & \mathsf{proj}(\mathcal{I}, a); \mathsf{proj}(\mathcal{I}', a) \\ \mathsf{proj}(\mathcal{I} \parallel \mathcal{I}', a) & = & \mathsf{proj}(\mathcal{I}, a) \mid \mathsf{proj}(\mathcal{I}', a) \\ \mathsf{proj}(\mathcal{I} + \mathcal{I}', a) & = & \mathsf{proj}(\mathcal{I}, a) + \mathsf{proj}(\mathcal{I}', a) \end{array}$$

We denote with $\parallel_{i \in I} \mathcal{S}_i$ the parallel composition of systems $\mathcal{S}_i$ for each $i \in I$.

Definition 3.2. *Given an IOC $\mathcal{I}$, the associated system $\mathcal{S}$ is defined by:*

$$\mathsf{proj}(\mathcal{I}) = \parallel_{a \in \mathsf{roles}(\mathcal{I})} \mathsf{proj}(\mathcal{I}, a)$$

The projection $\mathsf{proj}(\mathcal{I})$ of a IOC $\mathcal{I}$ is a system that behaves according to $\mathcal{I}$. However, "behaves according to" can be formalized in different ways, and will require different well-formedness conditions to be guaranteed, depending on the kind of properties that one wants to ensure. We will give now an informal description of the different possible relationships, while the rest of the paper is devoted to fully formalize the correspondence in terms of bisimilarity relations, and to discuss the necessary well-formedness conditions.

Let us consider the IOC $\mathcal{I} = a \xrightarrow{o} b; c \xrightarrow{o'} d$ from the Introduction, where a, b, c and d may or may not be distinct. In the system $\mathsf{proj}(\mathcal{I})$ there are two possibly distinct events for each interaction $a \xrightarrow{o} b$ in the IOC: the sending $\overline{o} : a$ and the reception $a \xrightarrow{o} b$. Let us denote with s_1 and s_2 the sending events from $a \xrightarrow{o} b$ and $c \xrightarrow{o'} d$ respectively, and similarly let us denote with r_1 and r_2 the corresponding receive events. We denote with e an arbitrary event, write $e_1 = e_2$ when the two events are synchronized and $e_1 < e_2$ when e_1 happens before e_2.

The condition that $a \xrightarrow{o} b$ has to be executed before $c \xrightarrow{o'} d$, expressed by the ; in the IOC, has to be mapped into a condition relating the corresponding events in the POC. We consider the following possibilities, a synchronous one and four asynchronous ones:

Synchronous semantics: it guarantees that the POC behaves as specified by the IOC when executed using the synchronous LTS. Because of synchronous semantics $s_1 = r_1$ and $s_2 = r_2$, thus the sequentiality condition can be expressed as $s_1 < s_2 \vee s_1 < r_2 \vee r_1 < s_2 \vee r_1 < r_2$;

Sender semantics: it guarantees that the sequentiality condition is verified from a sender perspective, i.e. that $s_1 < s_2$;

Receiver semantics: it guarantees that the sequentiality condition is verified from a receiver perspective, i.e. that $r_1 < r_2$;

Sender-receiver semantics: it guarantees that the sequentiality condition is verified from both a sender and a receiver perspective, i.e. that $s_1 < s_2 \wedge r_1 < r_2$; in the following we will not consider this semantics since it is simply the intersection of the sender semantics and the receiver semantics;

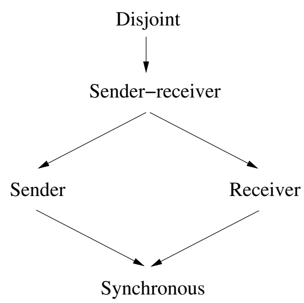

Figure 1. Partial order for connectedness.

Disjoint semantics: it requires that the intervals of execution of the first interaction (from s_1 to r_1) and of the second one (from s_2 to r_2) are completely disjoint: this can be formalized by $r_1 < s_2$.

The different conditions presented above form a partial order w.r.t. implication, e.g., if a system satisfies the conditions of the disjoint semantics then it also satisfies the conditions of the other semantics. The partial order is represented in Figure 1. The same implications are still satisfied when we generalize the conditions to take into account more complex choreographies.

Since in the POC different roles are executing in parallel, in order to enforce the conditions above the same role should occur in different interactions. We show below the conditions on roles required to enforce the semantics discussed above in our simple example:

Synchronous semantics : $\{a, b\} \cap \{c, d\} \neq \emptyset$;

Sender semantics : $c = a \vee c = b$;

Receiver semantics : $d = b \vee c = b$;

Disjoint semantics : $b = c$.

Let us analyze for instance the sender semantics: we require that the sending from c happens after the sending from a. If $a = c$ then a can enforce this condition. However if $b = c$ then b, when it receives the message, knows that the message has been sent, and thus can enforce the condition. We call this condition connectedness for sequence.

Similar conditions are required to ensure that all the participants are aware of which branch of a nondeterministic choice has been taken (we call it existence of a unique point of choice), and that different interactions using the same operation do not mix up (we call it causality-safety). We will refer to all these conditions as connectedness.

4 Synchronous conformance

In this section we discuss conformance and connectedness in the synchronous case.

We will formalize the notion of conformance between a IOC and a POC using bisimilarity [Mil89]. In the synchronous case a form of strong bisimilarity is used. Similar characterizations can also be given using trace equivalence [Hoa85].

