ELEKTROTEHNIŠKI VESTNIK 88(4): 190–196, 2021
ORIGINAL SCIENTIFIC PAPER

Realizable Choreographies for Systems of Components Communicating via Rendezvous, Mailboxes and Letter Queues

Monika Kapus-Kolar

Jožef Stefan Institute, Department of Communication Systems, Jamova 39, SI-1111 Ljubljana, Slovenia
E-mail: monika.kapus-kolar@ijs.si

Abstract. A recently proposed class of realizable compositionally specified choreographies for systems of components communicating over first-in-first-out channels with exactly one channel for every source-sink pair is generalized to systems with an arbitrary number of first-in-first-out channels, mailboxes and rendezvous channels, with no restrictions on who is allowed to exchange letters on individual channels.

Keywords: choreography, semantics, realizability, pomset, communicating state machine

Izvedljive koreografije za sisteme komponent, ki komunicirajo prek srečanj, poštnih nabiralnikov in pisemskih vrst

Pred kratkim predlagan razred izvedljivih kompozicijsko specificiranih koreografij za sisteme komponent, ki komunicirajo prek pisemskih vrst, z natanko eno vrsto za vsak par pošiljatelj-prejemnik, je posplošen na sisteme s poljubnim številom pisemskih vrst, poštnih nabiralnikov in kanalov za izmenjavo pisem prek srečanj, brez omejitev glede tega, kdo sme izmenjevati pisma prek posameznih komunikacijskih kanalov.

1 INTRODUCTION

When designing a distributed application supposed to run on a given distributed system, a possible way to proceed is to first conceive a choreography, i.e. a model of interactions among the system components from the global point of view. Ideally, the choreography is realizable, i.e. component processes for its correct implementation can be obtained simply by its projection.

The paper generalizes our recent extension [1] of the work of Tuosto and Guanciale [2] on compositional construction of realizable choreographies for systems in which (1) every communication channel is between a pair of two different components, (2) from any component to any other component, there is exactly one communication channel, and (3) every channel is an initially empty, infinite-capacity buffer in which messages are queued in the order of arrival and exactly the first in the queue is available for reception. Newly we allow (1) any number of channels, with no restrictions on who is allowed to use them for the exchange of letters (i.e. triplets consisting of the identifier of the sender, the identifier of the recipient and the message carried), (2) besides channels with an infinite-capacity buffer also channels whose buffer capacity is zero, so that any letter sent on it must be received in the same event - a rendezvous of its sender and its recipient, and (3) besides infinite-capacity communication buffers with the first-in-first-out (FIFO) policy also such (we call them mailboxes) in which letters are not queued and can be retrieved at any time.

In line with the argumentation of Tuosto and Guanciale [2] that in the formal study of choreographies, it pays to go abstract, we follow the abstract view of [1] that a choreography is a set of partially ordered multisets (shortly pomsets) of interactions. An interaction is an individual exchange of a certain letter on a certain buffer and consists of two actions: a transmission of an instance of the letter into the buffer and its reception from the buffer at the same or some latter point. In case of a zero-capacity buffer, the two actions by definition occur simultaneously, so that the interaction can be regarded as a compound action.

With the adopted abstract approach, we define the semantics, projection and well-formedness of choreographies in a way independent of their concrete syntax. Like [1], [2] and also [3], in which Tuosto and Guanciale discuss the realizability of choreographies with mailbox communication only, we define the semantics of a given choreography as a set of action pomsets, and component processes obtained by projection as communicating state machines (CSMs) [4]. Like [2] and [1], but unlike [3], we simplify the discussion by banning choreographies with auto-concurrency (i.e. with multiple concurrent instances per action) and by assuming that individual CSMs are allowed to permanently stop exactly when in a state with no further actions defined. We prove that all well-formed choreographies are realizable.

Received 6 May 2021
Accepted 24 August 2021

The choreography composition operators introduced as syntax sugar are the same as in [1], namely the operators of choice, parallel composition and sequential composition. For each of them we adapt to the generalized setting the operand constraints which [1] proves sufficient for the well-formedness of the choreography resulting from the composition.

As a final contribution, we compare our definition of well-formed choreographies with the conceptually similar definitions recently proposed in [5]–[7]. Contrary to what is allegedly proved in the three papers, we demonstrate that (and explain why) none of the similar definitions secures the realizability of every well-formed choreography.

2 Basic Concepts and Notations

2.1 (Inter)actions and Their Instances

We assume that choreographies are designed for a system with component set $\mathcal{C}$. Elements of $\mathcal{C}$ are ranged over by c.

Communication buffers, i.e. channels, are ranged over by b. For a given buffer b, $Fifo(b)$ denotes that it is a FIFO channel, $Mb(b)$ that it is a mailbox, and $Rv(b)$ that it is a rendezvous channel.

Messages are ranged over by m. A letter is denoted as (c, c', m) where c is its sender, c' its recipient and m its message. Letters are ranged over by l.

Interactions are ranged over by x. An interaction in which a given letter l is exchanged over a given buffer b is denoted as $b : l$, whereas its constituent transmission and reception are denoted as $b!l$ and $b?l$, respectively. In case of $Rv(b)$, the compound action representing a simultaneous execution of the two constituent actions of $b : l$ is denoted as $b!?l$.

