Automating Data Exchange in Process Choreographies (Extended Abstract)[3]

D. Fahland[1] Andreas Meyer, Luise Pufahl, Kimon Batoulis, Mathias Weske[2]

Abstract: Communication between organizations is formalized as process choreographies in daily business. While the correct ordering of exchanged messages can be modeled and enacted with current choreography techniques, no approach exists to describe and automate the exchange of data between processes in a choreography using messages. This paper describes an entirely model-driven approach for BPMN introducing a few concepts that suffice to model data retrieval, data transformation, message exchange, and correlation – four aspects of data exchange. For automation, this work utilizes a recent concept to enact data dependencies in internal processes. We present a modeling guideline to derive local process models from a given choreography; their operational semantics allows to correctly enact the entire choreography from the derived models only including the exchange of data. Targeting on successful interactions, we discuss means to ensure correct process choreography modeling. Finally, we implemented our approach by extending the *camunda BPM platform* with our approach and show its feasibility by realizing all service interaction patterns using only model-based concepts. The work summarized in this extended abstract has been published in [Me15].

Keywords: Process Modeling, Data Modeling, Process Choreography, Data Exchange, BPMN, SQL

1 Introduction and Problem Description

In daily business, organizations interact with each other, e.g., concluding contracts or exchanging information. Fig. 1(left) describes an interaction between a customer and a supplier with respect to a request for a quote. The customer sends the *request* to a chosen supplier which internally processes it and sends the resulting *quote* as response which then is handled internally by the customer. An interaction between business processes of multiple organizations via message exchange is called *process choreography*. In BPMN (Business Process Model and Notation), a *choreography diagram* describes the order of message *exchanges* between multiple participants from a global view (*global* choreography model). A *collaboration diagram* describes message exchanges are realized via *send* and *receive* activities, distributed over the different participants (*local choreography model*).

This paper considers how to implement local choreography models that adhere to a global agreement; we focus on a top-down approach where all participants jointly agree on a

[1] d.fahland@tue.nl, Eindhoven University of Technology, The Netherlands

[2] andreas.meyer@hpi.de, luise.pufahl@hpi.de, kimon.batoulis@hpi.de, mathias.weske@hpi.de, Hasso Plattner Institute at the University of Potsdam, Germany

[3] This article summarizes problem, approach, and selected findings of a study published as *Andreas Meyer, Luise Pufahl, Kimon Batoulis, Dirk Fahland, Mathias Weske: Automating data exchange in process choreographies. Inf. Syst. 53: 296-329 (2015)* doi:10.1016/j.is.2015.03.008 [Me15]

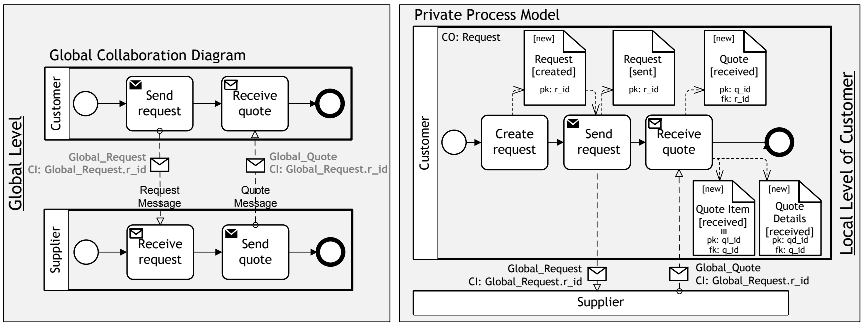

Fig. 1: Implementing a given global collaboration diagram (left) in a private process model (right)

