Handling Reparation in Incremental Construction of Realizable Conversation Protocols

Sarah Benyagoub, Yamine Aït-Ameur, Meriem Ouederni, Atif Mashkoor

Handling Reparation in Incremental Construction of Realizable Conversation Protocols*

Sarah Benyagoub[1,2], Yamine Aït-Ameur[2], Meriem Ouederni[2], and Atif Mashkoor[3,4]

[1] University of Mostaganem, Algeria
[2] IRIT-INP of Toulouse, France
[3] Software Competence Center Hagenberg GmbH
[4] Johannes Kepler University Linz, Austria
{sarah.benyagoub, meriem.ouederni, yamine}@enseeiht.fr,
atif.mashkoor@{scch|jku}.at

A main concern, already addressed by the research community, relates to the verification of Conversation Protocol (CP) realizability, which means the existence of a set of peers whose communication behavior is equivalent to a given conversation protocol. In this paper, we consider the incremental repairability of CPs identified as un-realizable using the set of composition operators, defined in [2] that satisfy sufficient conditions for realizability preservation. Reparation consists in identifying a set of changes completing intermediate un-realizable CPs so that the resulting CP becomes realizable. Our proposal is validated through a successful application of the presented approach on un-realizable CPs borrowed from the literature.

1 Introduction

In a previous work [2], we presented a correct-by-construction approach of distributed systems. There, the interaction between systems is described as a conversation protocol (CP). A set of operators allow a developer to incrementally build the distributed systems while preserving (by construction) their realizability at each application of these operators.

1.1 Basic definitions

In the following, we summarize our correct-by-construction approach for realizable choreographies. We recall the main definitions for CP realizability as well as the set of composition operators together with their corresponding sufficient conditions.

* The research reported in this paper has been partly supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

Definition 1 (CP). *A conversation protocol CP (Figure 1) associated with a set of peers* $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ *(Figure 2) is a LTS* $CP = (S_{CP}, s^0_{CP}, L_{CP}, T_{CP})$ *where* S_{CP} *is a finite set of states and* $s^0_{CP} \in S_{CP}$ *is the initial state;* L_{CP} *is a set of labels and* T_{CP} *is the finite set of transitions.*

Definition 2 (CP_b). *A basic* CP_b *is a CP with a single transition defined as* $CP_b =< S_{CP_b}, s^0_{CP_b}, L_{CP_b}, T_{CP_b} >$ *and* $T_{CP_b} = \{s^0_{CP_b} \xrightarrow{m^{\mathcal{P}_i \to \mathcal{P}_j}} s'_{CP_b}\}$ *with* $s^0_{CP_b} \neq s'_{CP_b}$.

Definition 3 (Peer). *A peer is a LTS* $\mathcal{P} = (S, s^0, \Sigma, T)$ *where* S *is a finite set of states,* $s^0 \in S$ *is the initial state,* $\Sigma = \Sigma^! \cup \Sigma^? \cup \{\tau\}$ *is a finite alphabet partitioned into a set of send messages, receive messages, and the internal action, and* $T \subseteq S \times \Sigma \times S$ *is a transition relation.*

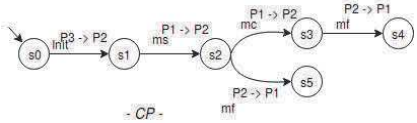

Fig. 1: Un-realizable CP.

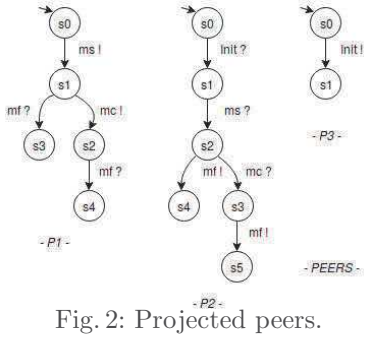

Fig. 2: Projected peers.

Definition 4 (Projection). *Let the projection function* $\downarrow CP$ *which returns the set of peers LTSs* $\mathcal{P}_i =< S_i, s^0_i, \Sigma_i, T_i >$. *The set is obtained by replacing in* $CP =< S_{CP}, s^0_{CP}, L_{CP}, T_{CP} >$ *each label* $(\mathcal{P}_j, m, \mathcal{P}_k) \in L_{CP}$ *with* $m!$ *if* $j = i$ *with* $m?$ *if* $k = i$ *and with* τ *(internal action). And finally removing the* τ*-transitions by applying standard minimization algorithms [5].*

Figures 1 and 2 show an example of a CP and its projection respectively.