Actions, i.e. transmissions, receptions and rendezvous regarded as compound actions, are ranged over by a, action sequences by α, and sets of action sequences by A. The participant set of a given action a of the form $b!(c, c', m)$, $b?(c, c', m)$ or $b!?(c, c', m)$ is denoted as $\mathrm{prt}(a)$ and defined as $\{c\}$, $\{c'\}$ or $\{c, c'\}$, respectively.

(Inter)action instances are alternatively called events. Events are ranged over by e, event sets by $\mathcal{E}$, and event sequences by ε.

For a given event e, $\lambda(e)$ denotes its label, i.e. the (inter)action of which it is an instance. For a given action instance sequence $\varepsilon = (e_i)_{i=1\ldots k}$, $\mathrm{asq}(\varepsilon)$ denotes the action sequence $(\lambda(e_i))_{i=1\ldots k}$.

Interaction instances and their sets are alternatively (to expose their nature) ranged over by g and $\mathcal{G}$, respectively. For a given interaction instance g, $e_g^!$ denotes the constituent transmission instance, and $e_g^?$ the constituent reception instance. For a given instance g of an interaction $b : l$ with $Rv(b)$, $e_g^{!?}$ denotes the action instance representing a simultaneous execution of $e_g^!$ and $e_g^?$. For a given instance g of an interaction $b : l$ with $\neg Rv(b)$ or $Rv(b)$, $\mathrm{ais}(g)$ denotes the action instance set defined as $\{e_g^!, e_g^?\}$ or $\{e_g^{!?}\}$, respectively.

2.2 Partially Ordered Sets of (Inter)action Instances

A binary relation on a given event set $\mathcal{E}$ is a subset of $\mathcal{E} \times \mathcal{E}$. If it is reflexive, anti-symmetric and transitive, it is called partial order. The transitive closure of a given binary relation R is denoted as $R^\star$.

A partially ordered set of events (shortly poset) is an event set $\mathcal{E}$ endowed with a partial order $\leq$ and denoted as $(\mathcal{E}, \leq)$. Posets are ranged over by p.

If for given posets $p = (\mathcal{E}, \leq)$ and $p' = (\mathcal{E}', \leq')$ there exist bijections $\phi : \mathcal{E} \to \mathcal{E}'$ and $\phi' : \leq \to \leq'$ with

(1) $\forall e \in \mathcal{E} : (\lambda(e) = \lambda(\phi(e)))$ and
(2) $\forall (e, e') \in \leq : (\phi'((e, e')) = (\phi(e), \phi(e')))$,

then p and p' are isomorphic.

For a given poset $p = (\mathcal{E}, \leq)$, $\mathrm{esq}(p)$ denotes the set of all event sequences $(e_i)_{i=1\ldots|\mathcal{E}|}$ with

(1) $\mathcal{E} = \{e_i\}_{i=1\ldots|\mathcal{E}|}$ and
(2) $\forall 1 \leq i < j \leq |\mathcal{E}| : (e_j \not\leq e_i)$.

For a given poset $p = (\mathcal{E}, \leq)$, $\mathrm{pf}(p)$ denotes the set of all its prefixes, i.e. the set of all posets $(\mathcal{E}', \leq')$ with

(1) $\mathcal{E}' \subseteq \mathcal{E}$,
(2) $\leq' = \leq \cap (\mathcal{E}' \times \mathcal{E}')$ and
(3) $\leq \cap ((\mathcal{E} \setminus \mathcal{E}') \times \mathcal{E}') = \emptyset$.

For given action instance poset p and action sequence α, $\mathrm{pf}(p, \alpha)$ denotes the set of all posets $p' \in \mathrm{pf}(p)$ whose $\mathrm{esq}(p')$ comprises an event sequence ε with $\mathrm{asq}(\varepsilon) = \alpha$, whereas $\mathrm{asq}(p)$ denotes the set of all action sequences α' with $\mathrm{pf}(p, \alpha') \neq \emptyset$.

2.3 Partially Ordered Multisets of (Inter)actions

A partially ordered multiset of (inter)actions (shortly pomset) is an isomorphism class of posets. The isomorphism class to which a given poset $p = (\mathcal{E}, \leq)$ belongs is denoted as $[p]$ or $[\mathcal{E}, \leq]$. Pomsets are ranged over by r, and their sets by $\mathcal{R}$.

A natural way to discuss properties of a given pomset is to discuss properties of a representative of the class. Likewise, a natural way to define a composition operator for pomsets is to do it in terms of selected representatives of individual operands, taking care that the representatives are non-intersecting (inter)action instance sets. When discussing or combining pomset sets, one would proceed analogously. We therefore define the following families of pomset and pomset set representatives:

(1) For given pomset r and (possibly omitted) natural i, $\mathrm{po}_i(r) = (\mathcal{E}_{r,i}, \leq_{r,i})$ is the poset selected as the default representative of the class r (for the natural i), with $\mathcal{E}_{r,i} \cap \mathcal{E}_{r',i'} = \emptyset$ for every pomset r' and natural i' with $(r, i) \neq (r', i')$.
(2) For given pomset set $\mathcal{R}$ and (possibly omitted) natural i, $\mathrm{pos}_i(\mathcal{R})$ denotes the poset set $\{\mathrm{po}_i(r) | r \in \mathcal{R}\}$.

