



HAL
open science

Position paper: An integration approach of service composition models: from choreography to orchestration

Sirine Rebai, Hatem Hadj Kacem, Ahmed Hadj Kacem, Nawal Guermouche

► To cite this version:

Sirine Rebai, Hatem Hadj Kacem, Ahmed Hadj Kacem, Nawal Guermouche. Position paper: An integration approach of service composition models: from choreography to orchestration. 2012. hal-00733861

HAL Id: hal-00733861

<https://hal.science/hal-00733861>

Submitted on 19 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Position paper: An integration approach of service composition models: From choreography to orchestration

Sirine Rebai ^{*1}, Hatem Hadj Kacem ^{†1}, Ahmed Hadj Kacem ^{‡1},
and Nawal Guermouche ^{§2,3}

¹ReDCAD Laboratory, University of Sfax, Tunisia

²CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse,
France

³Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France

March 31, 2012

Abstract

This paper introduces an approach addressing the transition from choreography to orchestration. As far as validation method is achieved through the use of model-checker. We are interested in various levels of transformations. In fact, transformations are different in their departure points. Initially, the process of transformation, which we want to establish is based on a set of heterogeneous business protocols. This unit will consequently provide a choreography as result, which unifies them. So, to ensure the integration processing, we define a meta-model for the choreography model. At this level, certain verifications will take place. Through them, we will check a set of properties related to the choreographies maintained after the integration. Then, this choreography will be transformed into an orchestration. Thus, coherence between choreography and orchestration must be checked while showing the relation of conformity between these two different composition models. The obtained orchestration must preserve the total behavior of the system as specified in choreography and support all the responsibility, which is already defined by the other model (choreography). So, to face the undecidability of the problem of synthesis, the choice of the best transformation will be supported by the number of preserved properties. Indeed, this passage is accompanied by a verification

*si.rebai@gmail.com

†hatem.hadjkacem@fsegs.rnu.tn

‡ahmed.hadjkacem@fsegs.rnu.tn

§Nawal.Guermouche@laas.fr

phase. A set of properties will be preserved, the ones, which refer to the choreography are considered as common and others, which are relative to those checked by the orchestration are judged as local. Towards the end of this process, and in order to validate the transformation, we must check the maintained properties. This checking will be illustrated by the use of the model-checker, which depends on the properties.

1 Introduction

Nowadays, we note that there is a significant increase in system complexity. Thus, the issue of reusability is always a major challenge. In fact, software architectures have always been the bridge between strategy and development of information systems [3]. They are an effective response to problems faced by companies in terms of reusability, interoperability in addition to reducing coupling between the various systems that implement their information systems. The service oriented approach seeks to implement the reuse of an existing service [18]. SOAs are better known under their version of web service, that appeared as a new technology offering a means to exchange data between applications [7]. Indeed, the development of applications based on web services is ensured by using the composition technique, which is essential to the applications implementation. It is defined as "the process of combining existing services to form new services." Choreography and orchestration are two different mechanisms used in service composition. Choreography provides an abstract specification to achieve a common business goal, while orchestration provides execution details which are needed to realize this goal. The orchestration process contains additional information not found in the choreography specification (Private or internal actions for each role, the details in the error handling, execution details...). Although, the choreography serves as a contract between the different service partners, it cannot be performed by itself: its models are not executable. To fill this gap, we have to transform choreography specification to orchestration process, which is the practical implementation of the choreography. It brings the contribution of one party to the service composition. In fact, these mechanisms are different in their nature and they complement each other.

Moreover, there are many works, which are interested in the transformation process. The majority of them, based on the Model Driven Architecture concepts, have studied the transformation from CDL specifications to BPEL process. In addition, they have been interested in verifying some aspects of conformance relationship between the choreography and the orchestration. In this work, we overcome these problems by introducing an approach, which consists in the integration of service composition models and we seek to validate our transformation process through the adequate model-checker.

We have structured this paper as follows: In Section 2, an overview of service composition is presented. We give a survey of related work in Section 3. In Section 4, we present our approach followed by a section describing a case study. The last section contains a short conclusion and tracks for future works.

2 Service Composition

Web services are a key technology for supporting and performing business processes [1]. The orchestration and choreography are two complementary mechanisms for service composition at different abstraction levels.

