

# Preserving Data Consistency in Process Choreographies by Design

Tom Lichtenstein

Hasso Plattner Institute, University of Potsdam, Potsdam, Germany  
Tom.Lichtenstein@hpi.de

**Abstract.** Data is essential for the execution of business processes. As today's organizations increasingly collaborate in process choreographies, data relevant to process execution is typically shared among participants. To avoid conflicts in the execution of process choreographies, the preservation of data consistency must be considered in the design of a choreography. However, current choreography modeling languages provide limited data modeling capabilities, thus potential conflicts arising from data inconsistencies at runtime may remain undetected during design time. Therefore, this paper motivates a framework allowing the design of data consistency-aware process choreographies.

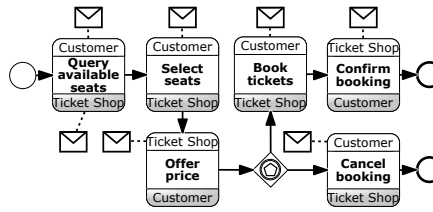
**Keywords:** Process Choreography · Data Consistency · Design Time

## 1 Introduction and Motivating Example

In the current age of information, business processes and their outcomes rely on data and its manipulation during process execution [16]. As today's organizations increasingly collaborate in *process choreographies* [18], data relevant to business process execution is typically shared across choreography participants. To ensure the correct execution of the individual business processes involved in a process choreography, *data consistency*, i.e., a uniform view of data shared across multiple nodes [15], is a desirable property to be maintained among participants [12]. As the participants can only exchange data in the form of messages, maintaining data consistency is particularly challenging in process choreographies. Local changes to data shared with other participants can result in inconsistent views of the data, which may lead to conflicts such as unexpected interactions or data constraint violations on the receiving end. Since any deviation from the agreed interaction behavior can affect operations and incur costs, process choreographies require careful coordination [18]. Therefore, interaction behavior arising from data inconsistencies needs to be considered during process choreography design. The consideration of data consistency during design time is particularly relevant for process choreographies since in process orchestrations consistency can usually be enforced via central database management systems.

To support the design of process choreographies, graphical modeling languages such as *Business Process Model and Notation* (BPMN) [1] can be used. However, most contemporary choreography modeling languages provide only

limited data modeling capabilities. Therefore, potentially erroneous behavior resulting from data inconsistencies may remain undetected during design time. To illustrate the relevance of data management in choreography design, we consider a simple online ticket reservation choreography depicted in Figure 1. The choreography starts with a customer querying and selecting available seats for an event. Then, depending on the price offer, the customer decides whether to cancel the reservation or book the tickets. In the latter case, the ticket store confirms the reservation. While the interaction behavior is locally enforceable [19], data inconsistencies may arise with concurrent executions. Assuming that the ticket shop can sell only one ticket for each seat, if two customers select overlapping seats and the reservation of the first customer is confirmed, the second customer has an inconsistent view of the available seats. Furthermore, since the confirmation of the second reservation would lead to a constraint violation on the part of the ticket shop, compensation behavior is required to restore the consistency between the participants. Still, the need for compensation and the compensation behavior itself are both not evident from the given model. In the following, related work in the area of data consistency preservation is discussed and a framework for designing consistency-aware process choreographies is motivated.



**Fig. 1.** Ticket reservation choreography between a customer and an online ticket shop

## 2 Related Work

Maintaining consistency in distributed environments is extensively studied in literature [3,6,10]. In particular, the preservation of data consistency is addressed by *consensus protocols* [11] which allow distributed nodes to agree on specific data values required for further computation, thus providing a consistent view of the data. The application of consensus protocols to process choreographies is discussed by Weber et al. [17]. The authors propose the use of blockchain technology [14] as an execution environment for process choreographies. To model and execute blockchain-driven choreographies, Ladleif et al. [9] refine BPMN 2.0 choreography diagrams [1] with blockchain-specific extensions. However, maintaining consensus in process choreographies may introduce synchronization overhead as not every change needs to be propagated to all participants. Moreover, in scenarios where inconsistencies rarely lead to conflicts, sustaining consensus among all participants may limit concurrent behavior and thus affect throughput. Incorporating data consistency management into the interaction design instead allows the choreography to be flexibly tailored to the use case.