For given action pomset r and action sequence α, $\mathrm{pf}(r,\alpha)$ denotes the set of all pomsets $[p]$ with $p \in \mathrm{pf}(\mathrm{po}(r),\alpha)$. For given action pomset set $\mathcal{R}$, $\mathrm{asq}(\mathcal{R})$ denotes the union of all action sequence sets $\mathrm{asq}(p)$ with $p \in \mathrm{pos}(\mathcal{R})$, whereas $\delta_{\mathcal{R}}$ denotes the function that for a given action sequence $\alpha \in \mathrm{asq}(\mathcal{R})$ returns the union of all pomset sets $\mathrm{pf}(r,\alpha)$ with $r \in \mathcal{R}$.

2.4 State Machines

An initialized, initially connected and deterministic state automaton whose individual steps represent individual actions (shortly state machine) is denoted as (A,δ) with A the set of all action sequences executable from the initial state, and δ the function that for any given action sequence in A returns the resulting state. For a given state machine $M=(A,\delta)$, $\mathrm{asq}(M)$ denotes A. For a given action pomset set $\mathcal{R}$, $\mathrm{sm}(\mathcal{R})$ denotes the state machine $(\mathrm{asq}(\mathcal{R}),\delta_{\mathcal{R}})$.

2.5 Projections

For a given action a and component $c \in \mathrm{prt}(a)$, $a\restriction_c$ denotes:

(1) if a is a $b!?(c,c',m)$ then the action $b!(c,c',m)$,
(2) if a is a $b!?(c',c,m)$ then the action $b?(c',c,m)$,
(3) otherwise the action a.

For given action instance set $\mathcal{E}$ and component c, $\mathcal{E}\restriction_c$ denotes the set of all events $e \in \mathcal{E}$ with $c \in \mathrm{prt}(\lambda(e))$. For given action instance poset $p=(\mathcal{E},\leq)$ and component c, $p\restriction_c$ denotes the poset $(\mathcal{E}\restriction_c, \leq \cap((\mathcal{E}\restriction_c)\times(\mathcal{E}\restriction_c)))$.

For given action pomset set $\mathcal{R}$ and component c, $\mathcal{R}\restriction_c$ denotes the set of all pomsets $[p\restriction_c]$ with $p \in \mathrm{pos}(\mathcal{R})$, $\mathrm{sm}_c(\mathcal{R})$ denotes the state machine obtained from $\mathrm{sm}(\mathcal{R}\restriction_c)$ by changing each of its actions a into $a\restriction_c$, and $\mathrm{asq}_c(\mathcal{R})$ denotes the action sequence set $\mathrm{asq}(\mathrm{sm}_c(\mathcal{R}))$. For a given action pomset set $\mathcal{R}$, $\lambda(\mathcal{R})$ denotes the set of all actions present in at least one sequence in the set $\bigcup_{c\in\mathcal{C}}\mathrm{asq}_c(\mathcal{R})$.

3 Well-Formed Choreographies

3.1 Choreographies and Their Normal Form

In our simple specification language, choreographies are defined as terms derived by the following grammar (parentheses not necessary for disambiguation can be omitted):

$G ::= \mathbf{0} \mid b:l \mid \mathcal{R} \mid (G_1|G_2) \mid (G_1;G_2) \mid (G_1+G_2)$

In the grammar, $\mathbf{0}$ denotes doing nothing, $b:l$ is assumed to be an interaction, $\mathcal{R}$ is assumed to be a non-empty set of interaction pomsets, '|' is the parallel composition operator, ';' is the sequential composition operator, and '+' is the choice operator.

To abstract away from the concrete syntax of choreographies, we define for them a normal form. The normal form of a given choreography G, denoted as $\langle\!\langle G\rangle\!\rangle$, is a non-empty set of interaction pomsets, with individual pomsets representing individual alternatives between which the system is supposed to choose when executing G. For our six choreography types, the normal form is defined as follows:

$\langle\!\langle \mathbf{0}\rangle\!\rangle = \{[\{\},\{\}]\}$

$\langle\!\langle b:l\rangle\!\rangle = \{[\{g\},\{(g,g)\}]\}$ with g being an instance of $b:l$.

The alternatives specified by a given interaction pomset set $\mathcal{R}$ are the pomsets. Accordingly, $\langle\!\langle\mathcal{R}\rangle\!\rangle$ is $\mathcal{R}$ itself.

The alternatives of a G_1+G_2 are the alternatives of G_1 and the alternatives of G_2. Accordingly, $\langle\!\langle G_1+G_2\rangle\!\rangle$ is the union of $\langle\!\langle G_1\rangle\!\rangle$ and $\langle\!\langle G_2\rangle\!\rangle$.

The alternatives of a $G_1|G_2$ are all those defined as a parallel composition of an alternative of G_1 and an alternative of G_2. Accordingly, $\langle\!\langle G_1|G_2\rangle\!\rangle$ is the set of all pomsets $[\mathcal{G}_1\cup\mathcal{G}_2, \leq_1\cup\leq_2]$ where $(\mathcal{G}_1,\leq_1)$ is a poset in $\mathrm{pos}_1(\langle\!\langle G_1\rangle\!\rangle)$ and $(\mathcal{G}_2,\leq_2)$ is a poset in $\mathrm{pos}_2(\langle\!\langle G_2\rangle\!\rangle)$.

