Data-Aware Service Choreographies through Transparent Data Exchange

Michael Hahn, Dimka Karastoyanova, and Frank Leymann

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{hahnml, karastoyanova, leymann}@iaas.uni-stuttgart.de

BIBTEX:

```
@inproceedings{HKL16,
  author    = {Michael Hahn and Dimka Karastoyanova and Frank Leymann},
  title     = {Data-Aware Service Choreographies through Transparent Data
               Exchange},
  booktitle = {Proceedings of the 16\textsuperscript{th} International
               Conference on Web Engineering, ICWE 2016, 6-9 June 2016, Lugano,
               Switzerland},
  year      = {2016},
  pages     = {357--364},
  doi       = {10.1007/978-3-319-38791-8_20},
  series    = {Lecture Notes in Computer Science (LNCS)},
  volume    = {9671},
  publisher = {Springer International Publishing}
}
```

Data-Aware Service Choreographies through Transparent Data Exchange

Michael Hahn, Dimka Karastoyanova, and Frank Leymann

Institute of Architecture of Application Systems (IAAS)
University of Stuttgart, Stuttgart, Germany
{firstname.lastname}@iaas.uni-stuttgart.de

Abstract Our focus in this paper is on enabling the decoupling of data flow, data exchange and management from the control flow in service compositions and choreographies through novel middleware abstractions and realization. This allows us to perform the data flow of choreographies in a peer-to-peer fashion decoupled from their control flow. Our work is motivated by the increasing importance and business value of data in the fields of business process management, scientific workflows and the Internet of Things, all of which profiting from the recent advances in data science and Big data. Our approach comprises an application life cycle that inherently introduces data exchange and management as a first-class citizen and defines the functions and artifacts necessary for enabling transparent data exchange. Moreover, we present an architecture of the supporting system that contains the Transparent Data Exchange middleware, which enables the data exchange and management on behalf of service choreographies and provides methods for the optimization of the data exchange during their execution.

Keywords: Service Choreographies, Transparent Data Exchange, Decentralized Data Flow, Data Flow Optimization

1 Introduction

With the advances in the fields of Big Data and the Internet of Things (IoT) the importance of data in terms of its business value and as a driver for gaining advantages over competitors is increasing significantly. The impact of this development on the domain of Business Process Management (BPM) has already been documented [9,11]. In the domain of eScience data-centric aspects of computations belong to the core requirements [1,12]. In recent years a convergence of approaches from BPM and eScience is taking place and business processes are successfully applied to automate computer-based experiments and scientific calculations. Through our experience in the fields of BPM and eScience, and based on existing literature, we argue that business processes need to reflect this paradigm shift to data-awareness and provide support for the efficient integration and exchange of heterogeneous data through a central role in the BPM life cycle.

Business processes implemented through service compositions can be specified by following one of two paradigms: service orchestrations and choreographies [6].

The former ones are also known as workflows and are modeled from the viewpoint of one party which acts as a central coordinator. Service choreographies provide a global perspective of the potentially complex conversations between multiple interacting services, which are often implemented by workflows again. Each party that takes part in the collaboration, a so-called participant, is able to model its conversations with the other parties by specifying corresponding message exchanges with other participants [5]. Participants in a choreography can communicate in a direct, peer-to-peer manner without requiring any central coordinator that controls their interaction. Service choreographies have been successfully applied in both the business and eScience domain [2, 6, 8, 14].

Existing research already shows that conducting the data exchange in a decentralized manner provides valuable performance benefits [3, 4, 7], however it fails to accommodate all requirements from both BPM and eScience perspective. For example, the model-driven approach presented in [8] introduces capabilities to model and enact data exchange on the level of choreographies, but fails to incorporate mechanisms to decouple the control and data flow since data is still passed through message exchanges between participants. The works of Barker et al. [2, 3] introduce a proprietary service choreography language and a framework for its execution, and a framework based on service proxies, respectively. Both works show performance improvements due to decentralized data exchange in a choreography-like manner, but miss other optimization opportunities like transparent data exchange in parallel to the actual control flow of the conversations. Approaches like [7] and [4], rely on the decomposition of service compositions into so-called service proxies or triggers based on analysis of their data dependencies. A central coordinator controls the tightly coupled control and data flow, whereas the decoupling of control commands and data exchange happens only on the level of the invoked services.