Despite the existence of approaches that enrich choreography models with data management-specific information [5,13], the preservation of data consistency in process choreographies with concurrent instances has received little attention in research. Hahn et al. [5] introduce an approach that decouples the data flow between participants from the message flow by introducing a middleware that coordinates cross-partner data objects among corresponding participants. Yet, the handling of concurrent accesses to data shared by multiple instances is not further specified in their approach. Haarmann et al. [4] introduce a framework to analyze the use of shared data in process models by defining data access semantics. While the framework includes concepts for data consistency preservation, it does not address data exchange between individual participants. Finally, Kopp et al. [8] propose choreography spheres ensuring transactional behavior for all included activities that may belong to different processes. Still, deciding on an adequate scope of the spheres might prove challenging without data-related information, as overly large transactions could create avoidable overhead.

### 3 Towards Data Consistency in Process Choreographies

Since the integration of existing data consistency-preserving concepts into the design of process choreographies poses various challenges as outlined in the previous section, this paper proposes the concept of a framework supporting the collaborative design of data consistency-aware process choreographies based on the BPMN modeling language. The framework is supposed to enable the detection of potential data inconsistencies between participants at design time that may lead to conflicts at runtime. To achieve this, the framework will include functionality to specify and verify data consistency-related information in process choreography diagrams. Thus, participants should be enabled to define data consistency constraints at the interaction level (i.e., in the public process) that must be followed in the individual data management of each participant's local behavior (i.e., in the private process). The specifications will also allow participants to identify activities or data objects that may be affected by inconsistent data and therefore require careful attention in their design. In addition, data consistency criteria are introduced to formally verify the preservation of consistency throughout the choreography with regard to the specifications. The criteria should allow the detection of interaction behaviors that potentially lead to conflicts considering data inconsistencies that are not resolved in subsequent interactions. To design the criteria, existing formal definitions of consistency models [15] and data-aware choreographies [2,7] will be considered. Formal verification based on choreography models may also require extending the BPMN modeling language to include information about the individual management of the exchanged data. Eventually, the framework can be used to automatically derive enhanced process designs when potential conflicts are identified.

Based on this framework, a methodology will be developed to enable business engineers to iteratively redesign interaction behaviors prone to inconsistencies. The redesigned behavior is supposed to either ensure data consistency until

the interactions affected by the data are completed or include appropriate compensation behavior to restore consistency. To realize the proposed concept, the following research questions need to be answered:

- Which information is required to reason about data consistency across multiple interactions while also considering concurrent behavior?
- How can data consistency-related information be integrated into the design of process choreographies?
- How can the behavior of process choreographies be enhanced to avoid conflicts caused by data inconsistencies?

## 4 Conclusion

This position paper discusses the need for considering the preservation of data consistency when designing process choreographies. Since current choreography modeling languages provide limited support for data modeling, potential conflicts due to data inconsistencies between participants may go undetected during design time. By integrating data consistency-related information into the design of process choreographies, corresponding conflicts can be detected and addressed to ensure a more reliable interaction behavior.