2.1 Orchestration

An orchestration is a process viewed only from the perspective of a business partner, which takes the control of the process. It offers a centralized vision. Also, in this composition model, the involved services are under control of a single endpoint central process (orchestrator). This process coordinates the execution of different actions on the involved web services. These latter do not know and do not need to know that they are involved in a composition process. In fact, orchestration describes the way in which separate web services can be brought together in a consistent manner to provide a higher value service. It could only describe a single process when multiple processes interact with each other.

2.2 Choreography

The choreography has been introduced as a new view on services interaction. It is a description of abstract protocols. It offers a collaborative decentralized coordination. Choreography, which is descriptive in nature, describes the interaction contract between two or more web services [12]. It helps to describe the services behavior in composition. For the rest we need to distinguish between the following two definitions:

- Local choreography: This type describes the flow from participants view point. It is known as conversational protocols. Local choreographies allow to the public parts of their local processes visible to other participants.
- Global choreography defines the inter-organizational process from a neutral perspective. It has the potential to achieve an agreement between partners.

In fact, the first definition was discussed in the work of [15] and it has been a central element throughout his work. As to the second definition, it represented the starting point of the transformation process presented in the same work. All the same, Eder benefitted well from these two definitions throughout his work [5].

Choreography languages reflect the long-term interactions. They allow the user to describe the peer-to-peer collaboration between services by defining their common observable behavior. In the realm of choreography, two different modeling approaches differ:

Interaction models based approach The elementary interactions, namely demand and the exchange of request-response messages, are the basic elements of this approach. Behavioral dependencies, shown in these interactions and

combinations of interactions, are grouped in complex interactions. According to this type, we can list WS-CDL and Let's Dance:

- **WS-CDL:** its specification is based on an XML document. It defines a common behavior for all participants and models a message between two (or more) participants. It provides more detailed models.
- **Let's Dance:** Its specification is based on graph. It's also based on independent visual notation. The communication action is performed by an actor playing a role. The exchanged messages are message sending and message receipt action.

Interconnection models based approach The control flow is defined by participants. The behavior of each participant and the exchanged messages are represented. However, the models could be incompatible.

- **BPMN (Business Process Modeling Notation):** it is a platform independent language. It uses a typical notation for every process. To make interconnecting different processes, this language uses messages flows. In that way, all interactions are listed with the definition of the control flow between them. Hence, communications are established by using data objects and communicating activities. Nevertheless, BPMN does not support multiple instances of participants. So, to solve this problem, it uses a Pool set to represent multiple participants in one conversation and uses PBD (Participant Behavior Description) as views from the individual partners.
- **BPEL4Chor:** It is a BPEL extension and it is based on "Abstract Process Profile for observable Behavior". BPEL4Chor uses abstract processes and supports all the different choreographies design phases. It ensures communication between participants by using message links. This language is built on three artifact:
 - *Participant Behavior Description:* defines the control flow dependencies between activities of involved participants in the choreography;
 - *Participant Topology:* defines the participants type, their references and the message link which build the choreography structure;
 - *Participant Grounding:* it contains the technical configuration and enables the reuse of choreography specification .

Each of these languages has brought new concepts, and redefined ones already known by others. This created a multitude of concepts, sometimes overlapping and a multitude of ways to manipulate. In fact, concepts, defined by orchestration and choreography languages are interesting, and provide additional skills and knowledge in the world of web services composition. However, sometimes these skills are applicable in a particular area and for special needs that are not always necessary in other areas. In fact, an integrated solution has to address all aspects of developments to facilitate the work of different stakeholders and reduce development time. Yet, the integration of services from heterogeneous sources is still a major challenge facing the developer.

3 Integration of Composition Models

In this section, we briefly mention the research developments that aim to transform choreography to orchestration and those interested in the integration of services composition models.

3.1 Composition of composition models

In [9], the research is specifically focused on processes and construction of processes based on software. The authors study the coordination of applications managed by the process, particularly orchestration and choreography of web services field. They also implement an application for orchestration of web services, using the platform Melusine which they designed themselves, to establish processes software. To set this application, they were inspired by a sample in the language specification WSCI. To illustrate their contributions, they present an application of web services orchestration of only two services. In addition, they define a process meta-model containing a minimum of concepts used to build coordination processes that are understandable and easy to handle for the software developer. We see through this example of services orchestration, they offer an architecture that provides a solution to the shortages of orchestration and choreography languages (many specialized concepts; concepts overlap; close to programming languages) that make the realization of a unifying meta-model utopian. To build a composition software using this approach, it is important that the concepts representing the logic of the application are separated from their implementation, and reified at runtime. Thus, it is possible to use this approach to build a web service composition whenever the software is based on the three tiered architecture or not.