The alternatives of a $G_1;G_2$ are all those defined as a sequential composition of an alternative of G_1 and an alternative of G_2. Accordingly, $\langle\!\langle G_1;G_2\rangle\!\rangle$ is the set of all pomsets $[\mathcal{G}_1\cup\mathcal{G}_2, (\leq_1\cup\leq_2\cup(\mathcal{G}_1\times\mathcal{G}_2))^\star]$ where $(\mathcal{G}_1,\leq_1)$ is a poset in $\mathrm{pos}_1(\langle\!\langle G_1\rangle\!\rangle)$ and $(\mathcal{G}_2,\leq_2)$ is a poset in $\mathrm{pos}_2(\langle\!\langle G_2\rangle\!\rangle)$.

3.2 Choreography Semantics

For given interactions $x_1 = b_1 : (c_1, c_1', m_1)$ and $x_2 = b_2 : (c_2, c_2', m_2)$, let $Ord_1(x_1,x_2)$ denote that $c_2 \in \{c_1, c_1'\}$. For given actions a_1 and a_2, let $Ord_2(a_1,a_2)$ denote that (a_1,a_2) is

(1) an $(a, b!(c,c',m))$ with $c \in \mathrm{prt}(a)$ or
(2) an $(a, b!?(c,c',m))$ with $c \in \mathrm{prt}(a)$ or
(3) a $(b?l_1, b?l_2)$ with $Fifo(b)$ or
(4) a $(b?l, b?l)$ with $Mb(b)$.

In the normal form $\langle\!\langle G\rangle\!\rangle$ of a given choreography G, each constituent alternative of G is represented by a pomset specifying the alternative very abstractly, in terms of interactions. To obtain the semantics of G, we refine every pomset r in $\langle\!\langle G\rangle\!\rangle$ into a pomset r' specifying the alternative of G less abstractly, in terms of actions. To obtain r', we take the interaction instance poset p that is the default representative of the isomorphism class r, refine it into the action instance poset $[\![p]\!]$ below defined as the semantics of p, and set r' to the isomorphism class to which $[\![p]\!]$ belongs. In other words, we define that the semantics of a given choreography G, denoted as $[\![G]\!]$, is the action pomset set $\{[[\![p]\!]] \mid p \in \mathrm{pos}(\langle\!\langle G\rangle\!\rangle)\}$.

In our semantics $[\![p]\!]$ of a given interaction instance poset p, each of the interaction instances is represented by its constituent action instances, whereas the extent to which constituents of interaction instances ordered in p are ordered in $[\![p]\!]$ is selected in line with the following assumptions which [1] implicitly makes for every specified pair (g_1,g_2) of ordered interaction instances, with $\lambda(g_i)$ a $b_i : (c_i, c_i', m_i)$ for $i \in \{1,2\}$:

(1) The ordering of g_1 and g_2 means just that $e^{!}_{g_2}$ (in case of $Rv(b_2)$ a part of $e^{!?}_{g_2}$) is supposed to be

delayed with respect to $e^{!}_{g_1}$ (in case of $Rv(b_1)$ a part of $e^{!?}_{g_1}$).
(2) The component responsible for the delaying is c_2.
(3) If the delaying can be implemented at c_2, which is in case of $Ord_1(\lambda(g_1), \lambda(g_2))$, it must be implemented. Otherwise, the ordering of g_1 and g_2 has been specified unintentionally and one is supposed to pretend that it has not been specified (at least not explicitly, for note that there is also ordering because of the transitivity of the ordering relation).

Formally, our semantics $[\![p]\!]$ of a given interaction instance poset $p = (\mathcal{G}, \leq)$ is an action instance poset $(\mathcal{E}, (\leq_1 \cup \leq_2 \cup \leq_3)^\star)$ where

$\mathcal{E} = \bigcup_{g \in \mathcal{G}} \mathrm{ais}(g)$
$\leq_1 = \{(e, e) | e \in \mathcal{E}\}$
$\leq_2 = \{(e^{!}_g, e^{?}_g) | (g \in \mathcal{G}) \wedge (\mathrm{ais}(g) = \{e^{!}_g, e^{?}_g\})\}$
$\leq_3 = \{(e, e') | ((e, e') \in \leq_4) \wedge Ord_2(\lambda(e), \lambda(e'))\}$
$\leq_4 = \bigcup_{((g,g') \in \leq_5) \wedge (g \neq g')} (\mathrm{ais}(g) \times \mathrm{ais}(g'))$
$\leq_5 = (\{(g, g') | ((g, g') \in \leq) \wedge Ord_1(\lambda(g), \lambda(g'))\})^\star$

3.3 Choreography Projection, Realizability and Reception-Completeness

The projection of a given choreography G onto a given component c is the CSM $\mathrm{sm}_c([\![G]\!])$. The CSM system of a given choreography G is considered correct (i.e. G is considered realizable) if, starting with every constituent CSM in the initial state and every buffer empty, it is unable to reach a deadlock (i.e. a state in which it cannot proceed in spite of some buffers non-empty or some constituent CSMs in a state with further actions defined) or execute a global action sequence not in $\mathrm{asq}([\![G]\!])$. If, in addition, the CSM system cannot reach any state that is reception-incomplete (see below), we say that the system and G are reception-complete. The current system state is called reception-incomplete if there exist a buffer b and a letter (c, c', m) for which one of the following is currently true:

(1) $Rv(b)$ and c is ready for $b!(c, c', m)$, but c' is not ready for $b?(c, c', m)$.
(2) $\neg Rv(b)$ and b allows immediate execution of $b?(c, c', m)$, but c' is not ready for it.