Definition 4.1 (Synchronous bisimilarity). *A synchronous bisimulation is a relation R between IOCs and POCs such that if $(\mathcal{I}, \mathcal{S}) \in R$ then:*

- *if $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ then $\mathcal{S} \xrightarrow{a \xrightarrow{o} b}_s \mathcal{S}'$ and $(\mathcal{I}', \mathcal{S}') \in R$;*
- *if $\mathcal{S} \xrightarrow{a \xrightarrow{o} b}_s \mathcal{S}'$ then $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ and $(\mathcal{I}', \mathcal{S}') \in R$.*

Synchronous bisimilarity $\sim_s$ is the largest synchronous bisimulation.

The aim of this section is to give all the tools to (make formal and) prove the following theorem:

Theorem 4.1 (Synchronous conformance). *Let $\mathcal{I}$ be IOC and $\mathcal{S} = \text{proj}(\mathcal{I})$ be its projection. If $\mathcal{I}$ satisfies the connectedness conditions for the synchronous semantics then $\mathcal{I} \sim_s \mathcal{S}$.*

We start by formalizing the connectedness conditions for the synchronous semantics. A few auxiliary functions are needed. Functions $\text{transI}(\bullet)$ and $\text{transF}(\bullet)$ compute respectively the sets of initial and final interactions in a IOC:
$\text{transI}(a \xrightarrow{o} b) = \text{transF}(a \xrightarrow{o} b) = \{a \xrightarrow{o} b\}$
$\text{transI}(\mathbf{1}) = \text{transI}(\mathbf{0}) = \text{transF}(\mathbf{1}) = \text{transF}(\mathbf{0}) = \emptyset$
$\text{transI}(\mathcal{I} \parallel \mathcal{I}') = \text{transI}(\mathcal{I} + \mathcal{I}') = \text{transI}(\mathcal{I}) \cup \text{transI}(\mathcal{I}')$
$\text{transF}(\mathcal{I} \parallel \mathcal{I}') = \text{transF}(\mathcal{I}) \cup \text{transF}(\mathcal{I}')$
$\text{transF}(\mathcal{I} + \mathcal{I}') = \text{transF}(\mathcal{I}) \cup \text{transF}(\mathcal{I}')$
$\text{transI}(\mathcal{I}; \mathcal{I}') = \text{transI}(\mathcal{I}')$ if $\mathcal{I} \xrightarrow{\surd}$, $\text{transI}(\mathcal{I})$ otherwise
$\text{transF}(\mathcal{I}; \mathcal{I}') = \text{transF}(\mathcal{I})$ if $\mathcal{I}' \xrightarrow{\surd}$, $\text{transF}(\mathcal{I}')$ otherwise

Definition 4.2 (Synchronous connectedness for sequence). *An IOC $\mathcal{I}$ is synchronous connected for sequence if for each subterm of the form $\mathcal{I}; \mathcal{J}$ we have $\forall a \xrightarrow{o} b \in \text{transF}(\mathcal{I}), \forall c \xrightarrow{o'} d \in \text{transI}(\mathcal{J}), \{a, b\} \cap \{c, d\} \neq \emptyset$.*

Definition 4.3 (Synchronous unique point of choice). *An IOC $\mathcal{I}$ has synchronous unique points of choice if for each subterm of the form $\mathcal{I} + \mathcal{J}$ we have $\forall a \xrightarrow{o} b \in \text{transI}(\mathcal{I}), \forall c \xrightarrow{o'} d \in \text{transI}(\mathcal{J}), \{a, b\} \cap \{c, d\} \neq \emptyset$. Furthermore $\text{roles}(\mathcal{I}) = \text{roles}(\mathcal{J})$.*

The two conditions above are enough when each operation occurs just once in the IOC. However the same operation can be used more than once, provided that special care is taken to ensure that different occurrences do not interfere. This is formalized below, requiring a causality relationship between interactions using the same operation.

For defining the causality relation we need to index interactions inside IOC, and we use natural numbers to this end. Indexes are preserved by the projection, i.e. the input and the output obtained by projecting interaction i have both index i. We call a POC input and a POC output with the same index matching events. We denote with $\overline{e}$ the event matching event e. An event is unmatched if it has no matching events. An annotated IOC (resp. POC) is an IOC (resp. POC) with indexes.

Definition 4.4 (Synchronous causality relation). *Let us consider an annotated IOC $\mathcal{I}$. A synchronous causality relation $\leq_s$ is a partial order among events in the projection $\mathcal{S}$ of $\mathcal{I}$. We define $\leq_s$ as the minimum partial order satisfying:*

sequentiality: *for each $\mathcal{I};\mathcal{I}'$, if i is an interaction in $\mathcal{I}$, j is an interaction in $\mathcal{I}'$, and e_i and e_j are events in the same role then $e_i \leq_s e_j$;*

synchronization: *for each i, j if $e_i \leq_s e_j$ then $\overline{e}_i \leq_s e_j$.*

Definition 4.5 (Synchronous causality-safety). *An IOC is synchronous causality-safe iff for each pair of interactions i and j using the same operation, either $s_i \leq_s r_j \wedge r_i \leq_s s_j$ or $s_j \leq_s r_i \wedge r_j \leq_s s_i$.*

To understand the need for causality-safety consider the following IOC: $a \xrightarrow{o} b \parallel c \xrightarrow{o} d$. Here the two interactions exploit the same operation o, but there are no causal dependencies between the events corresponding to the two interactions, i.e., the IOC is not causality-safe. In fact the projection has the transition $(\overline{o})_a \parallel (o)_b \parallel (\overline{o})_c \parallel (o)_d \xrightarrow{a \xrightarrow{o} d} (\mathbf{1})_a \parallel (o)_b \parallel (\overline{o})_c \parallel (\mathbf{1})_d$ which is not allowed by the IOC.

