Information Systems 53 (2015) 296–329

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Automating data exchange in process choreographies

Andreas Meyer[a,*], Luise Pufahl[a], Kimon Batoulis[a], Dirk Fahland[b], Mathias Weske[a]

[a] Hasso Plattner Institute at the University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, D-14482 Potsdam, Germany
[b] Eindhoven University of Technology, Den Dolech 2, 5612 AZ Eindhoven, The Netherlands

ARTICLE INFO

Available online 8 May 2015

Keywords:
Process modeling
Data modeling
Process choreographies
Data exchange
BPMN
SQL

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.

1. Introduction

In daily business, organizations interact with each other, e.g., concluding contracts or exchanging information. Fig. 1 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* [1]. The industry standard BPMN (Business Process Model and Notation) [2] provides the following concepts to model process choreographies. A *choreography diagram* describes the order of message *exchanges* between multiple participants from a global view, called *global choreography model*. The message exchanges are then refined into *send* and *receive activities* distributed over the different participants. This can be captured in *collaboration diagrams* describing how each participant's *public* process interacts with other participants [3], also called *local choreography model*.

Problem context: This problem of implementing local choreography models that adhere to a global agreement can be approached in two ways: top-down or bottom-up. Following the top-down methodology, all participants jointly agree on a 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 [4]. Conversely, if local process or data models are the starting point and *not* to be changed, often, a

[*] Corresponding author.
E-mail addresses: andreas.meyer@hpi.de (A. Meyer), luise.pufahl@hpi.de (L. Pufahl), kimon.batoulis@hpi.de (K. Batoulis), dfahland@tue.nl (D. Fahland), mathias.weske@hpi.de (M. Weske).

http://dx.doi.org/10.1016/j.is.2015.03.008
0306-4379/© 2015 Elsevier Ltd. All rights reserved.

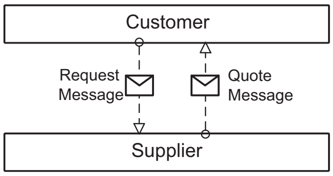

Fig. 1. Request for quote choreography.

mediator is required to realize the collaboration. Such mediator can be implemented through the Enterprise Integration Patterns [5] or orchestration services [6]. In this paper, we follow the top-down methodology.

Moreover, there are two fundamental paradigms for modeling processes, either focusing on the process' activities or on the process' objects. In the latter paradigm, the process is described in terms of the involved objects, their state changes, and synchronizations between objects [7–11]. In contrast, in the activity-centric paradigm, the process is described by the activities that can be performed and their order; states of data objects can be used to describe pre- and post-conditions for activity execution and to describe decisions. We refer the reader to [12] for a thorough discussion on differences and commonalities between both paradigms. Our approach bases on the activity-driven paradigm with BPMN [2] being the industry standard.

Deriving a local choreography from a global one is a non-trivial step; various techniques are required [13] 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.

Typically, choreography models are used to globally agree on a contract about the messages exchanged and their order. In the above example, both participants agreed that first the customer may send a request to the supplier which is then answered with a quote by the supplier. Based on the agreement, each participant has to implement its public process as a *private* process describing the executable part of this participant including the interactions with other participants as described in the choreography; this private process is called a *process orchestration* [14]. Existing approaches for deriving an orchestration for each participant from a choreography, such as the *Public-to-Private* approach [3], only cover the control-flow perspective of the contract: ensuring the correct *order* of messages. In the following, we address the correct *contents* of messages to achieve a correct *data exchange* that realizes the choreography.

Specific challenges: Generally, organizations store their data in local databases that other choreography participants cannot access. These databases follow local data schemes which differ among the organizations. However, the interacting organizations want to exchange data and therefore have to provide the information to be sent in a format which is understood at the receiving side. Thus, an agreed exchange message format has to be part of the global contract mentioned above. For a successful process choreography, it has to be ensured that messages to be sent are provided correctly and that received messages are processed correctly *based on the global contract*. In more detail, three challenges for collaboration with respect to data flow arise:

*C*1—*Data heterogeneity*: Interacting participants, such as our customer and supplier, each implement their own data schema for handling their private data. For sending a message to another participant, this local data has to be transformed into a message the recipient can understand. In turn, the received message has to be transformed into the local data schema to allow storing and processing by the recipient; i.e., C1 is about mapping between global and local data models.

*C*2*a*—*Correlation*: A participant may run *multiple instances* simultaneously. A message sent to a participant is typically intended for a particular process instance and must only be received by that instance. Assigning a message to the intended process instance is called correlation and may happen through dedicated *correlation identifiers* stored in the message. The challenge here is to populate correlation identifiers correctly and to correctly match a message to the right process instance.

*C*2*b*—1:*n communication*: In addition to one participant running multiple instances of its process, a single instance of that process may need to interact with *multiple* process instances of another participant at the same time. For example one customer may send multiple requests for a quote to multiple suppliers and receives multiple corresponding answers. The challenge here is to produce multiple messages for different participants and to process multiple incoming messages from different participants.

Although challenges C2a and C2b are closely related, they require distinct solutions. A correlation mechanism ensures that one message arrives at its intended receiver. Here 1:n communication adds another dimension to the problem as it requires to consistently handle a *set* of correlation identifiers and to process *sets* of messages and *all* their contents.

Current choreography modeling languages such as BPMN do not provide concepts to solve C1, C2a, and C2b. Instead, each participant manually implements message creation and processing for their private process, which is error-prone, hard to maintain, and easily results in incompatibilities to other participants in the choreography.

Proposed solution: In this paper, we combine several existing approaches to automate data exchange in process choreographies entirely model-driven as follows:

1. All participants agree on a global choreography model expressed in BPMN [2]; BPMN will also be used for the local choreography models.
2. In addition, we introduce that all participants globally agree to specific data exchange formats used in the collaboration modeled in UML [15].
3. For mapping the control flow of a global choreography model into local ones, we utilize the Public-to-Private approach [3] unchanged.
4. We use a straight-forward attribute-level data mapping between global and local data models to address challenge C1.
5. We utilize correlation identifiers, proposed in the BPMN standard, that are specified as part of the data exchange format and naturally translate to locally usable correlation keys by the above-mentioned data mapping to address challenge C2a.