In this work we present our vision for an approach towards introducing data as a first-class citizen in service choreographies. With this approach we want to provide support for the specification and handling of data-related aspects throughout the whole BPM life cycle and to resolve the tight coupling of data flow from control flow, which in choreographies results mainly from the fact that data can only be passed through pre-specified conversations between participants. Towards this goal, in Sect. 2 we introduce an extended choreography management life cycle that supports data-related aspects throughout all phases. In Sect. 3 an architecture for a modeling and enactment environment is presented that implements the introduced data-aware life cycle based on a new Transparent Data Exchange (TraDE) middleware layer. The research challenges we face towards achieving our goals are described in Sect. 4. Finally, we present a summary and conclusions in Sect. 5.

2 Approach

To account for data-awareness in service choreographies we use an approach of introducing modeling abstractions, data exchange and management methods to the traditional BPM life cycle. In Fig. 1 we present our proposal for a data-aware service choreography management life cycle that is based on the traditional BPM life cycle [16] and available extensions for choreographies in [6,14]. In the following, we describe the life cycle phases, their relations, the software artifacts they produce or consume and how each of the phases employs the new *TraDE methods* to support data awareness as a separate concern in the development and execution of a choreography.

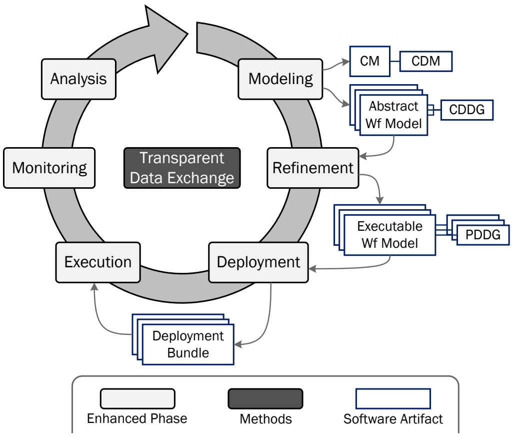

Figure 1. Data-Aware Service Choreography Management Life Cycle

The *TraDE methods* bundle a set of data-related methods and transformations to support data awareness throughout the whole life cycle and potential optimizations regarding the data perspective of choreographies.

In the *Modeling* phase the different stakeholders, e. g., domain experts of different fields in eScience or business specialists from different companies, who want to collaborate, define their interactions by specifying corresponding participants and their conversations, called also message exchanges, in a choreography model. The choreography model can be seen as a collaboration contract on which all participants agree. The contract contains the definition of visible behavior of the participants that can potentially be realized using executable workflows and the message exchange definitions. BPEL4Chor models [5], BPMN collaboration models or Let's Dance models [18] can be used as underlying modeling notation to represent choreography models. Our approach extends this collaboration contract with the data being exchanged through the conversations between the different partners by introducing an explicit data model and data flow between the participants on the level of the choreography. The resulting *Choreography Data Model* (CDM) provides the foundations for the data awareness in the later life cycle phases and allows us to realize most of the phase transitions and model enhancements in a (semi-)automated manner. Based on the analysis of the specified message exchanges a corresponding CDM can be generated and then manually refined or extended for further use by the TraDE methods in later life cycle phases.

The target of the *Analysis* phase in conjunction with the modeling phase is to produce choreography and workflow models that are optimal with respect to a set of requirements. Additionally, already existing models from earlier life cycle iterations together with their monitoring data are taken into consideration to find optimizations for the models.