Annotated IOCs and POCs are used also in the proof of our main theorem to deal with nondeterministic choice: when a choice is performed in the POC, some garbage is kept in form of events whose matching events has been discarded. Consider, e.g., the IOC transition $(a \xrightarrow{o} b; b \xrightarrow{o'} c) + (b \xrightarrow{o''} c; c \xrightarrow{o'''} a) \xrightarrow{a \xrightarrow{o} b} (b \xrightarrow{o'} c)$. The corresponding POC transition is: $(\overline{o}; \mathbf{1} + \mathbf{1}; o''')_a \parallel (o; \overline{o'} + \overline{o''}; \mathbf{1})_b \parallel (\mathbf{1}; o' + o''; \overline{o'''})_c \xrightarrow{a \xrightarrow{o} b} (\mathbf{1}; \mathbf{1})_a \parallel (\mathbf{1}; \overline{o'})_b \parallel (\mathbf{1}; o' + o''; \overline{o'''})_c$. In the result events o'' and $\overline{o'''}$ are unmatched, thus can never be executed and can be discarded. We define the function $\mathrm{rem}(\bullet)$ below to this end.

Definition 4.6. *Let $\mathcal{S}$ be an annotated POC. We denote with $\mathrm{rem}(\mathcal{S})$ the POC obtained from $\mathcal{S}$ by repeating the following pruning operations while possible:*

- *replace an unmatched event e in $\mathcal{S}$ with $\mathbf{0}$;*
- *replace each subterm $\mathbf{0}; P$ by $\mathbf{0}$, each subterm $\mathbf{0}+P$ by P and each subterm $\mathbf{0}|P$ by P.*

POCs obtained from annotated connected IOCs enjoy particular properties.

Definition 4.7 (Synchronous well-annotated POC). *A POC $\mathcal{S}$ is synchronous well-annotated for a causality relation $\leq_s$ iff for each index i there are at most two events with index i and in this case they are matching events. Furthermore, for each pair of events e_1 and e_2 on the same operation o with different indexes either $e_1 \leq_s e_2$ or $e_2 \leq_s e_1$. Finally, if $e_1 \leq_s e_2$ then e_2 can become enabled only after e_1 has been executed.*

Lemma 4.1. *Let $\mathcal{S}$ be a synchronous well-annotated POC for $\leq_s$. We have $\mathcal{S} \xrightarrow{a \xrightarrow{o} b}_s \mathcal{S}'$ iff $\mathrm{rem}(\mathcal{S}) \xrightarrow{a \xrightarrow{o} b}_s \mathrm{rem}(\mathcal{S}')$.*

Proof. By induction on the number of pruning operations in $\mathrm{rem}(\mathcal{S})$. The base case is trivial. Let us consider the inductive case. If the last pruning operation has been applied the thesis follows from the definition of the operational semantics. If the first one has been applied, we have to prove that e cannot interact. The proof is by contradiction. Suppose e interacts with e_1. They must be on the same operation. Since e is unmatched they can not have the same index. Thus since $\mathcal{S}$ is well-annotated there should be a causal dependency between e and e_1. Thus at most one of them can be enabled. This provides the contradiction. ☐

Lemma 4.2. *Let $\mathcal{I}$ be a causality-safe IOC. Then $\mathrm{proj}(\mathcal{I})$ is a well-annotated POC w.r.t. $\leq_s$.*

Proof. The proof is by structural induction on $\mathcal{I}$. The only condition difficult to prove is that if $e_1 \leq_s e_2$ then e_2 can become enabled only after e_1 has been executed.

We will prove by contradiction that at each step only minimal events can be enabled. The thesis will follow trivially. Suppose e_i is enabled but not minimal, i.e. there is e_j such that $e_j \leq_s e_i$. If there is more than one e_j consider the one such that the length of the derivation of $e_j \leq_s e_i$ is minimal. This should have length one, and this should result from an application of the rule on sequential composition. The thesis follows by definition of projection. ☐

Lemma 4.3. *If $\mathcal{S}$ is a well-annotated POC and $\mathcal{S} \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'$ then $\mathcal{S}'$ is a well-annotated POC.*

Lemma 4.4. *If $\mathcal{I} \xrightarrow{\surd}$ then for each role $r \in \mathrm{roles}(\mathcal{I})$ $\mathrm{proj}(\mathcal{I}, r) \xrightarrow{\surd}$.*

Lemma 4.5. *Let $\mathcal{I}$ be a synchronous connected IOC and $a \xrightarrow{o} b$ be an interaction in $\mathcal{I}$ with index i. If $\overline{o}$ and o have both index i and are both enabled in $\mathrm{proj}(\mathcal{I})$ then $a \xrightarrow{o} b \in \mathrm{transI}(\mathcal{I})$.*

Proof. By structural induction on $\mathcal{I}$. The cases for **1**, **0** and interactions are trivial. For parallel composition and choice just consider that since the two events have the same index then they are from the same component, and the thesis follows by inductive hypothesis. Let us consider sequential composition. Suppose $\mathcal{I} = \mathcal{I}';\mathcal{I}''$. If $a \xrightarrow{o} b \in \mathcal{I}'$ the thesis follows by inductive hypothesis. Otherwise by inductive hypothesis $a \xrightarrow{o} b \in \mathrm{transI}(\mathcal{I}'')$. Thus from synchronous connectedness for sequence there exists $c \xrightarrow{o'} d \in \mathrm{transF}(\mathcal{I}')$ with $\{a, b\} \cap \{c, d\} \neq \emptyset$. If the corresponding event is not part of a choice then either o or $\overline{o}$ is not enabled, and we get an absurd. If it is part of a choice, the same role should occur in all the other branches, and we have the absurd again. □

Remember that an IOC $\mathcal{I}$ is synchronous connected if it is synchronous connected for sequence, has synchronous unique points of choice and is synchronous causality-safe. We can now prove Theorem 4.1.