3.4 Auto-Concurrency

For a given action pomset set $\mathcal{R}$, let $Ac(\mathcal{R})$ denote the presence of auto-concurrency, i.e. that for some poset $(\mathcal{E}, \leq) \in \mathrm{pos}(\mathcal{R})$ and event pair $(e, e') \in \mathcal{E} \times \mathcal{E}$ with $\lambda(e) = \lambda(e')$, neither $e \leq e'$ nor $e' \leq e$ is true.

Lemma 1. *If a given choreography G satisfies $(|\langle\!\langle G \rangle\!\rangle| = 1) \wedge \neg Ac([\![G]\!])$, it is realizable and reception-complete.*

Proof: Suppose that the premise is true and consider the CSM system of G. What G prescribes is that (1) of any given action, the system executes exactly a certain number of instances, and (2) action instances are executed in a certain partial order.

Of any given action a, every component $c \in \mathrm{prt}(a)$ is ready to execute exactly the prescribed number of instances of $a\lfloor_c$. Hence, if G runs to completion, every component reaches a state with no further actions defined.

For any given buffer b with $\neg Rv(b)$ and letter l, the prescribed number of instances is the same for $b!l$ and $b?l$. Hence, if G runs to completion, every letter instance put in a buffer is also retrieved from it.

For any given action a, by $\neg Ac([\![G]\!])$, G prescribes a total ordering of its instances and every component $c \in \mathrm{prt}(a)$ implements a total ordering of instances of $a\lfloor_c$. Hence, for any given buffer b, letter l and natural i, one can, even if $Mb(b)$, safely assume that the i^{th} instance of $b?l$ corresponds to the i^{th} instance of $b!l$, and in case of $Rv(b)$ to the i^{th} instance of $b!?l$.

For any prescribed direct ordering (e_1, e_2) of two different action instances e_1 and e_2, one of the following is true:

(1) $\lambda(e_2)$ is a $b!(c, c', m)$ with $c \in \mathrm{prt}(\lambda(e_1))$, in which case e_2 is delayed by c until after e_1.
(2) $\lambda(e_2)$ is a $b!?(c, c', m)$ with $c \in \mathrm{prt}(\lambda(e_1))$, in which case the transmission instance in e_2 (and thereby e_2 itself) is delayed by c until after e_1.
(3) e_2 is an instance of a $b?l$ and e_1 is the corresponding transmission instance, in which case b delays its support for e_2 until after e_1.
(4) e_2 is an instance of a $b?l_2$ with $Fifo(b)$ and e_1 is an instance of a $b?l_1$ for whose corresponding transmission instance it is prescribed that it comes before the one corresponding to e_2, in which case b delays its support for e_2 until after e_1.
(5) e_2 is an instance of a $b?l$ with $Mb(b)$ and e_1 is an instance of $b?l$ for whose corresponding transmission instance it is prescribed that it comes before the one corresponding to e_2, in which case b delays its support for e_2 until after e_1.

Hence, no event is executed prematurely. Moreover, any given component c is at any given time during the execution of G ready for its part of any instance of a $b!?(c', c, m)$ that is currently supported by c', and for any instance of a $b?(c', c, m)$ with $\neg Rv(b)$ that is currently supported by b, implying that the CSM system of G is reception-complete and, hence, also deadlock-free. ■

3.5 Local Choice

For given non-empty action pomset sets $\mathcal{R}_1$ and $\mathcal{R}_2$, presumably two alternative sets of alternative behaviours of the system, let $Lc(\mathcal{R}_1, \mathcal{R}_2)$ denote that the choice between the two sets is local, i.e. that if it is ever made by the system, this is upon a transmission (possibly a part of a rendezvous) executed by a preselected component, after (or upon) which every other component for which the two alternatives are not identical is in time informed of the choice, upon a reception (possibly a part of a rendezvous). Formally, $Lc(\mathcal{R}_1, \mathcal{R}_2)$

denotes that there exists such a component set $\mathcal{C}'$ with $|\mathcal{C}'| \leq 1$ that for every component c, action sequence $\alpha \in (\mathrm{asq}_c(\mathcal{R}_1) \cap \mathrm{asq}_c(\mathcal{R}_2))$, $i \in \{1,2\}$ and action a with $\alpha a \in (\mathrm{asq}_c(R_i) \setminus \mathrm{asq}_c(\mathcal{R}_{3-i}))$, all the following is true:

(1) There exists an action a' with $\alpha a' \in \mathrm{asq}_c(\mathcal{R}_{3-i})$.
(2) If $c \in \mathcal{C}'$, then a is a $b!(c,c',m)$, otherwise it is a $b?(c',c,m)$.