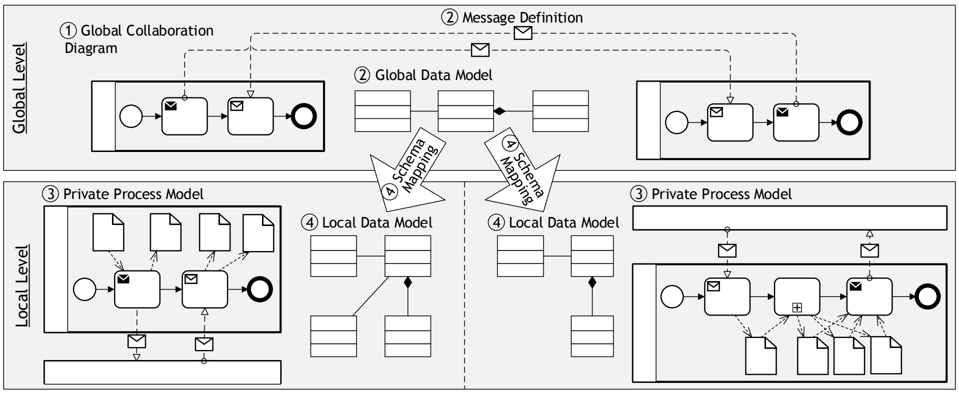

Fig. 2: Modeling Guideline for deriving local process models from a global choreography model.

global data exchange and collaboration model to which each participant's local process and data models either must adhere or are required to be changed accordingly [vdAW01]. Deriving a local choreography from a global one is a non-trivial step; various techniques are required [DW11] including *locally enforcing* the order of globally specified message exchanges. In general, both control-flow (order of message exchange) and data-flow (actual message contents) need to be addressed when transitioning from global to local models. *The paper contributes an approach to for integrating control-flow and data-flow in message exchange; we present a few syntactic extensions to BPMN together with operational semantics that allow to model and realize the intended global interaction using model-based concepts only.*

2 Approach

We combine several existing approaches to automate data exchange in process choreographies entirely model-driven as follows; these approaches are used along the general guideline shown in Fig. 2. (1) All participants agree on a global choreography model expressed in BPMN as shown in Fig. 1(left); BPMN will also be used for the local choreography models.

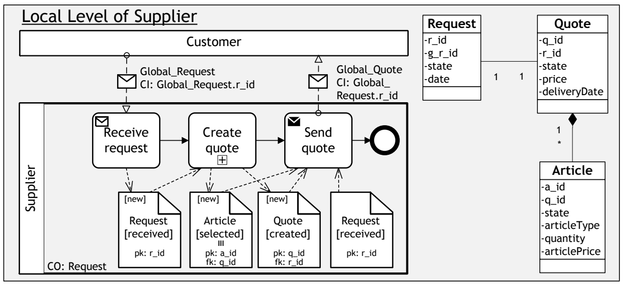

Fig. 3: Local process model and local data model of the supplier

(2) In addition, we introduce that all participants globally agree to specific data exchange formats used in the collaboration modeled in UML; Fig 4(top) shows the global data model. (3) For mapping the control-flow of a global choreography model into local ones, we utilize the Public-to-Private approach [vdAW01] unchanged; Fig. 1(right) and Fig. 3 show the two local process models obtained from the choreography of Fig. 1(left).

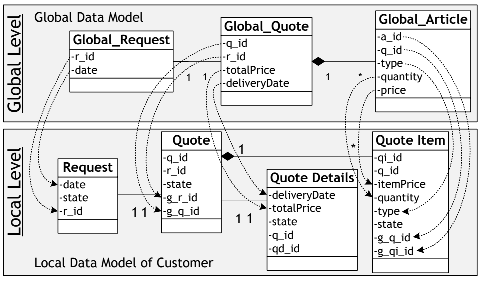

Fig. 4: Data schema mapping from global data model (messages) to local data model of the Customer.

(4) Next, we map the data perspective of the global choreography to each local process model by defining a straightforward attribute-level data schema mapping between global and local data models. When exchanging messages local data is translated from the local data model of the sender to the globally agreed on data model, and then translated by the recipient to its local data model. This way heterogenous local data models are isolated from each other and are synchronized. By marking specific attributes of the global data model as *correlation identifiers* (e.g., *Global_Request.r_id* in Fig. 1(left)) we obtain locally usable correlation keys through the above mentioned data schema mapping; these local correlation identifiers can then be used as proposed in the BPMN standard to associate an incoming message to the right process instance.