Proof of Theorem 4.1. We will show that the relation

$$R = \{(\mathcal{I}, \mathcal{S}) | \, \mathrm{rem}(\mathcal{S}) = \mathrm{proj}(\mathcal{I})\}$$

where $\mathcal{I}$ is synchronous connected and $\mathcal{S}$ is synchronous well-annotated is a bisimulation. Thanks to Lemma 4.2 all $\mathrm{proj}(\mathcal{I})$ are well-annotated. Thanks to Lemma 4.1 and Lemma 4.3 it is enough to consider the case $\mathcal{S} = \mathrm{proj}(\mathcal{I})$. The proof is by structural induction on the IOC $\mathcal{I}$. All the subterms of a synchronous connected IOC are synchronous connected, thus the induction can be performed.

Case 1, 0, $a \xrightarrow{o} b$**:** trivial;

Case $\mathcal{I};\mathcal{I}'$**:** from the definition of the projection function $\mathcal{S} = \|_r \ \mathrm{proj}(\mathcal{I}, r); \mathrm{proj}(\mathcal{I}', r)$. Suppose that $\mathcal{I};\mathcal{I}' \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$. There are two possibilities: either $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'''$ and $\mathcal{I}'' = \mathcal{I}''';\mathcal{I}'$ or $\mathcal{I} \xrightarrow{\surd}$ and $\mathcal{I}' \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$. In the first case by inductive hypothesis $\|_r \ \mathrm{proj}(\mathcal{I}, r) \xrightarrow{a \xrightarrow{o} b} \|_r \ \mathrm{proj}(\mathcal{I}''', r)$, thus $\|_r \ \mathrm{proj}(\mathcal{I}, r); \mathrm{proj}(\mathcal{I}', r) \xrightarrow{a \xrightarrow{o} b} \|_r \ \mathrm{proj}(\mathcal{I}''', r); \mathrm{proj}(\mathcal{I}', r)$ and the thesis follows.

If $\mathcal{I} \xrightarrow{\surd}$ and $\mathcal{I}' \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$ then by inductive hypothesis $\mathrm{proj}(\mathcal{I}') \xrightarrow{a \xrightarrow{o} b} \mathrm{proj}(\mathcal{I}'')$. The thesis follows since thanks to Lemma 4.4 also $\mathrm{proj}(\mathcal{I};\mathcal{I}') \xrightarrow{a \xrightarrow{o} b} \mathrm{proj}(\mathcal{I}'')$.

Let us consider the other condition. Suppose $\mathcal{S} = \|_r \ \mathrm{proj}(\mathcal{I}, r); \mathrm{proj}(\mathcal{I}', r) \xrightarrow{a \xrightarrow{o} b} \|_r \ \mathcal{S}'_r$. Thus $\mathrm{proj}(\mathcal{I};\mathcal{I}', a) \xrightarrow{\langle o \rangle} \mathcal{S}_a$ and $\mathrm{proj}(\mathcal{I};\mathcal{I}', b) \xrightarrow{o} \mathcal{S}_b$. The two events should have the same index thanks to Lemma 4.2 and to the definition of well-annotated POC (otherwise they could not be both enabled). Thus they are either both from $\mathcal{I}$ or both from $\mathcal{I}'$.

In the first case we have also $\|_r \ \mathrm{proj}(\mathcal{I}, r) \xrightarrow{a \xrightarrow{o} b} \|_r \ \mathcal{S}''_r$ with $\mathcal{S}'_r = \mathcal{S}''_r; \mathrm{proj}(\mathcal{I}', r)$. Thus by inductive hypothesis $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$ and $\|_r \ \mathcal{S}''_r$ is the projection of $\mathcal{I}''$. Also $\mathcal{I};\mathcal{I}' \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'';\mathcal{I}'$. The thesis follows. In the second case thanks to Lemma 4.5 $a \xrightarrow{o} b \in \mathrm{transI}(\mathcal{I};\mathcal{I}')$. Thus $\mathcal{I} \xrightarrow{\surd}$ and $\mathcal{I}' \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$. Thanks to Lemma 4.4 then $\mathrm{proj}(\mathcal{I}', a) \xrightarrow{\langle o \rangle} \mathcal{S}_a$, $\mathrm{proj}(\mathcal{I}', b) \xrightarrow{o} \mathcal{S}_b$ and $\mathrm{proj}(\mathcal{I}') \xrightarrow{a \xrightarrow{o} b} \|_r \ \mathcal{S}'_r$. The thesis follows by inductive hypothesis.

Case $\mathcal{I} \parallel \mathcal{I}'$**:** similar to the previous one.