Lemma 2. *If given realizable and reception-complete choreographies G_1 and G_2 satisfy $Lc([\![G_1]\!],[\![G_2]\!])$, the choreography $G = G_1 + G_2$ is realizable and reception-complete.*

Proof: Suppose that the premise is true and consider the CSM system of G. By the realizability of G_1 and G_2, the choice of a certain alternative $G_i \in \{G_1, G_2\}$ is made, if ever, when a certain component c for the first time executes an action which at the particular point it is ready to execute in G_i, but not in G_{3-i}. For any such event e, one of the following is true:

(1) e is an instance of a $b!(c,c',m)$ (possibly a part of a rendezvous). Hence, c is, by the constraint (2) in the definition of $Lc([\![G_1]\!],[\![G_2]\!])$, the preselected global selector.
(2) e is an instance of a $b?(c',c,m)$ and executed as a part of a rendezvous e' in which the transmission part is an instance e'' of $b!(c',c,m)$. e'' occurs at a point when, by the reception-completeness of G_{3-i}, c' is ready to execute $b!(c',c,m)$ in G_i, but not in G_{3-i}. By the assumption that G_i is not selected until upon e, the latter implies that c' also chooses G_i exactly upon e'. One can, hence, safely say that the choice of G_i is actually made not by c upon e, but by c' upon e''. Moreover, by the constraint (2) in the definition of $Lc([\![G_1]\!],[\![G_2]\!])$, c' is the preselected global selector.
(3) e is an instance of a $b?(c',c,m)$ with $\neg Rv(b)$ and executed at a point when $b?(c',c,m)$ is allowed by b in G_i, but, by the reception-completeness of G_{3-i}, not in G_{3-i}. The latter, however, implies that just before e, the fact that G_i is selected in the particular run is already evident from the current contents of b and, hence, also from the current event history, which contradicts the assumption that the choice is not made until upon e.

Hence, the first component, if any, to make the choice of a certain $G_i \in \{G_1, G_2\}$ is the preselected global selector, a c. Now suppose that at some later point, some other component c' becomes the first one to choose G_{3-i} instead, by the constraint (2) of $Lc([\![G_1]\!],[\![G_2]\!])$ upon executing a certain instance e of a certain $b?(c'',c',m)$ which at the particular point it is ready to execute in G_{3-i}, but not in G_i. By the reception-completeness of G_i, one of the following is true:

(1) $c'' \neq c$ and e is executed as a part of a rendezvous whose transmission part is executed by c'' when the component has already selected G_{3-i}.
(2) $\neg Rv(b)$ and e is executed when the contents of b is such as currently impossible in G_i, which can only be because at least one component in $\mathcal{C} \setminus \{c, c'\}$ has already selected G_{3-i}.

Both cases contradict the assumption that the first component to choose G_{3-i} is c'. Hence, every component chooses, if ever, the same G_i as c. Moreover, when a given component completes its part of G_i, it is, by the constraint (1) in the definition of $Lc([\![G_1]\!],[\![G_2]\!])$, in a state with no further actions defined. Hence, by the realizability of G_1 and G_2, G is realizable and, by the reception-completeness of G_1 and G_2, also reception-complete. ■

3.6 Choreography Well-Formedness

For a given action pomset set $\mathcal{R}$, let $Wb(\mathcal{R})$ denote that $\mathcal{R}$ is well-branched, i.e. that either $|\mathcal{R}| = 1$ or there exist non-empty action pomset sets $\mathcal{R}_1 \subset \mathcal{R}$ and $\mathcal{R}_2 \subset \mathcal{R}$ satisfying $(\mathcal{R}_1 \cup \mathcal{R}_2 = \mathcal{R}) \wedge Wb(\mathcal{R}_1) \wedge Wb(\mathcal{R}_2) \wedge Lc(\mathcal{R}_1, \mathcal{R}_2)$. Like [1], we define that a given choreography G is well-formed, which we denote as $Wf(G)$, if $\neg Ac([\![G]\!]) \wedge Wb([\![G]\!])$.

Proposition 1. *If a given choreography G satisfies $Wf(G)$, it is realizable and reception-complete.*

Proof: Suppose that the premise is true. Hence, there exists a choreography G' with $[\![G']\!] = [\![G]\!]$ in which every subterm is either a G'' with $(|\langle\!\langle G'' \rangle\!\rangle| = 1) \wedge \neg Ac([\![G'']\!])$ or a $G_1 + G_2$ with $Lc([\![G_1]\!],[\![G_2]\!])$. G' and all its subterms are well-formed. By induction on increasingly larger subterms of G', in each individual induction step using Lemma 1 or Lemma 2, respectively, one can, hence, prove that each of the subterms and G' itself are realizable and reception-complete. Hence, by $[\![G]\!] = [\![G']\!]$, G is realizable and reception-complete. ■

3.7 Some Rules for the Inference of Choreography Well-Formedness

In Section 2.5, the definition of $\lambda(\mathcal{R})$ for a given action pomset set $\mathcal{R}$ is uplifted to our more general setting. Below, the same is done for the predicate Ls used in [1]. With the two redefinitions, all the following inference rules proposed in [1] remain valid in the more general setting, together with their proofs provided in [1]:

Proposition 2. *If a choreography G satisfies $(|[\![G]\!]| = 1) \wedge \neg Ac([\![G]\!])$, then $Wf(G)$.*

Proposition 3. *If a choreography G is of the form $\mathbf{0}$ or $b : l$, then $Wf(G)$.*

Proposition 4. *If given choreographies G and G' satisfy $([\![G]\!] = [\![G']\!]) \wedge Wf(G)$, then $Wf(G')$.*