In [4], Clémentine presents two types of orchestration composition. The first one is the composition of two orchestrations that are already the result of a service composition. The second deals with the composition of an orchestration (the result of a set of services composition) and an elementary orchestration, which can be, sometimes, considered as a service. The author claims that some problems may be encountered when carrying out the composition of the orchestrations, which must always be taken into account:

- The orchestrations composition requires the recognition of common elements in order to avoid redundant calls;
- The orchestrations complexity and the multiplicity of data block automatic resolution of conflicts;
- The input parameters composition;
- Workflows obtained must meet certain properties such as the absence of expectations that can lead to errors;
- The order of composition should not change the outcome.

Through his research, Clémentine aspires to compose two orchestrations to get a third. He considers it as a fusion. In order to define a fusion algorithm, he creates a meta-model composition, which includes all the concepts. At first, he identifies the identical elements in order to unify them and will, eventually, call the user in cases of conflict. Towards the end, to identify conflicts and check the compatibility properties between the orchestrations, he relied on formal methods that offer a variety of solutions that can be validated.

In [6], the author presents a new conceptual model for web services choreographies. Through his work, he considers that there is a choreography shared among several partners. In fact, he admits that "*The global choreography captures the core of a business process*"[5]. Therefore, each partner is responsible for the realization of each choreography part. To achieve this, he uses the following definition "*The other choreographies describe parts of the global choreography in the needed detail for implementation*". So, he introduces a new approach called view driven federated choreographies where the latter are judged as federations of choreographies and orchestrations [17]. To achieve his purpose, Eder assumes that choreographies must be realized by orchestrations of partners. In fact, the choreography defines the communication among orchestrations. Besides, he defines a meta-model, which regards the representation of orchestration and choreography described as workflows model. In addition, he considers the inter-layer conformance as a central requirement of the obtained model (federated choreographies). However, he is limited only to structural conformance, temporal conformance, messaging conformance and data flow conformance.

3.2 From choreography to orchestration Models

In [20], the authors discuss the relationship between the two different mechanisms for the service composition in their work. They were the first who introduced the concept of information gap to mention the different levels of details between choreography and execution business process. They present their transformation process in a virtual organization. Although, it is an unidirectional transformation, its implementation detects errors or violated constraints.

Regarding [15], the transformation from CDL specification to BPEL process is presented. In such transformation, the choreography is viewed from two aspects, namely as a global and a local choreography. The global choreography corresponds to the message exchange from a global perspective. It is also considered as a coordination protocol. While local choreography corresponds to the message exchange from a single party perspective. The authors generate local choreography from global ones, which will be used to generate an orchestration process for each part. During the transformation process, the mapping is well documented and explained to generate a BPEL process for all involved parties. Although, it is a bidirectional transformation, the authors do not explicitly mention the relationship between choreography and orchestrations and they ignore some transformation mapping information.

In [10], the author illustrates the basic idea of MDA (Model Driven Architecture)

and model transformations. The objective of this work is to evaluate the feasibility of particularly using the driven models technology based on meta-model transformations to realize the transformation of choreography to a set of orchestrations. Thus, this approach offers a solution; including a methodology that allows (semi-)automatically the transformation process. In addition, throughout this work, he studies the relationship between choreography and orchestration at the architectural level rather than language. In fact, such architectural relationship between the two composition models promotes a transformation, which is independent of all the specification languages for both choreography and orchestration. They formalize the mapping to avoid ambiguous interpretation.

In [14], the authors represent an automated approach of synthesis of orchestrator from choreography. Their choreography specified in BPMN/BPEL is translated to a BPMN/BPEL, which is, also, at the base of orchestrator. In their work, they have chosen BPMN/BPEL choreography, represented in Finite State Machine (FSM), as a base for their formal approach of synthesizing an orchestrator. The correctness verification methods of the orchestrator, detecting and avoiding deadlock situations are included in the synthesizing algorithm, which constructs the orchestrator. The Petri-net based model is then later transformed to BPMN/ BPEL.