Case $\mathcal{I} + \mathcal{I}'$**:** from the definition of the projection function $\mathcal{S} = \|_r \ \mathrm{proj}(\mathcal{I}, r) + \mathrm{proj}(\mathcal{I}', r)$. If $\mathcal{I} + \mathcal{I}'$ can perform an interaction, i.e. $\mathcal{I} + \mathcal{I}' \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$, then one of its two components can perform the same interaction. Let it be $\mathcal{I}$. Thus $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$. By inductive hypothesis $\|_r \ \mathrm{proj}(\mathcal{I}, r) \xrightarrow{a \xrightarrow{o} b} \|_r \ \mathrm{proj}(\mathcal{I}'', r)$. Thus $\|_r \ \mathrm{proj}(\mathcal{I}, r) + \mathrm{proj}(\mathcal{I}', r) \xrightarrow{a \xrightarrow{o} b} \|_r \ \mathcal{S}''_r$. We have to show that $\mathrm{rem}(\|_r \ \mathcal{S}''_r) = \|_r \ \mathrm{proj}(\mathcal{I}'', r)$. For roles a and b this is trivial. For other roles, if the interaction is initial then it can be discarded by $\mathrm{rem}(\bullet)$. In fact, because of the existence of unique points of choice one of its events is at a or at b, thus the matching event becomes unmatched and can be discarded (first pruning operation). We prove by induction on the structure of the term that if the initial transitions of a term can be discarded, then all its transitions can be discarded. The only difficult case is sequential composition. Let $\mathcal{J};\mathcal{J}'$ be the term. For interactions in $\mathcal{J}$ the thesis follows by inductive hypothesis. It is enough to prove that the initial interactions in $\mathcal{J}'$ can be discarded. Let $a \xrightarrow{o} b$ be such an interaction. Because of synchronous connectedness for sequence then a or b occur also in an interaction of $\mathcal{J}$, which is discarded, i.e. replaced by $\mathbf{0}$. Thus using the second pruning operation an event in the projection of $a \xrightarrow{o} b$ can be discarded, the other becomes unmatched and can be discarded too.

For the other direction, we have an input and an output on the same operation o enabled. Suppose they are both in $\text{proj}(\mathcal{I})$. Then $\text{proj}(\mathcal{I})$ has the same transition, i.e. $\text{proj}(\mathcal{I}) \xrightarrow{a \xrightarrow{o} b} \mathcal{S}''$, and by inductive hypothesis $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$ and thus $\mathcal{I} + \mathcal{I}' \xrightarrow{a \xrightarrow{o} b} \mathcal{I}''$. Also $\text{proj}(\mathcal{I} + \mathcal{I}') \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'''$. We have to show that $\text{rem}(\mathcal{S}''') = \text{proj}(\mathcal{I}'')$. The technique is the same as for the other direction. The thesis follows.

It is not possible that the input and output events are one in $\mathcal{I}$ and the other in $\mathcal{I}'$ since otherwise because of synchronous causality-safety they could not be both enabled.

□

5 Asynchronous conformances

In this section we discuss the different possibilities of conformance and connectedness that arise when the asynchronous semantics for POC is used. In fact, while in the IOC an interaction is an atomic event, in the POC for each interaction two events are performed: the sending and the receiving of the corresponding message. Thus different conformance relations are possible, depending on whether the IOC is used to specify the ordering of sendings, of receivings, or both the orderings. These correspond respectively to the sender, receiver and sender-receiver semantics. We also consider the disjoint semantics, which considers also the ordering of sendings and receivings mixed.

Definition 5.1 (Asynchronous unique point of choice). *An IOC $\mathcal{I}$ has asynchronous unique points of choice if for each subterm of the form $\mathcal{I} + \mathcal{J}$ we have $\forall a \xrightarrow{o} b \in \text{transI}(\mathcal{I}), \forall c \xrightarrow{o'} d \in \text{transI}(\mathcal{J}).a = c$. Furthermore $\text{roles}(\mathcal{I}) = \text{roles}(\mathcal{J})$.*

In order to define causality and well-annotated POCs, in addition to inputs and outputs events as in the synchronous case, we have to consider also messages $\langle o \rangle$. Messages are considered output events, they inherit the index of the output that generates them, and are matched with inputs with the same index.

Definition 5.2 (Asynchronous causality relation). *Let us consider an annotated IOC $\mathcal{I}$. An asynchronous causality relation $\leq_a$ is a partial order among events in the projection $\mathcal{S}$ of $\mathcal{I}$. We define $\leq_a$ as the minimum partial order satisfying:*

sequentiality: *for each $\mathcal{I};\mathcal{I}'$, if i is an interaction in $\mathcal{I}$, j is an interaction in $\mathcal{I}'$, and r_i and e_j are respectively a receive and a generic event in the same role then $r_i \leq_a e_j$;*

synchronization: *for each i, j if $r_i \leq_a e_j$ then $s_i \leq_a e_j$ (here s_i can be an output or a message).*

Notice that here outputs can not enforce sequentiality, since they can be executed asynchronously. The definition of asynchronous causality-safety is equal to the synchronous one (see Definition 4.5), but exploiting the asynchronous causality relationship $\leq_a$. The definition of the function $\text{rem}(\bullet)$ is unchanged too.

Definition 5.3 (Asynchronous well-annotated POC). *A POC $\mathcal{S}$ is asynchronous well-annotated for a causality relation $\leq_a$ iff it is synchronous well-annotated and each unmatched event is an input.*

Lemma 5.1. *Let $\mathcal{S}$ be an asynchronous well-annotated POC for $\leq_a$. We have $\mathcal{S} \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'$ (resp. $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'$) iff $\text{rem}(\mathcal{S}) \xrightarrow{a \xrightarrow{o} b} \text{rem}(\mathcal{S}')$ (resp. $\text{rem}(\mathcal{S}) \xrightarrow{\overline{o}:a} \text{rem}(\mathcal{S}')$).*

Lemma 5.2. *Let $\mathcal{I}$ be an asynchronous causality-safe IOC. Then $\text{proj}(\mathcal{I})$ is an asynchronous well-annotated POC w.r.t. $\leq_a$.*

Lemma 5.3. *If $\mathcal{S}$ is an asynchronous well-annotated POC and $\mathcal{S} \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'$ or $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'$ then $\mathcal{S}'$ is an asynchronous well-annotated POC.*

5.1 Sender conformance

In the sender case we use IOC to determine when messages are sent, disregarding when they are received.