Proposition 5. *If for a given choreography G, there exist non-empty interaction pomset sets $\mathcal{R}_1$ and $\mathcal{R}_2$ with $(\mathcal{R}_1 \cup \mathcal{R}_2 = \langle\!\langle G \rangle\!\rangle) \wedge Wf(\mathcal{R}_1 + \mathcal{R}_2)$, then $Wf(G)$.*

Proposition 6. *If given choreographies G_1 and G_2 satisfy $Wf(G_1) \wedge Wf(G_2) \wedge Lc([\![G_1]\!], [\![G_2]\!])$, then $Wf(G_1 + G_2)$.*

Proposition 7. *If given choreographies G_1 and G_2 satisfy $Wf(G_1) \wedge Wf(G_2) \wedge (\lambda([\![G_1]\!]) \cap \lambda([\![G_2]\!]) = \emptyset)$, then $Wf(G_1|G_2)$.*

For the next rule, we first redefine the predicate $Ls(G_1, G_2)$ [1] that for given choreographies G_1 and G_2 denotes that they are locally strictly sequenced. In our more general setting, this is in case that for every interaction instance poset pair $((\mathcal{G}_1, \leq_1), (\mathcal{G}_2, \leq_2))$ in $\mathrm{pos}_1(\langle\!\langle G_1 \rangle\!\rangle) \times \mathrm{pos}_2(\langle\!\langle G_2 \rangle\!\rangle)$, the action instance poset $[\![(\mathcal{G}_1 \cup \mathcal{G}_2, (\leq_1 \cup \leq_2 \cup (\mathcal{G}_1 \times \mathcal{G}_2))^\star)]\!]$ is an $(\mathcal{E}, \leq)$ with $\leq \supseteq \bigcup_{c \in \mathcal{C}}((\bigcup_{g \in \mathcal{G}_1} \mathrm{ais}(g)\downarrow_c) \times (\bigcup_{g \in \mathcal{G}_2} \mathrm{ais}(g)\downarrow_c))$.

Proposition 8. *If given choreographies G_1 and G_2 satisfy $Wf(G_1) \wedge Wf(G_2) \wedge Ls(G_1, G_2)$, then $Wf(G_1; G_2)$.*

3.8 Inadequacy of the Three Recently Proposed Similar Definitions of Well-Formed Choreographies

In the examples presented in this section, any action $b!(c, c', m)$, $b?(c, c', m)$ or $b!?(c, c', m)$ whose (b, c, c') is evident from the context is denoted simply as $!m$, $?m$ or $!?m$, respectively.

Speaking in terms of the abstract concepts introduced in the previous subsections of Section 3, the conceptual difference between our definition of well-formed choreographies and those in [5]–[7] is that in [5]–[7], the underlying definition of the choreography semantics is virtually based on slightly modified predicate Ord_2, and in [5] also on slightly modified predicate Ord_1. The two modifications are as follows:

(1) The $Ord_2(a_1, a_2)$ which [5]–[7] implicitly employ for given actions a_1 and a_2 is virtually true also if a_2 is a $b!?(c, c', m)$ with $c' \in \mathrm{prt}(a_1)$ or a $b?(c, c', m)$ with $c' \in \mathrm{prt}(a_1)$.
(2) The $Ord_1(x_1, x_2)$ which [5] implicitly employs for given interactions $x_1 = b_1 : (c_1, c_1', m_1)$ and $x_2 = b_2 : (c_2, c_2', m_2)$ is virtually true also if $Rv(b_2) \wedge (c_2' \in \{c_1, c_1'\})$.

With the modification of Ord_2, particularly if the modification of Ord_1 is also present, it is no longer secured that the semantics $[\![G]\!]$ of a given choreography G avoids prescribing delayed reception, i.e. that the CSM system of G is reception-complete. Note, however, that in our proof of Lemma 1, the deadlock-freeness of the CSM system of the considered well-formed G with $|\langle\!\langle G \rangle\!\rangle| = 1$ is deduced from its reception-completeness, and that in our proof of Lemma 2, the reception-completeness assumed for the considered alternative choreographies G_1 and G_2 is employed for deducing that no letter instance belonging to a given $G_i \in \{G_1, G_2\}$ is interpreted by its recipient as one belonging to G_{3-i}. As the two lemmas are employed in our proof of Proposition 1, it is, hence, possible that with the modification of Ord_2, well-formed choreographies are not necessarily realizable. The following examples prove that this is indeed the case:

Example 1. Consider the choreography
$\mathsf{G} = (\mathsf{b_{AB}} : (\mathsf{A}, \mathsf{B}, \mathsf{x}); \mathsf{b_{BC}} : (\mathsf{B}, \mathsf{C}, \mathsf{y});$
$\mathsf{b_{CA}} : (\mathsf{C}, \mathsf{A}, \mathsf{a}); \mathsf{b_{AB}} : (\mathsf{A}, \mathsf{B}, \mathsf{x})) +$
$(\mathsf{b_{AC}} : (\mathsf{A}, \mathsf{C}, \mathsf{z}); \mathsf{b_{CB}} : (\mathsf{C}, \mathsf{B}, \mathsf{b});$
$\mathsf{b_{CA}} : (\mathsf{C}, \mathsf{A}, \mathsf{c}); \mathsf{b_{AB}} : (\mathsf{A}, \mathsf{B}, \mathsf{x}))$
where each of the buffers is a FIFO channel or a mailbox. With the modification of Ord_2, all the following is true, regardless of whether the modification of Ord_1 is also employed:
(1) $Wf(\mathsf{G})$
(2) $[\![\mathsf{G}]\!]$ forbids that !y and !z are both executed.
(3) In the CSM system of G, A chooses between the action sequences !x?a!x and !z?c!x, B chooses between ?x!y?x and ?b?x, and C chooses between ?y!a and ?z!b!c.
(4) Hence, the system possibly executes the action sequence !z?z!b!c?c!x?x!y (and thereby !y after !z) and then terminates, in a state in which $\mathsf{b_{BC}}$ and $\mathsf{b_{CB}}$ are non-empty and B wants to execute another ?x.

Example 2. Consider the choreography
$\mathsf{G} = (\mathsf{b_{AB}} : (\mathsf{A}, \mathsf{B}, \mathsf{x}); \mathsf{b_{AC}} : (\mathsf{A}, \mathsf{C}, \mathsf{y});$
$\mathsf{b_{CD}} : (\mathsf{C}, \mathsf{D}, \mathsf{z}); \mathsf{b_{BD}} : (\mathsf{B}, \mathsf{D}, \mathsf{w})) +$
$(\mathsf{b_{AC}} : (\mathsf{A}, \mathsf{C}, \mathsf{y}); \mathsf{b_{AB}} : (\mathsf{A}, \mathsf{B}, \mathsf{x});$
$\mathsf{b_{BD}} : (\mathsf{B}, \mathsf{D}, \mathsf{w}); \mathsf{b_{CD}} : (\mathsf{C}, \mathsf{D}, \mathsf{z}))$
where each of the buffers is a rendezvous channel. With the modification of Ord_1 only or with the modification of Ord_2 only, $[\![\mathsf{G}]\!]$ is as without the modifications and G is realizable and reception-complete. With both modifications, however, all the following is true:
(1) $Wf(\mathsf{G})$
(2) $[\![\mathsf{G}]\!]$ forbids the action sequence !?x!?y!?w!?z.
(3) In the CSM system of G, A chooses between the action sequences !x!y and !y!x, B's only option is ?x!w, C's only option is ?y!z, and D chooses between ?z?w and ?w?z.
(4) Hence, the system possibly executes the forbidden !?x!?y!?w!?z.

4 Final Remarks

By forbidding auto-concurrency and assuming that in every rendezvous, the only participants are a predefined sender and a predefined recipient, we easily adapted the in [1] proposed definition of well-formed choreographies to the considered more general kind of systems. In the future, it would be interesting to extend the search for easy-to-check sufficient conditions for choreography realizability also to choreographies in which individual rendezvous have multiple participants, and individual

rendezvous participants have no predefined roles, for note that symmetric multiway rendezvous are a very useful concept, particularly in early system design phases when the system is considered at a high level of abstraction [8].

ACKNOWLEDGEMENT

This work was financed by the Slovenian Research Agency (research programme P2-0095).

REFERENCES

[1] M. Kapus-Kolar, "Realizable causal-consistent reversible choreographies for systems with first-in-first-out communication channels," *J. Log. Algebraic Methods Program.*, vol. 114, p. 100560, 2020.

[2] E. Tuosto and R. Guanciale, "Semantics of global view of choreographies," *Journal of Logical and Algebraic Methods in Programming*, vol. 95, pp. 17–40, 2018.

[3] R. Guanciale and E. Tuosto, "Realisability of pomsets," *J. Log. Algebr. Meth. Program.*, vol. 108, pp. 69–89, 2019.

[4] D. Brand and P. Zafiropulo, "On communicating finite-state machines," *J. ACM*, vol. 30, no. 2, pp. 323–342, Apr. 1983.

[5] F. Barbanera, I. Lanese, and E. Tuosto, "Choreography automata," in *Proc. COORDINATION 2020* (Lecture Notes in Computer Science, vol. 12134), Springer, Berlin, 2020, pp. 86–106.

[6] K. Schewe, Y. A. Ameur, and S. Benyagoub, "Realisability of choreographies," in *Proc. FoIKS 2020* (Lecture Notes in Computer Science, vol. 12012), Springer, 2020, pp. 263–280.

[7] U. de'Liguoro, H. C. Melgratti, and E. Tuosto, "Towards refinable choreographies," in *Proc. ICE 2020* (EPTCS, vol. 324), 2020, pp. 61–77.

[8] H. Garavel and W. Serwe, "The unheralded value of the multiway rendezvous: Illustration with the production cell benchmark," in *Proc. MARS@ETAPS 2017* (EPTCS, vol. 244), 2017, pp. 230–270.

Monika Kapus-Kolar received her B.Sc. degree in electrical engineering from the University of Maribor, Slovenia, and her M.Sc. and Ph.D. degrees in computer science from the University of Ljubljana, Slovenia, in the years 1981, 1984 and 1989, respectively. Since 1981 she has been with the Jožef Stefan Institute, Ljubljana, where she is currently a senior researcher at the Department of Communication Systems. Her current research interests include formal specification techniques and methods for the development of real-time, concurrent and reactive systems.