Definition 5 (Realizability). *The definition of Realizability we use in this paper is borrowed from [1]. It is decomposed as the conjunction of three properties as Realizability = Equivalence* $\wedge$ *Synchronizability* $\wedge$ *Well-formedness.*

- **Equivalence** ($\equiv$). $CP \equiv Sys_{sync}(\downarrow CP)$ *iff CP and* $Sys_{sync}(\downarrow CP)$ *have equal message exchanges sequences, i.e., trace equivalence.*
- **Synchronizability**. *The synchronous system* $Sys_{sync}(\downarrow CP)$ *and the asynchronous system* $Sys_{async}(\downarrow CP)$ *are synchronizable iff the system behavior is still the same in both synchronous and asynchronous communications.*
- **Well-Formedness (WF)**. $Sys_{async}(\downarrow CP)$ *is well formed, i.e.,* $Sys_{async}(\downarrow CP) \in WF$ *iff all the queues of the asynchronous system become empty at the end of system composition.*

A correctness proof of global system realizability using Event-B is available in [3]. This approach is a posteriori, it is based on the whole CP and is not incremental.

1.2 Correct-by-construction realizable CP's operators

To avoid a posteriori global verification of realizablity, we have set up an incremental verification of realizability using a correct-by-construction approach to build CPs. This approach is based on the application of composition operators on basic realizable CPs. All these operators satisfy sufficient conditions which guarantee realizability. These operators are briefly described below.

Definition 6. (Sequential Composition $\otimes_{(\gg, s^f_{CP})}$). *Given a CP, a state $s_{CP} \in S^f_{CP}$, and a CP_b where $T_{CP_b} = \{s_{CP_b} \xrightarrow{l_{CP_b}} s'_{CP_b}\}$, the sequential composition $CP_{\gg} = \otimes_{(\gg, s_{CP})}(CP, CP_b)$ means that CP_b must be executed after CP starting from s_{CP}, and:*

- $S_{CP_{\gg}} = S_{CP} \cup \{s'_{CP_b} | \; s_{CP_b} \xrightarrow{l_{CP_b}} s'_{CP_b} \in T_{CP_b}\}$
- $L_{CP_{\gg}} = L_{CP} \cup \{l_{CP_b}\}$
- $T_{CP_{\gg}} = T_{CP} \cup \{s_{CP} \xrightarrow{l_{CP_b}} s'_{CP_b}\}$
- $S^f_{CP_{\gg}} = (S^f_{CP} \setminus \{s_{CP}\}) \cup \{s'_{CP_b}\}$

Definition 7. (Choice Composition $\otimes_{(+, s^f_{CP})}$). *Given a CP, a state $s_{CP} \in S^f_{CP}$, a set $\{CP_{bi} \mid i = [1..n], n \in \mathbb{N}\}$ such that $\forall\, T_{CP_{bi}}$, $T_{CP_{bi}} = \{s_{CP_{bi}} \xrightarrow{l_{CP_{bi}}} s'_{CP_{bi}}\}$, the branching composition $CP_+ = \otimes_{(+, s_{CP})}(CP, \{CP_{bi}\})$ means that CP must be executed before $\{CP_{bi}\}$ and there is a choice between all $\{CP_{bi}\}$ at s_{CP}, and*

- $S_{CP_+} = S_{CP} \cup \{s'_{CP_b 1}, \ldots, s_{CP'_{bn}} | \; s_{CP_{bi}} \xrightarrow{l_{CP_{bi}}} s'_{CP_{bi}} \in T_{CP_{bi}}\}$
- $L_{CP_+} = L_{CP} \cup \{l_{CP_{bi}}, \ldots, l_{CP_{bn}}\}$
- $T_{CP_+} = T_{CP} \cup \{s_{CP} \xrightarrow{l_{CP_{b1}}} s'_{CP_{b1}}, \ldots, s_{CP} \xrightarrow{l_{CP_{bn}}} s'_{CP_{bn}}\}$
- $S^f_{CP_+} = (S^f_{CP} \setminus \{s_{CP}\}) \cup \{s'_{CP_{b1}}, \ldots, s'_{CP_{bn}}\}$

Definition 8. (Loop Composition $\otimes_{(\circlearrowleft, s^f_{CP})}$). *Given a CP, a state $s_{CP} \in S^f_{CP}$ and a basic CP noted CP_b, with $T_{CP_b} = \{s_{CP_b} \xrightarrow{l_{CP_b}} s'_{CP_b}\}$ and $s'_{CP_b} \in S_{CP}$, then the loop composition $CP_{\circlearrowleft} = \otimes_{(\circlearrowleft, s_{CP})}(CP, CP_b)$ is defined as follows.*