3.3 Synthesis

By way of recapitulation, we note that many works have focused on the transformation from choreography to orchestration. To achieve their goals, they essentially focused on the principles of MDA. Furthermore, they ignored the other landings of transformation (processing) and did not try to verify the relation of conformity between both models. In fact, we observe that there was ignorance of other levels either the passage from "local" to "global" choreography or the passage from orchestration to Promela. In addition, they aren't interested in checking the compatibility of the two processes of transformation. Likewise, concerning the composition of service models composition, we believe that little work has dealt with the choreographies composition realm. Indeed, our research is part of that route. In fact, the works, which are most closely related to our work are described in [6], [10] and [14]. Indeed, we will be in continuity, on composition of composition models side, with the work of Eder [6] and, on the other side, with those from Kadhka [10] and McIlvenna [14].

However, to fill the gaps and the shortages encountered by the various works cited above (no compatibility check, no validation processing, one transformation level, not taking into account the service sources heterogeneity and the overlapping concepts ...), we propose an approach, which is based on transformation process from choreography to orchestration. The transition from choreography to orchestration is carried out since the choreography tools do not directly model the majority of activities. In fact, it is a non-executable, abstract process that defines the message exchange protocol and needs details, which will be implemented by the orchestration, to be executed. As a first step, we perform

an integration of a set of heterogeneous choreographies in a global choreography unifying them. At this level, we present a meta-model of choreography. We perform this integration phase to work independently of modeling languages and to treat all services from heterogeneous sources. So, we can include all the different concepts already defined by the different choreography languages. Also, we will avoid the concepts overlap and the redundant calls. This integration will build a starting point for a second phase transformation from choreography to orchestration to finally reach the phase of the transformation validation which can be illustrated by the use of different model-checker.

We have to mention that our contributions are not limited to this, but we choose to undertake audits of properties that will be maintained after each refinement. Hence, this checks if the overall choreography satisfy all the requirements of architecture specifications.

4 Approach

In this section, we present, in detail, our transformation and validation approach. We will scan our work in a vertical manner from top to bottom.

4.1 From heterogeneous local choreography to global choreography

Our work is interested in different levels of transformation process. In fact, the transformations differ in their starting point. As a first step, the process is based on a set of heterogeneous conversational protocols(as shown in Figure 1(a)). This group will provide a choreography, that unifies them, as a result. For this purpose, we will create a meta-model for choreography, which allows us to work independently of modeling languages. At this level, there will be checks to take place. Through them, we will check a set of properties relating to choreography maintained after integration [16] [15] [13] [19]. We note that the global choreography may include different conversational protocols as well as orchestrations, which are considered by some authors as compositions mechanisms from local views [6] .

4.2 From choreography to orchestration

At this level, we distinguish between two types of transformations, either one-to-one or one-to-many. In fact, the orchestration process must be developed in accordance with the choreography. In our research work, we aim to realize a transition from choreography to orchestration (as shown in Figure 1(b)). The resulting orchestration should preserve the overall behavior of the system as specified in the choreography. So it should be in accordance with it and support the responsibility that is already defined. The correctness of the transformation consists in determining if the business goal of an initiator at the choreography level is preserved by the orchestration. During this process of transformation,