## References

1. Business process model and notation (BPMN), version 2.0 (2011), <http://www.omg.org/spec/BPMN/2.0>
2. Corradini, F., Muzi, C., Re, B., Rossi, L., Tiezzi, F.: Animating multiple instances in BPMN collaborations: From formal semantics to tool support. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *Business Process Management, Lecture Notes in Computer Science*, vol. 11080, pp. 83–101. Springer International Publishing, Cham (2018). [https://doi.org/10.1007/978-3-319-98648-7\\_6](https://doi.org/10.1007/978-3-319-98648-7_6)
3. Gilbert, S., Lynch, N.: Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News* **33**(2), 51–59 (2002). <https://doi.org/10.1145/564585.564601>
4. Haarmann, S., Weske, M.: Cross-case data objects in business processes: Semantics and analysis. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) *Business Process Management Forum, Lecture Notes in Business Information Processing*, vol. 392, pp. 3–17. Springer International Publishing, Cham (2020). [https://doi.org/10.1007/978-3-030-58638-6\\_1](https://doi.org/10.1007/978-3-030-58638-6_1)
5. Hahn, M., Breitenbücher, U., Kopp, O., Leymann, F.: Modeling and execution of data-aware choreographies: an overview. *Computer Science - Research and Development* **33**(3-4), 329–340 (2018). <https://doi.org/10.1007/s00450-017-0387-y>
6. Herlihy, M.P., Wing, J.M.: Linearizability: a correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems* **12**(3), 463–492 (1990). <https://doi.org/10.1145/78969.78972>
7. Knuplesch, D., Pryss, R., Reichert, M.: Data-aware interaction in distributed and collaborative workflows: Modeling, semantics, correctness. In: Pu, C., Joshi, J., Nepal, S. (eds.) *Proceedings of the 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing. IEEE* (2012). <https://doi.org/10.4108/icst.collaboratecom.2012.250443>

8. Kopp, O., Wieland, M., Leymann, F.: Towards choreography transactions. In: Oliver Kopp, Niels Lohmann (eds.) 1st Central-European Workshop on Services and their Composition, ZEUS 2009, Stuttgart, Germany, March 2-3, 2009. Proceedings. CEUR Workshop Proceedings, CEUR-WS.org (2009)
9. Ladleif, J., Weske, M., Weber, I.: Modeling and enforcing blockchain-based choreographies. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) Business Process Management, Lecture Notes in Computer Science, vol. 11675, pp. 69–85. Springer International Publishing, Cham (2019). [https://doi.org/10.1007/978-3-030-26619-6\\_7](https://doi.org/10.1007/978-3-030-26619-6_7)
10. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM* **21**(7), 558–565 (1978). <https://doi.org/10.1145/359545.359563>
11. Lamport, L.: The part-time parliament. *ACM Transactions on Computer Systems* **16**(2), 133–169 (1998). <https://doi.org/10.1145/279227.279229>
12. Liu, X.: Data consistency enforcement on business process transactions. In: Service-Oriented Computing, pp. 394–399. Springer Berlin Heidelberg (2013). [https://doi.org/10.1007/978-3-642-37804-1\\_40](https://doi.org/10.1007/978-3-642-37804-1_40)
13. Meyer, A., Pufahl, L., Batoulis, K., Fahland, D., Weske, M.: Automating data exchange in process choreographies. *Information Systems* **53**, 296–329 (2015). <https://doi.org/10.1016/j.is.2015.03.008>
14. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
15. Tanenbaum, A.S., van Steen, M.: Distributed Systems: Principles and Paradigms. Pearson Education (2008)
16. Tsoury, A., Soffer, P., Reinhartz-Berger, I.: Data impact analysis in business processes. *Business & Information Systems Engineering* **62**(1), 41–60 (2020). <https://doi.org/10.1007/s12599-019-00611-5>
17. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted business process monitoring and execution using blockchain. In: La Rosa, M., Loos, P., Pastor, O. (eds.) Business Process Management, Lecture Notes in Computer Science, vol. 9850, pp. 329–347. Springer International Publishing, Cham (2016). [https://doi.org/10.1007/978-3-319-45348-4\\_19](https://doi.org/10.1007/978-3-319-45348-4_19)
18. Weske, M.: Business Process Management. Springer Berlin Heidelberg, Berlin, Heidelberg (2019). <https://doi.org/10.1007/978-3-662-59432-2>
19. Zaha, J., Dumas, M., Hofstede, A., Barros, A., Decker, G.: Service interaction modeling: Bridging global and local views. In: 2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC’06). pp. 45–55. IEEE (10/16/2006 - 10/16/2006). <https://doi.org/10.1109/EDOC.2006.50>