As in the classical BPM life cycle at the end of the modeling phase a *Transformation* step takes place and produces an abstract workflow model for each of the specified choreography participants. The generated workflow models together implement the globally agreed collaboration behavior and are used as templates in the refinement phase. These workflows are normally not directly executable since they lack required details for successful deployment. BPMN process models or abstract BPEL processes can be used, e. g., to represent these abstract workflows. The transformation uses the results from the TraDE methods like the defined CDM and the modeled choreography data flow to generate the abstract workflow models. In our approach, the modeled data flow and the resulting data dependencies between the choreographed participants are transformed to a corresponding *Choreography Data Dependence Graph* (CDDG). Furthermore, the workflows are enriched with so-called *Staging Elements* that reflect data exchange between participants from their own viewpoint. The final output of the modeling phase comprises the choreography model (CM), its data model (CDM) and the generated abstract workflow models with the overall CDDG as shown in Fig. 1.

During the *Refinement* phase IT specialists refine the generated abstract workflow models into executable ones. This comprises the specification of the participants internal logic by adding new model constructs like activities, the control flow and data flow between them, as well as the required configuration data for the envisaged run time environment where the choreographed workflows will be executed. When specifying corresponding data flow between the activities the IT specialists have to model where and how the shared data is used in the workflow. After the manual refinement is completed we are analyzing the workflow models to extract the new information about the internal data dependencies and the data flow. For each executable workflow model a so-called *Participant Data Dependence Graph* (PDDG) is generated. For this, the CDDG created during transformation is split into subgraphs where each subgraph represents the data dependencies of one participant (PDDG). Additionally, all activities added during refinement that read or write data from or to a Data Object are added to the PDDG. At the end of the refinement phase a collection of executable workflow models together with their PDDGs is available for deployment (Fig. 1).

In the *Deployment* phase the executable workflow models are packaged in the required *Deployment Bundles* format and deployed to the target workflow middleware. It is the responsibility of the TraDE methods to identify the appropriate static or dynamic deployment strategy based on information like data dependence graphs, monitoring data or manually defined deployment requirements.

After the executable workflow models are deployed they enter the *Execution* phase. By instantiating one or more of the deployed models, e. g., on behalf of a client's requests, the overall choreography is executed through the started interrelated workflow instances which together realize the modeled behavior of the choreography. In the following we use the term *choreography instance* introduced in [15] to describe these groups of interrelated workflow instances without implying that there is a central entity coordinating them. During the execution of the choreography instance each of the participating workflow instances produces a set of events that provide information about executed activities, control and data flow, occurred exceptions or faults and many other aspects. These events are analyzed by the TraDE methods to detect potential data flow optimizations during run time of a choreography instance, e. g., in terms of strategies for optimal data placement, transferring data in advance based on predictions calculated using monitoring information or optimal data life cycle management so that the data is only stored as long as required and as short as possible.

The execution events are collected and analyzed during the *Monitoring* phase. For the monitoring of choreography instances the event data of the involved workflow instances needs to be analyzed, combined and interpreted. For example, the status of the choreography instance has to be calculated based on the status of all workflow instances. The resulting data can be expressed in form of higher-level choreography events, so that the interpretation and combination is done only once and other interested parties are able to directly consume the choreography events. An environment that enables the monitoring of choreographies is introduced in [17]. To support data-awareness, the explicitly modeled data flow and any data flow adaptations during run time triggered by optimization have to be captured.

3 Architecture

Figure 2 shows the overall architecture of the software system enabling the modeling and enactment of data-aware service choreographies. Each participant has a *Choreography and Orchestration Modeling Environment* to model his part of the overall choreography. The modeling environment supports the transformation of the choreography model to a set of abstract workflows where each of the abstract workflows can be further refined to an executable workflow model. The TraDE facilities are integrated into the modeling environment and enable the analysis of the choreography data model (CDM), the generation of the choreography (CDDG) and participant data dependence graphs (PDDG) and the optimization of data-related aspects on the level of both the choreography and the workflow models based on analysis results.