- $S_{CP_{\circlearrowleft}} = S_{CP}$
- $L_{CP_{\circlearrowleft}} = L_{CP} \cup \{l_{CP_b}\}$
- $T_{CP_{\circlearrowleft}} = T_{CP} \cup \{s_{CP} \xrightarrow{l_{CP_b}} s'_{CP_b}\}$
- $S^f_{CP_{\circlearrowleft}} = S^f_{CP}$

The condition $s'_{CP_b} \in S_{CP}$ means that the target state of CP_b is a state of CP. It defines a cycle in the built $CP_{\circlearrowleft}$, thus a loop and an iteration. The final states remain unchanged.

According to [2], we have identified a set of sufficient conditions which entail realizability when the CPs are built using the previously defined operators. Let us first formally define these conditions.

Condition 1 (Deterministic Choice (DC)) *Given a CP, deterministic choice property, denoted DC(CP), holds iff $\forall s_{CP} \in S_{CP}, \nexists \{s_{CP} \xrightarrow{m^{P_i,P_j}} s'_{CP}, s_{CP} \xrightarrow{m^{P_i,P_j}} s''_{CP}\} \subseteq T_{CP}$, such that $s'_{CP} \neq s''_{CP}$*

Condition 2 (Parallel-Choice Freeness (PCF)) *Let PCF be the set of CPs. The parallel choice freeness property (PCF), denoted as $CP \in PCF$, holds iff* $\forall s_{CP} \in S_{CP}, \nexists\{s_{CP} \xrightarrow{m^{P_i,P_j}} s'_{CP},\ s_{CP} \xrightarrow{m'^{P_k,P_q}} s''_{CP}\} \subseteq T_{CP}$ *such that* $P_i \neq P_k$ *and* $s'_{CP} \neq s''_{CP}$.

Condition 3 (Independent Sequences Freeness (ISeqF)) *Let ISeqF be the set of CPs free of independent sequences. The independent sequence freeness property, denoted as $CP \in ISeqF$ holds iff* $\forall s_{CP} \in S_{CP}$, $\nexists\{s_{CP} \xrightarrow{m^{P_i,P_j}} s'_{CP},\ s'_{CP} \xrightarrow{m'^{P_k,P_q}} s''_{CP}\} \subseteq T_{CP}$ *such that* $P_i \neq P_k$ *and* $P_j \neq P_k$.

The sufficient conditions associated with each composition operators can be defined. Table 1 recalls all the theorems that ensure the realizability of a *CP* built incrementally using each composition operator. Each theorem relies on the previously introduced sufficient conditions. More details on the definitions and proofs of these theorems are available in [2].

Theorem 1	$CP_b \in R$
Theorem 2	$CP \in R \wedge CP_b \in R \wedge CP_{\gg} = \otimes_{(\gg, s^f_{CP})}(CP, CP_b) \in \text{ISeqF} \Rightarrow CP_{\gg} \in R$
Theorem 3	$CP \in R \wedge \{CP_{bi}\} \subseteq R \wedge CP_{+} = \otimes_{(+, s^f_{CP})}(CP, \{CP_{bi}\}) \in \text{DC}$ $\wedge\ CP_{+} \in \text{ISeqF} \wedge\ CP_{+} \in \text{PCF} \ \Rightarrow\ CP_{+} \in R$
Theorem 4	$CP \in R \wedge CP_b \in R \wedge CP_{\circlearrowleft} = \otimes_{(\circlearrowleft, s^f_{CP})}(CP, CP_b) \in \text{ISeqF} \Rightarrow\ CP_{\circlearrowleft} \in R$

Table 1: Theorems for realizable by construction CPs

1.3 Related work

The choreography repair technique presented in [**?**] depends on examining and analyzing the cause of violation of the realizablity condition [1]. In other words, the approach propose a realizability verification and reparation on the whole CP, to check the equivalence, the synchronizability and the well-formedness properties. Both verification and reparation techniques require building of synchronous and asynchronous traces that increase the complexity of verification and reparation.
The verification and reparation approach of [4] proposes an automated and non-intrusive solution for enforcing realizability when a choreography is not realizable. Their idea is to generate distributed controllers that are in charge of correcting ordering issues to make the corresponding distributed peers respect the choreography requirements. To do this, both synchronous and asynchronous communications are needed to check the realizability condition given in [1]. Notice that, the reparation proposed in [4] is not a generic repair method. Such that, a choreography is not repairable when at some point in its behavior there is a choice between interactions involving different sending peers. In that case, realizability cannot be enforced.
To avoid the aforementioned situations, the idea is, instead of checking and repairing the realizability on the whole system, we propose to check and repair the CP incrementally starting from an empty CP. To achieve this objective, our reparation strategy is based on the sufficient conditions satisfied by the set of composition operators [2]. Each