(5) To process (create and store) messages in a model-driven fashion, we apply (and slightly extend) the approach in [Me13] for automatically deriving SQL queries from BPMN data objects to enact complex data-dependencies. Thereby, we utilize the notion of dedicated (multi-instance) case objects for subprocesses from to realize 1:n communication with a set of participants. Fig. 5 illustrates the *syntax* and the *operational semantics* we provided with this extension. *1.)* To send *Global_Quote* message, activity *Send Quote* first automatically derives SQL queries based on the annotations to its input data objects [Me13] (a *Quote* object related to the quote instance, a *Request* and several *Article* object related to the quote). *2.)* The retrieved local data is transformed into a message carrying a data object of the global data model by an XQuery generated from the data schema mapping of Fig. 4. *3.+4.)* The message is sent and received using standard communication protocols. *5.)* The recipient *correlates* the received message through the designated correlation key *Global_Request.r_id*; technically, we identify the process instance of the *Customer* that holds a *Quote* object with attribute *r_id* having the same value as the message;

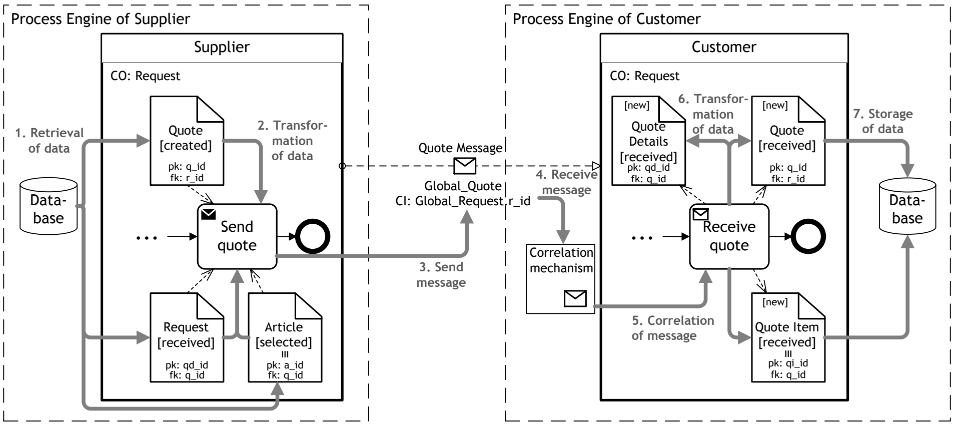

Fig. 5: Operational semantics of data exchange from model-driven concepts.

this query is generated automatically. *6.+7.)* The received message is transformed into local data objects (using the data schema mapping) which are then stored in the database of the *Customer*, again using queries derived from the output data objects of task *Receive Quote* [Me13].

3 Results

We integrated several existing, isolated techniques for fully model-based definition and enactment of choreographies. We provided a few minor syntactic extensions of BPMN with complimentary operational semantics that translates model features into executable, platform-independent code [Me15]; we implemented our approach based on the *Camunda* BPM engine; see `http://bpt.hpi.uni-potsdam.de/Public/BPMNData`. Using this engine, we created fully model-driven implementations of *all service interaction patterns* [BDtH05] including patterns on 1:n message exchange with multiple participants using identical/different correlation identifiers, and referral of correlation identifiers.

References

[BDtH05] Barros, Alistair; Dumas, Marlon; ter Hofstede, Arthur H. M.: Service interaction patterns. In: BPM 2005. volume 3649 of LNCS. Springer, pp. 302–318, 2005.

[DW11] Decker, Gero; Weske, Mathias: Interaction-centric modeling of process choreographies. Information Systems, 36(2):292–312, 2011.

[Me13] Meyer, Andreas; Pufahl, Luise; Fahland, Dirk; Weske, Mathias: Modeling and Enacting Complex Data Dependencies in Business Processes. In: BPM 2013. volume 8094 of LNCS. Springer, pp. 171–186, 2013.

[Me15] Meyer, Andreas; Pufahl, Luise; Batoulis, Kimon; Fahland, Dirk; Weske, Mathias: Automating data exchange in process choreographies. Inf. Syst., 53:296–329, 2015.

[vdAW01] van der Aalst, Wil M. P.; Weske, Mathias: The P2P approach to interorganizational workflows. In: Advanced Information Systems Engineering. Springer, pp. 140–156, 2001.