The corresponding notion of bisimilarity is weak w.r.t. inputs. As a notation we will write $\Rightarrow_i$ for $\xrightarrow{a_1 \xrightarrow{o_1} b_1} \cdots \xrightarrow{a_n \xrightarrow{o_n} b_n}$ (zero or more transitions).

Definition 5.4 (Sender bisimilarity). *A sender bisimulation is a relation R between IOCs and POCs such that if $(\mathcal{I}, \mathcal{S}) \in R$ then:*

- *if $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ then $\mathcal{S} \Rightarrow_i \xrightarrow{\overline{o}:a} \mathcal{S}'$ and $(\mathcal{I}', \mathcal{S}') \in R$;*
- *if $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'$ then $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ and $(\mathcal{I}', \mathcal{S}') \in R$;*

- *if* $\mathcal{S} \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'$ *then* $(\mathcal{I}, \mathcal{S}') \in R$.

Sender bisimilarity $\sim_n$ *is the largest sender bisimulation.*

We will develop the tools to prove:

Theorem 5.1 (Sender conformance). *Let* $\mathcal{I}$ *be an IOC and* $\mathcal{S} = \text{proj}(\mathcal{I})$ *be its projection. If* $\mathcal{I}$ *satisfies the connectedness conditions for the sender semantics then* $\mathcal{I} \sim_n \mathcal{S}$.

We start by showing the connectedness conditions.

Definition 5.5 (Sender connectedness for sequence). *An IOC* $\mathcal{I}$ *is sender connected for sequence if for each subterm of the form* $\mathcal{I}; \mathcal{J}$ *we have* $\forall a \xrightarrow{o} b \in \text{transF}(\mathcal{I}), \forall c \xrightarrow{o'} d \in \text{transI}(\mathcal{J}). a = c \vee b = c$.

Lemma 5.4. *Let* $\mathcal{I}$ *be a sender connected IOC and* $a \xrightarrow{o} b$ *be interaction in* $\mathcal{I}$ *with index* i. *If* $\overline{o}$ *has index* i *and is enabled in* $\text{proj}(\mathcal{I})$ *then* $a \xrightarrow{o} b \in \text{transI}(\mathcal{I})$.

Lemma 5.5. *Let* R' *be a relation between IOCs and POCs. Let* $R = \{(\mathcal{I}, \mathcal{S}') | \mathcal{S}' \Rightarrow_i \mathcal{S} \wedge (\mathcal{I}, \mathcal{S}) \in R'\}$. *Suppose that in each* $\mathcal{S}'$ *there is no mixed choice and in input choices at most one branch at the time is enabled. If* R' *is such that if* $(\mathcal{I}, \mathcal{S}) \in R'$ *then:*

- *if* $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ *then* $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'$ *and* $(\mathcal{I}', \mathcal{S}') \in R$;
- *if* $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'$ *then* $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ *and* $(\mathcal{I}', \mathcal{S}') \in R$;
- $\mathcal{S}$ *has no input transitions.*

then R *is a sender bisimilarity.*

Proof. By coinduction, exploiting the fact that given the conditions on choices POC computations enjoy nice properties allowing to reorder transitions. □

We can now prove Theorem 5.1.

Proof of Theorem 5.1. The proof shows that the relation

$$R = \{(\mathcal{I}, \mathcal{S}) | \mathcal{S} \Rightarrow_i \mathcal{S}' \wedge \text{rem}(\mathcal{S}') = \text{proj}(\mathcal{I})\}$$

where $\mathcal{I}$ is a connected IOC and $\mathcal{S}$ is a well-annotated POC is a sender bisimulation. Since the conditions of Lemma 5.5 on choice and input transitions are satisfied (the first thanks to the existence of unique points of choice, the second by definition of the asynchronous semantics) then it is enough to prove that

$$R' = \{(\mathcal{I}, \mathcal{S}') | \, \text{rem}_s(\mathcal{S}') = \text{proj}(\mathcal{I})\}$$

satisfies the conditions of Lemma 5.5. Thanks to Lemma 5.1, Lemma 5.2 and Lemma 5.3 one can just consider the case $\mathcal{S}' = \text{proj}(\mathcal{I})$.

The proof is by structural induction on $\mathcal{I}$, and the cases are similar to the ones of Theorem 4.1. The case for sequential composition exploits Lemma 5.4, while the case for choice exploits the function $\text{rem}(\bullet)$. □

5.2 Receiver conformance

In the receiver case we use IOC to determine when messages are received, disregarding when they are sent.

The corresponding notion of bisimilarity is weak w.r.t. outputs. As a notation we will write $\Rightarrow_o$ for $\xrightarrow{\overline{o_1}:a_1} \ldots \xrightarrow{\overline{o_2}:a_2}$ (zero or more transitions).

Definition 5.6 (Receiver bisimilarity). *A receiver bisimulation is a relation* R *between IOCs and POCs such that if* $(\mathcal{I}, \mathcal{S}) \in R$ *then:*

- *if* $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ *then* $\mathcal{S} \Rightarrow_o \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'$ *and* $(\mathcal{I}', \mathcal{S}') \in R$;
- *if* $\mathcal{S} \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'$ *then* $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ *and* $(\mathcal{I}', \mathcal{S}') \in R$;
- *if* $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'$ *then* $(\mathcal{I}, \mathcal{S}') \in R$.

Receiver bisimilarity $\sim_r$ *is the largest receiver bisimulation.*

We will develop the tools to prove:

Theorem 5.2 (Receiver conformance). *Let* $\mathcal{I}$ *be an IOC and* $\mathcal{S} = \text{proj}(\mathcal{I})$ *be its projection. If* $\mathcal{I}$ *satisfies the connectedness conditions for the receiver semantics then* $\mathcal{I} \sim_r \mathcal{S}$.