operator can build a realizable CP from another realizable CP and a basic one without needing the projected peers or the synchronous and asynchronous traces. Notice that, there is no general repair method for un-realizable CP. Each violated sufficient condition gives rules for reparation, by adding a synchronization transition which reestablishes the sufficient conditions that restore the CP realizability.

1.4 Case study

In order to illustrate our approach, we use a case study borrowed from [**?**]. The choreography describes a simple file transfer protocol where P_1 is a client asking for the file transfer, P_2 is a file server and P_3 initializes the communication between a client and a server. This CP is depicted in Figure 1. First, the client sends a message ($init$) to the server to request the server to start the transfer (ms). When the transfer is finished, the server sends the "Transfer Finished" (mf) message and the protocol terminates. However, the client may decide to cancel the transfer before hearing back from the server by sending a "Cancel Finished" message (mc) in which case the server responds with "Transfer Finished" (mf) message, which, again, terminates the protocol.

In order to check the realizability condition given in Definition 5, we rely on a stepwise correct-by-construction approach to build incrementally a realizable CP. The approach consists in applying the different operators on a set of basic CPs by checking the sufficient conditions associated with each composition operator. A sequence of steps is set up to build the conversation protocol of Figure 1 as follows.

1. Identification of the set of basic CPs involved in the CP of Figure 1.
   - CP $= \varnothing$
   - $CP_{b0} = s_0 \xrightarrow{Init^{P3\to P2}} s_1$
   - $CP_{b1} = s_1 \xrightarrow{ms^{P1\to P2}} s_2$
   - $CP_{b2} = s_2 \xrightarrow{mc^{P1\to P2}} s_3$
   - $CP_{b3} = s_3 \xrightarrow{mf^{P2\to P1}} s_4$
   - $CP_{b4} = s_2 \xrightarrow{mf^{P2\to P1}} s_5$

2. Application of the composition operators.
   (a) $CP_1 = \otimes_{(\gg, s^1_{CP})}(CP, cp_{b0}), \checkmark$ $\quad CP_1 \in$ ISeqF
   (b) $CP_2 = \otimes_{(\gg, s^1_{CP})}(CP_1, cp_{b1}), \times$ $\quad CP_2 \notin$ ISeqF

The sequence of composition starts from an empty CP. The sufficient condition ISeqF holds for the first composition CP_1. So, by Theorem 1 of Table 1, CP_1 (a) is realizable. However, realizability does not hold for CP_2 (b) where the ISeqF property is violated.

In the following section, we show how such un-realizable CPs can be repaired.

2 Incremental Reparation

2.1 General Idea

The sufficient conditions are not satisfied by the CP in Figure 1. In this example, both sequences and branches violate the associated sufficient conditions.

Therefore, the CP must be transformed in order to restore ISeqF and PCF properties while preserving the initial communication purpose. To address this issue, we propose to introduce synchronization transitions with synchronization messages. These messages are not relevant for the communication purpose, but they are added for synchronization and realizability purposes.

Two reparation cases can be distinguished for both sequence and branch operators as follows.

– *Sequence property repair.* Following the ISeqF definition, the reparation of the sequence transitions (ISeqF violation) requires the introduction of a novel transition with message $Sync0$ (bold-dotted in Figures 3 and 4) between the two independent sequences. This transition exchanges a synchronization message between the sender or the receiver peers of the first transition and the sender of the second transition.

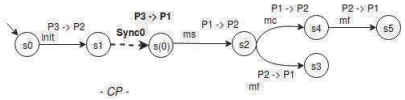

Fig. 3: ISeqF repair proposition 1.

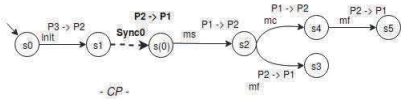

Fig. 4: ISeqF repair proposition 2.

– *Branch properties repair.* Following the PCF definition, the reparation of the branch transitions, (PCF violation) requires the introduction of a novel transition with message $Sync1$ (bold-dotted in Figures 5 and 6) before one of the branches transitions. This transition exchanges a synchronization message between the same sender peer as the other branches and the receiver one.