The deployment bundles contain the executable workflow models and their PDDGs. The workflow models (WfM) representing the choreographed services are deployed (depicted by the solid black arrows in Fig. 2) into a corresponding workflow management system (WfMS) for execution and the PDDGs to a TraDE middleware. The deployed WfMs and PDDGs are necessary to conduct the overall choreography and the exchange, placement and staging of the related data in an

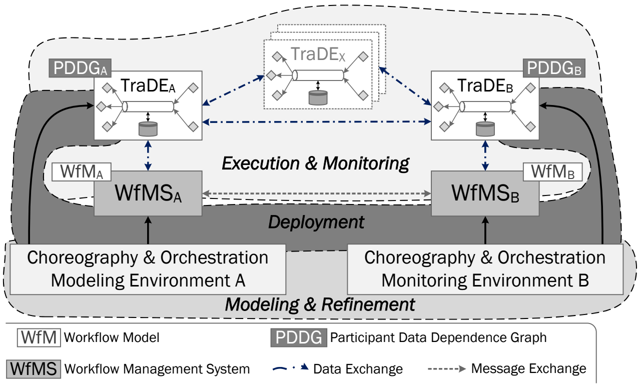
PDDG_A
TraDE_A
TraDE_X
PDDG_B
TraDE_B
Execution & Monitoring
WfM_A
WfMS_A
WfMS_B
WfM_B
Deployment
Choreography & Orchestration Modeling Environment A
Choreography & Orchestration Monitoring Environment B
Modeling & Refinement
WfM Workflow Model
PDDG Participant Data Dependence Graph
WfMS Workflow Management System
Data Exchange
Message Exchange

Figure 2. Architecture of an environment that supports the modeling and enactment of data-aware service choreographies

optimal manner. The TraDE middleware uses the information collected in the PDDGs and the CDDG as well as event data of previously executed choreography executions to perform and optimize the data exchange and placement according to the chosen optimization strategy.

As shown in Fig. 2, the TraDE middleware is not a single software component, but rather a network of multiple TraDE nodes in a distributed system. The TraDE middleware clients experience the whole collection of nodes as one single coherent middleware [13]. During choreography execution the WfMSs executing the participant workflows communicate with each other through messages that transport data or trigger corresponding functionality at other participants. In addition, the WfMS and its associated TraDE node are also communicating in terms of handling and optimizing the data exchange between the participants. Based on how the two middleware systems are integrated, this communication looks different. One approach is to extend the WfMS so that it actively invokes corresponding functionality at the TraDE node through its APIs. Alternatively, the TraDE node can be loosely integrated with the WfMS by consuming all emitted execution events of the WfMS to react accordingly based on the information stored in the PDDGs, i. e., reading or writing data through the APIs of the WfMS and transferring it to other TraDE nodes. The TraDE middleware also emits data-related events that allow for the monitoring of the data staging, placement and exchange to ensure that the optimized data flow is still carried out according to the choreography and workflow models.

4 Research Challenges

On the road towards realizing our Transparent Data Exchange vision we face research challenges related to both the modeling aspects of data-aware choreographies and execution and monitoring aspects. The challenges are on the levels of new abstractions, architecture and realization mechanisms.