We start by showing the connectedness conditions.

Definition 5.7 (Receiver connectedness for sequence). *An IOC* $\mathcal{I}$ *is receiver connected for sequence if for each subterm of the form* $\mathcal{I}; \mathcal{J}$ *we have* $\forall a \xrightarrow{o} b \in \text{transF}(\mathcal{I}), \forall c \xrightarrow{o'} d \in \text{transI}(\mathcal{J}). b = c \vee b = d$.

Lemma 5.6. *Let* $\mathcal{I}$ *be a receiver connected IOC and* $a \xrightarrow{o} b$ *be an interaction in* $\mathcal{I}$ *with index* i. *If there exists* $\mathcal{S}$ *such that* $\text{proj}(\mathcal{I}) \Rightarrow_o \mathcal{S}$ *and* o *and* $\langle o \rangle$ *have both index* i *and are both enabled in* $\mathcal{S}$ *then* $a \xrightarrow{o} b \in \text{transI}(\mathcal{I})$.

Lemma 5.7. *Let* $\mathcal{I}$ *be a receiver connected IOC. If* $\text{proj}(\mathcal{I}) \Rightarrow_o \mathcal{S}' \xrightarrow{a \xrightarrow{o} b} \mathcal{S}''$ *and* $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ *then* $\text{proj}(\mathcal{I}') \Rightarrow_o \text{rem}(\mathcal{S}'')$.

Proof. The proof is by structural induction on $\mathcal{I}$. One applies the inductive hypothesis to the component that performs the transition $\xrightarrow{a \xrightarrow{o} b}$, and notices that the other component can perform just some outputs that commute with the main transition. □

We can now prove Theorem 5.2.

Proof of Theorem 5.2. The proof shows that the relation

$$R = \{(\mathcal{I}, \mathcal{S}') | \operatorname{proj}(\mathcal{I}) \Rightarrow_o \mathcal{S} \wedge \operatorname{rem}_s(\mathcal{S}') = \mathcal{S}\}$$

where $\mathcal{I}$ is a connected IOC and $\mathcal{S}'$ is a well-annotated POC is a receiver bisimulation. Thanks to Lemma 5.1, Lemma 5.2 and Lemma 5.3 it is enough to consider the case $\mathcal{S}' = \mathcal{S}$. The proof is by structural induction on $\mathcal{I}$. The cases are similar to the ones of Theorem 4.1, and exploit Lemma 5.7. □

5.3 Disjoint conformance

In the disjoint case we use IOC to determine both when messages are sent and when they are received.

The corresponding notion of bisimilarity is strong and considers both inputs and outputs.

Definition 5.8 (Disjoint bisimilarity). *A disjoint bisimulation is a relation R between IOCs and POCs such that if $(\mathcal{I}, \mathcal{S}) \in R$ then:*

- *if $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ then $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'' \xrightarrow{a \xrightarrow{o} b} \mathcal{S}'$ and $(\mathcal{I}', \mathcal{S}') \in R$; furthermore if $\mathcal{S}'' \xrightarrow{\gamma} \mathcal{S}'''$ then $\mathcal{S} \xrightarrow{\gamma} \xrightarrow{\overline{o}:a} \mathcal{S}'''$;*
- *if $\mathcal{S} \xrightarrow{\overline{o}:a} \mathcal{S}'$ then $\mathcal{S}' \xrightarrow{a \xrightarrow{o} b} \mathcal{S}''$ and $\mathcal{I} \xrightarrow{a \xrightarrow{o} b} \mathcal{I}'$ and $(\mathcal{I}', \mathcal{S}'') \in R$.*

Disjoint bisimilarity $\sim_d$ is the largest disjoint bisimulation.

The diamond property condition in the first item is needed to ensure that the output does not make any new transition enabled, but for the corresponding input. Without this condition e.g. the IOC $a \xrightarrow{o} b; a \xrightarrow{o'} c$ would be bisimilar to its projection $(\overline{o}; \overline{o})_a \parallel (o; \mathbf{1})_b \parallel (\mathbf{1}; o')_c$, but the projection can perform the output on o' before the input of o, thus violating the disjointness property we want to guarantee. This is more easily formalizable with trace equivalence, saying that the traces of a sequential composition are compositions of traces of the two components.

We will develop the tools to prove:

Theorem 5.3 (Disjoint conformance). *Let $\mathcal{I}$ be IOC and $\mathcal{S} = \operatorname{proj}(\mathcal{I})$ be its projection. If $\mathcal{I}$ satisfies the connectedness conditions for the disjoint semantics then $\mathcal{I} \sim_d \mathcal{S}$.*

We start by showing the connectedness conditions.

Definition 5.9 (Disjoint connectedness for sequence). *An IOC $\mathcal{I}$ is disjoint connected for sequence if for each subterm of the form $\mathcal{I}; \mathcal{J}$ we have $\forall a \xrightarrow{o} b \in \operatorname{transF}(\mathcal{I}), \forall c \xrightarrow{o'} d \in \operatorname{transI}(\mathcal{J}). b = c$.*

Lemma 5.8. *Let $\mathcal{I}$ be a disjoint connected IOC and $a \xrightarrow{o} b$ be an interaction in $\mathcal{I}$ with index i. If $\overline{o}$ has index i and is enabled in $\operatorname{proj}(\mathcal{I})$ then also o (with index i) is enabled in $\operatorname{proj}(\mathcal{I})$ and $a \xrightarrow{o} b \in \operatorname{transI}(\mathcal{I})$.*

We can now prove Theorem 5.3.