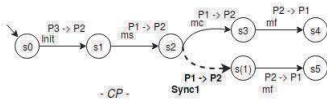

Fig. 5: PCF repair proposition 1.

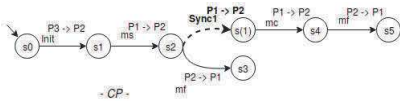

Fig. 6: PCF repair proposition 2.

2.2 Application to the case study

According to the previous reparation possibilities, four reparation scenarios are possible. One of the possible CP reparation is obtained by combination one reparation from the two sequence reparations and one from the two branches reparations. The CP of Figure 1 is depicted in Figure 7. The realizable projection is presented in Figure 8.

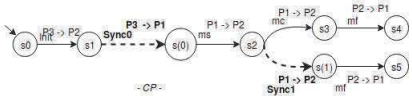

Fig. 7: Un-realizable CP repair.

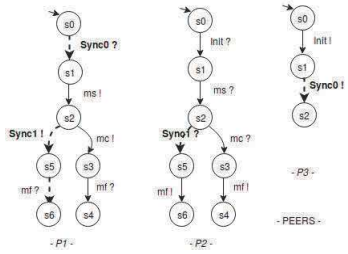

Fig. 8: Projected peers repair.

In the sequel, we show that there exists a sequence of compositions of operators that lead the CP depicted in Figure 7. This sequence is defined as follows.

– Identification of the set of basic CPs and initialization of CP

- CP $= \varnothing$
- $CP_{b0} = s_0 \xrightarrow{Init^{P3 \to P2}} s_1$
- $CP_{b1} = s_1 \xrightarrow{Sync0^{P3 \to P1}} s(0)$
- $CP_{b2} = s(0) \xrightarrow{ms^{P1 \to P2}} s_2$
- $CP_{b3} = s_2 \xrightarrow{mc^{P1 \to P2}} s_3$
- $CP_{b4} = s_3 \xrightarrow{mf^{P2 \to P1}} s_4$
- $CP_{b5} = s_2 \xrightarrow{Sync1^{P1 \to P2}} s(1)$
- $CP_{b6} = s(1) \xrightarrow{mf^{P2 \to P1}} s_5$

– Application of the composition operators.

1. $CP_1 = \otimes_{(\gg, s^1_{CP})}(CP, CP_{b0}), \checkmark$ $\quad CP_1 \in$ ISeqF
2. $CP_2 = \otimes_{(\gg, s(0)_{CP})}(CP_1, CP_{b1}), \checkmark$ $\quad CP_2 \in$ ISeqF
3. $CP_3 = \otimes_{(+, s^2_{CP})}(CP_1, \{CP_{b3}, CP_{b5}\}), \checkmark$ $\quad CP_3 \in$ ISeqF $\wedge\, CP_3 \in$ DC $\wedge CP_3 \in$ PCF
4. $CP_4 = \otimes_{(\gg, s^3_{CP})}(CP_3, CP_{b4}), \checkmark$ $\quad CP_2 \in$ ISeqF
5. $CP_5 = \otimes_{(\gg, s(1)_{CP})}(CP_4, CP_{b6}), \checkmark$ $\quad CP_5 \in$ ISeqF

The previous composition operators are successfully applied. So, the obtained CP is realizable.

3 Conclusion

In this paper, we present a top down approach to repair an un-realizable distributed systems. The proposal is based on the application of composition operators to check the realizability of systems. In case where the sufficient conditions associated with each operator are not satisfied, intermediate CPs, behaving as synchronization transitions, are introduced for adaptation purposes. In the future, we aim at implementing the reparation strategy we have introduced in this paper using the correct-by-construction Event-B method. The idea consists in introducing reparation events corresponding to the different situations of sufficient conditions violations.

References

1. Basu, S., Bultan, T., Ouederni, M.: Deciding Choreography Realizability. In: Proc. of POPL'12. pp. 191–202. ACM (2012)

2. Basu, S., Bultan, T.: Automatic choreography repair. Theor. Comput. Sci. (2015)
3. Farah, Z., Ait-Ameur, Y., Ouederni, M., Tari, K.: A Correct-by-Construction Model for Asynchronously Communicating Systems. International Journal STTT pp. 1–21 (2016)
4. Güdemann, M., Poizat, P., Salaün, G., Ye, L.: Verchor: a framework for the design and verification of choreographies. IEEE Transactions on Services Computing 9(4), 647–660 (2016)
5. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison Wesley (1979)