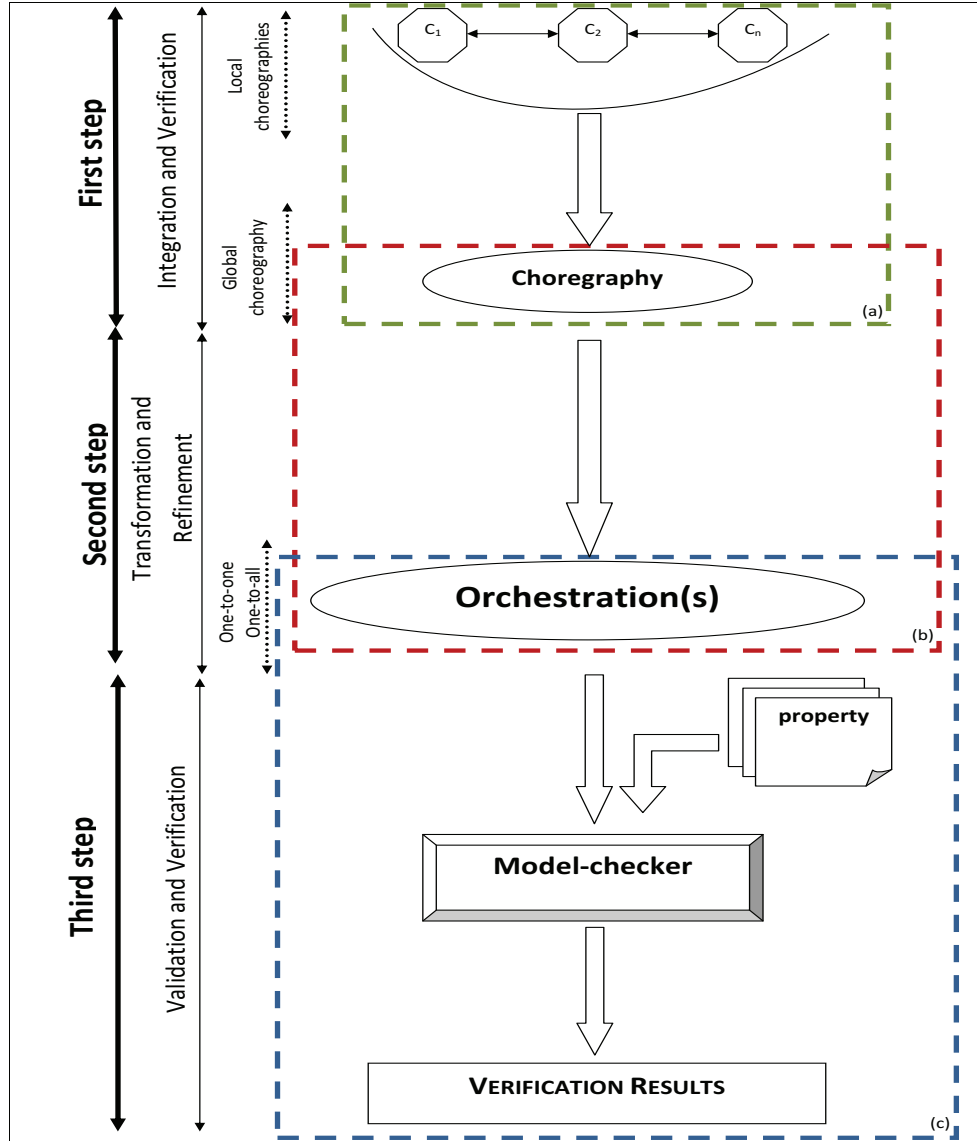


Figure 1: Integration approach of service composition models

some properties should be preserved. Among these properties, the local properties refer to the orchestration properties; while the others, which are common are reference for the properties verified by the choreography mechanism. Our work is subdivided into two essential parts. The first one is devoted to present

the transformation process. However, synthesizing the orchestration from choreography is distributed problem. In fact, at this level, we notice two different types of transformation. One is dedicated for the one-to-one transformation while the other is for the one-to-many. So, the consistency between choreography and orchestration is verified by showing conformance relationship between them [11]. To deal with undecidability of this problem, the choice of the best transformation will be argued in the number of the preserved properties. We opt for the transformation branch which guarantees the highest number of the required properties.

4.3 Validation

Indeed, verification of the properties have decorated each phase of transformation. For the first step, the checks were for choreography properties [2] [16] [15] [13] [19]. Among this properties, we quote:

- **Composability:** The existing choreographies should be able to be combined to form new choreographies that can be reused in different contexts;
- **Asynchronous communication:** Choreography must allow participants to communicate and synchronize their states as well as the information they share;
- **Conformance:** Conformance checking aims at both quantifying of conformance and locating the mismatch, if it exists;
- **Error Handling:** Choreography should support a mechanism for error handling.

Still, this are not the only properties, but we are interested in the second phase of transformation in other audits. These checks include, as mentioned above, properties maintained after the process of integration and properties related to the process of orchestration and that cannot be checked by the choreography.

- **Compatibility of services:** It shows how it is possible to run a consistently different services which use different protocols;
- **Conformity:** It is whether an orchestration satisfies a choreography or no;
- **Error compensation:** An orchestration should support a mechanism for error handling and compensate for later.