Proof of Theorem 5.3. The proof shows that the relation

$$R = \{(\mathcal{I}, \mathcal{S}) | \operatorname{rem}(\mathcal{S}) = \operatorname{proj}(\mathcal{I})\}$$

where $\mathcal{I}$ is a connected IOC and $\mathcal{S}$ is a well-annotated POC is a disjoint bisimulation. Thanks to Lemma 5.1, Lemma 5.2 and Lemma 5.3 one can just consider the case $\mathcal{S}' = \operatorname{proj}(\mathcal{I})$. The proof is by structural induction on $\mathcal{I}$, and the cases are similar to the ones of Theorem 4.1. The case for choice exploits the function $\operatorname{rem}(\bullet)$, while the case for sequential composition exploits Lemma 5.4. Notice that the diamond property is guaranteed since no output execution can make new parts of sequential compositions enabled. □

6 Possible extensions

We discuss here some possible extensions to this work. They will be considered in future works.

Other operators. The theory developed in previous sections can be easily extended to deal with more operators, such as internal actions τ_a and guarded recursion. In both the cases the same operator should be added to both the IOC and the POC. The projection of τ_a on a should be τ_a, while its projection on the other roles should be $\mathbf{1}$. Note that we can not represent it as $a \xrightarrow{o} a$, since in the synchronous semantics the projection will not be executable. For recursion, projection is an homomorphism. Here the important thing is connectedness: every unfolding of a recursion is connected if its one-level unfolding is (see e.g. [HYC08]).

Data. Our input and output operations abstract message passing, without showing the actual values. All the results can be generalized to message passing, provided that corresponding choices are done both in IOCs and in POCs: for instance, a natural assumption is to have data localized at some roles, and to check that the sender has the necessary data available. Also, deterministic choice could be introduced instead of nondeterministic choice. In this case the unique point of choice should coincide with the role that evaluates the condition (and which owns the used variables), and a nondeterministic choice should be used in the projection on other roles.

Bisimilarity and refinements. One can define bisimulations or simulations on both IOCs and POCs. For instance, one could use strong bisimulation for both IOC and synchronous POC. Interestingly, these notions are compatible with synchronous conformance and projection: e.g., the projections of two bisimilar IOCs are bisimilar. Similar compatibility relations emerge when sender (resp. receiver) semantics are used, but here at the POC level one can use a bisimilarity weak w.r.t. inputs (resp. outputs). For the disjoint semantics strong bisimilarity is required instead. In this way one can e.g. check that a POC $\mathcal{S}$ partially implements an IOC $\mathcal{I}$, by checking that $\mathrm{proj}(\mathcal{I})$ simulates $\mathcal{S}$.

7 Related work

The problem of conformance between a POC and a IOC has been considered many times in the literature. [CHY07] and [HYC08] use a global calculus and an end-point calculus to describe IOC and POC respectively. Since the language is quite complex types are used as abstractions to check conformance. The language has prefix instead of sequential composition, and labeled choice in the session types style instead of nondeterministic choice. In [CHY07] a synchronous semantics is used, and the relation between IOC and POC corresponds to our synchronous bisimulation. The constraints imposed on IOCs are however stricter than ours, since for sequence they correspond to our disjoint connectedness. In [HYC08] instead the asynchronous case is considered. The semantics therein corresponds to our receiver semantics, but they preserve the order of messages from the same sender and on the same operation. Their conditions are again stricter than ours, since they do not allow the same role to occur in different parallel components, while we do, and they require projections of non initiator roles in choice to coincide in every branch, while we allow different projections.

In [BZ07] trace inclusion (with a synchronous semantics) is used to relate service contracts and the roles of a choreography. This is similar to our synchronous conformance, but in [BZ07] the participants may provide additional functionalities, provided that they are not used inside the choreography. Also, connectedness is defined only from a behavioral point of view, but no syntactic criterion ensuring this is presented.

In [BGG+05] and [BGG+06] different bisimilarities are used to characterize conformance of a POC w.r.t. a IOC. These bisimilarities generalize respectively our synchronous and receiver conformances, allowing a role in a IOC to be implemented by many processes in a POC. However, the problems of automatically generating the processes via projection and of deciding whether a IOC can be implemented are not considered.

References

[BGG+05] N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro. Choreography and orchestration: A synergic approach for system design. In *Proc. of ICSOC'05*, volume 3826 of *LNCS*, pages 228–240. Springer, 2005.

[BGG+06] N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro. Choreography and orchestration conformance for system design. In *Proc. of Coordination'06*, volume 4038 of *LNCS*, pages 63–81. Springer, 2006.

[BZ07] M. Bravetti and G. Zavattaro. Towards a unifying theory for choreography conformance and contract compliance. In *Proc. of SC'07*, volume 4829 of *LNCS*, pages 34–50. Springer, 2007.

[CHY07] M. Carbone, K. Honda, and N. Yoshida. Structured communication-centred programming for web services. In *Proc. of ESOP'07*, volume 4421 of *LNCS*, pages 2–17. Springer, 2007.

[DKLW07] G. Decker, O. Kopp, F. Leymann, and M. Weske. Bpel4chor: Extending bpel for modeling choreographies. In *Proc. of ICWS'07*, pages 296–303. IEEE Computer Society Press, 2007.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[HYC08] K. Honda, N. Yoshida, and M. Carbone. Multiparty asynchronous session types. In *Proc. of POPL'08*, pages 273–284. ACM Press, 2008.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[OAS] OASIS. *Web Services Business Process Execution Language Version 2.0 OASIS Standard*. `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf`.

[Wor] World Wide Web Consortium. *Web Services Choreography Description Language Version 1.0*. `http://www.w3.org/TR/ws-cdl-10/`.