Through the extension of the traditional BPM life cycle with data management functions we provide a preliminary approach towards rendering data exchange in choreographies and orchestrations as first-class citizen. The architecture we presented accommodates our vision for transparent data exchange and is supported by the newly introduced modeling artifacts like CDM, CDDG, and PDDG. The modeling of choreographies decoupling data exchange from their conversations will require in addition formal definitions of CDM, CDDG, PDDG and corresponding data analysis and optimization algorithms to derive the dependency graphs, suggest improvements, and allow for propagating the data dependencies from the level of the workflow models by refining the PDDGs. Therefore, realizing the transformation step during the modeling and the refinement phase will be one of our major objectives. Addressing the challenges with respect to the modeling aspects is a prerequisite to enable the execution of data-aware choreographies. A major objective of ours is the architecture and realization of the distributed TraDE middleware, the TraDE nodes and the communication protocols among nodes, the most appropriate integration approach with the WfMS as well as enforcing data security. The TraDE middleware will also (a) rely on fault-tolerant, asynchronous data exchange among participants for which we will define models and protocols, (b) will incorporate data shipping mechanisms, which poses the question of how these mechanisms are going to be integrated into the WfMS and ESB middlewares, (c) will enable data reuse across choreographies and services, which is a matter of data identification and mechanisms for their transparent delivery, and (d) will allow for the use of different data sources and formats by using our pluggable data management framework SIMPL [10]. A challenge concerning all components of the execution environment is the correlation of data exchange to the correct instance of a choreography, workflow or service. Monitoring of the data exchange, data staging and placement will require special attention and will provide valuable input to our optimization algorithms and strategies, which are also part of the conceptual work with respect to the execution perspective of our vision.

5 Conclusion

The efficient exchange of data between choreographed services is a crucial factor in classical data-centric domains like eScience. However, with evolving paradigms like Big data or IoT data exchange becomes also an important factor for the business domain. Existing research showed that in terms of data exchange the most promising approach is to decouple the data flow from the control flow definition and handle it in a decentralized manner by exchanging the data directly between the composed services. While most of the existing approaches only utilize the performance benefits from decentralizing the data flow, we want to provide further optimizations throughout all life cycle phases and especially during choreography run time. Towards this goal, we introduced a data-aware service choreography management life cycle that is enriched with so-called TraDE methods for data flow analysis and optimization. Furthermore, a system architecture that implements the extended life cycle was introduced. Based on our experiences and the discussed related work, we presented a set of research challenges that represent our road map for future work.

Acknowledgment

The authors would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/2) at the University of Stuttgart.

References

1. Barga, R., Gannon, D.: Workflows for e-Science: Scientific Workflows for Grids, chap. Scientific versus Business Workflows. Springer (2007)
2. Barker, A., et al.: Choreographing web services. IEEE Transactions on Services Computing (2009)
3. Barker, A., et al.: Reducing data transfer in service-oriented architectures: The circulate approach. IEEE Transactions on Services Computing (2012)
4. Binder, W., et al.: Decentralized orchestration of composite web services. In: ICWS'06 (2006)
5. Decker, G., et al.: BPEL4Chor: Extending BPEL for modeling choreographies. In: ICWS'07 (2007)
6. Decker, G., et al.: An introduction to service choreographies. Information Technology (2008)
7. Liu, D., et al.: Data-flow distribution in ficas service composition infrastructure. In: ICPDCS'02 (2002)
8. Meyer, A., et al.: Automating data exchange in process choreographies. Information Systems (2015)
9. Meyer, S., et al.: Towards modeling real-world aware business processes. In: WoT'11 (2011)
10. Reimann, P., et al.: Simpl-a framework for accessing external data in simulation workflows. In: BTW (2011)
11. Schmidt, R., et al.: Big data as strategic enabler - insights from central European enterprises. In: BIS. Springer (2014)
12. Slominski, A.: Workflows for e-Science: Scientific Workflows for Grids, chap. Adapting BPEL to Scientific Workflows. Springer (2007)
13. Tanenbaum, A.S., Van Steen, M.: Distributed systems. Prentice-Hall (2007)
14. Weiß, A., Karastoyanova, D.: A life cycle for coupled multi-scale, multi-field experiments realized through choreographies. In: EDOC'14 (2014)
15. Weiß, A., et al.: Rewinding and repeating scientific choreographies. In: OTM 2015 Conferences. Springer (2015)
16. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer Science & Business Media (2012)
17. Wetzstein, B., et al.: Cross-organizational process monitoring based on service choreographies. In: SAC'10 (2010)
18. Zaha, J., et al.: Let's dance: A language for service behavior modeling. In: OTM 2006 Conferences. Springer (2006)

All links were last followed on March 22, 2016.