Now, the transition from global choreography to orchestration is done, so it is time to validate this transformation, which constitutes the third phase of our process transformation (as shown in Figure 1(c)). In [8], the authors propose a verification approach of web service orchestration. The validation is achieved by using the SPIN model-checker. As a first embodiment, they completed the transition process from BPEL to Promela. In this passage, they test a set of properties related to the orchestration process. In fact, during this step they

also seek to verify the properties that were already maintained at the previous passage. They developed a tool in order to verify non-functional properties of a web services orchestration. Non-functional properties include properties such as performance, reliability, security, robustness and scalability of service. This tool takes a web services orchestration specified with BPEL 2.0, as an input and transforms it to Promela code, the input language of SPIN, where some properties which are expressed in LTL will be checked.

The verification concerns generic (checked for any invoked web services) and specific properties, which are different interdependence relationships between activities within a WS-BPEL process. Among the generic ones, we mention the availability property which means the ability of a service to accomplish a task after an invocation. To this end, we tried to make further contributions to the existing works that were interested in verifying properties related to service composition. In fact, previously, the verification process was centered on the behavioral properties that distinguish themselves through the liveness, safety and reachability properties. Moreover, we are in continuity with the other work. The transformation process from BPEL to Promela is developed with the use of ANTLR.

5 Case Study

A service choreography specifies a communication protocol for all involved partners. In this composition model, the process definition is viewed from a global perspective. Each partner involved in a global choreography has its own parts. To better explain our purpose, we choose the typical scenario "shopping on line" [6], which assumes two (or more) partners involved in one choreography can also participate in another choreography, which is not visible to other partners implied in the first one. Still, we note that the realization of the main goal of this business process is divided between different elementary choreographies. In fact, the responsibility is shared between three various business protocols which are "Purchase Processing", "Shipment Processing choreography" and "Payment processing choreography". In case of shopping online, the different partners involved in this choreography are: the shipper, the client, the bank and the seller. By referring to our example, the seller takes part in several other choreographies, which are not visible to the rest of partners. The seller and the shipper realize another protocol they agreed upon containing other actions like money transfer from seller's bank to the shipper (as shown in Figure 3). In addition, the client and the seller carry out another protocol, which is not visible to the other partners (as shown in Figure 4)

The choreography describes the "Purchase Processing" choreography (as shown in Figure 2). Three partners participate: a Buyer, a Seller and a Shipper collaborate. The Seller while receiving a request quote from the Buyer answers if the requested item is deliverable or not. If the item is deliverable, the buyer places an order, the seller processes the order and puts forward the shipment details to the shipper. The shipper ships the products to the buyer and informs

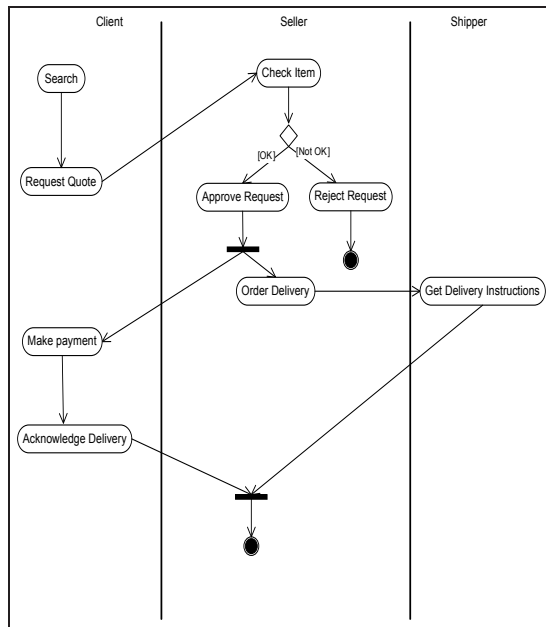


Figure 2: Purchase Processing choreography

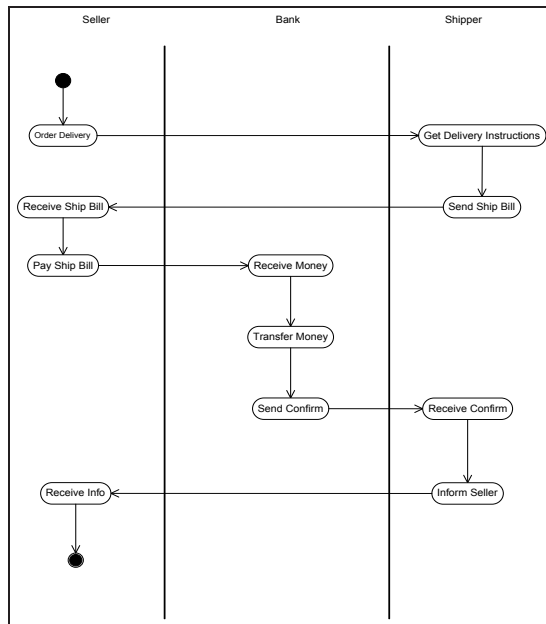


Figure 3: Shipment Processing choreography

the seller about the details. The seller, while receipt of information, sends the bill to the buyer. When the buyer has received both the bill and the ordered

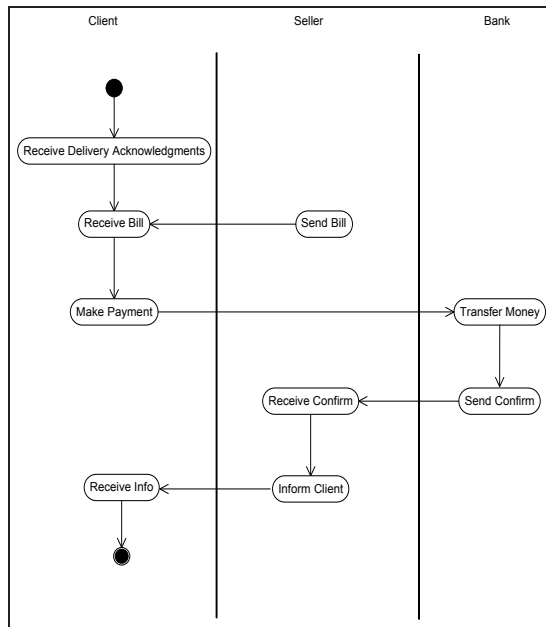


Figure 4: Payment Processing choreography

goods, makes the payment to the seller. The process ends when receipt of the payment by the seller. This choreography involves interactions between the Buyer and the Seller, the Seller and the Shipper and between the Buyer and the Shipper. Each of these interactions can be represented as a separate choreography. These choreographies may be combined by means of composition where existing choreography definitions can be reused and recursively combined into more complex choreographies. Choreographies just describe abstract processes, whereas an orchestration describes an executable process run by all the partners. In the above example, the different choreographies overlap in some parts. To avoid this problem, we propose a new approach where existing business protocols can be integrated into one choreography to reach the goal of the business process. The fruit of this integration process presents the coordination of the different conversational protocols realized by the different partners (as shown in Figure 5).

The business protocols will be integrated into a more complex choreography. The idea of integration process is presented in 5. At first, we consider a choreography divided between different partners. For example, Purchase processing choreography shared between Buyer, Seller, and Shipper. A business protocol may support another one. This means the former contributes to the latter and partially elaborates it. So, we aim to integrate the different elementary choreographies in a more complex choreography. In order to ensure the compatibility of the system, a conformance test between the different business protocols is necessary. Our metamodel enables the representation of choreographies. In fact, we

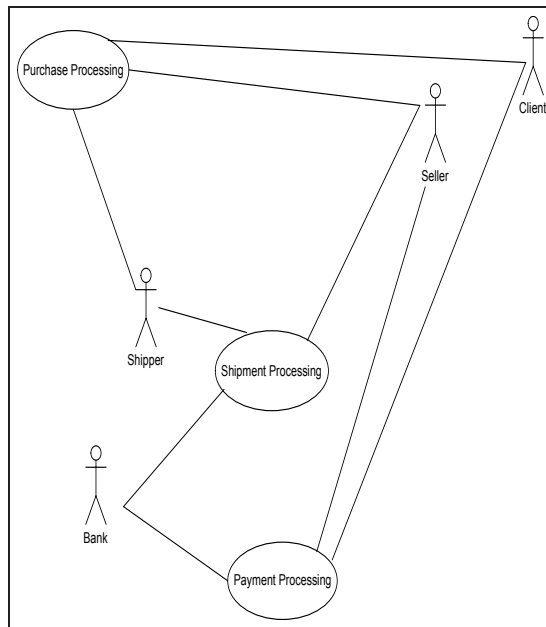


Figure 5: Choreography Example

opt to use our own metamodel which will lead to describe several choreographies on different levels of detail.

6 Conclusion and future work

In this paper, we presented our approach, which consists in the transformation from choreography to orchestration. This transformation process is based on a set of heterogeneous local choreography, which will be integrated into a global choreography. Through this work, we verify in each step a set of properties which are preserved. At first, these properties refer to the ones verified by choreography specification. Second, we notice two types of properties, local properties (reference for choreography ones) and the others are common (reference for orchestration ones). At last, a verification process will occur by the use of model-checker, which will depend on the properties type.

References

- [1] S. Aissi, P. Malu, and K. Srinivasan. E-business process modeling: the next big step. *IEEE Computer society*, 35:55–62, 2002.
- [2] Matteo Baldoni, Cristina Baroglio, Alberto Martelli, Viviana Patti, and Claudio Schifanella. Goal preservation by choreography-driven matchmaking. In *ICSOC Workshops*, 2007.

- [3] S. Chardigny. *Extraction d'une architecture logicielle à base de composants depuis un système orienté objet : Une approche par exploration*. PhD thesis, Thèse de doctorat : Université de Nantes, Laboratoire d'informatique de Nantes Atlantique, France, 2009.
- [4] N. Clementine. Vers la composition d'orchestrations de services. Master's thesis, Université de Nice Sophia-Antipolis, France, 2006.
- [5] Johann Eder, Nico Kerschbaumer, Julius Köpke, Horst Pichler, and Amirreza Tahamtan. View-based interorganizational workflows. *CompSysTech*, pages 1–10, 2011.
- [6] Johann Eder, Marek Lehmann, and Amirreza Tahamtan. Choreographies as federations of choreographies and orchestrations. In *ER (Workshops) p (183-192)*, 2006.
- [7] N. Guermouche. *Etude Des Interactions Temporisées dans la Composition de Services Web*. PhD thesis, Doctorat de l'université Henri Poincaré - Nancy 1, France, 2010.
- [8] H. Hadj-Kacem, W. Sellami, and A. Hadj-Kacem. A verification approach for non-functional properties in web service orchestration. In *21st IEEE International Conference on Collaboration Technologies and Infrastructures (WETICE'2012), Toulouse, France*, 2012.
- [9] S. Jamal. Environnement de procédé extensible pour l'orchestration - application aux services web. Master's thesis, Thèse de doctorat : Université Joseph Fourier de Grenoble, France, December 2005.
- [10] R. Khadka. Model-driven development of service compositions: Transformation from service choreography to service orchestrations. Master's thesis, University of Twente, Enschede, The Netherlands, 2010.
- [11] Kais Klai, Samir Tata, and Jörg Desel. Symbolic abstraction and deadlock-freeness verification of inter-enterprise processes. In *BPM*, 2009.
- [12] O. Kopp and F. Leymann. Choreography design using ws-bpel. *IEEE Data Eng. Bull.*, 31(3):31–34, 2008.
- [13] Niels Lohmann. Correcting deadlocking service choreographies using a simulation-based graph edit distance. In *BPM*, 2008.
- [14] S. McIlvenna, M. Dumas, and M.T. Wynn. Synthesis of orchestrators from service choreographies. *Proceedings of The Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009). Wellington, New Zealand, Australian Computer Society*, 96:129–138, 2009.
- [15] J. Mendling and M. Hafner. From ws-cdl choreography to bpel process orchestration. *Journal of Enterprise Information Management (JEIM)*, pages 506–515, 2008.

- [16] Jan Mendling and Michael Hafner. From inter-organizational workflows to process execution: Generating bpm from ws-cdl. In *OTM Workshops*, 2005.
- [17] Amirreza Tahamtan and Johann Eder. View driven federation of choreographies. In *Advances in Intelligent Information and Database Systems*, 2010.
- [18] L. Touseau. Accords de niveau de service dans les plateformes dynamiques à services. Master's thesis, Master 2 Recherche Systèmes et Logiciels, Université Joseph Fourier (Grenoble 1), France, 2005.
- [19] Wil M. P. van der Aalst, Marlon Dumas, Chun Ouyang, Anne Rozinat, and Eric Verbeek. Conformance checking of service behavior. *ACM Trans. Internet Techn*, 2008.
- [20] I. Weber, J. Haller, and J.A. Mülle. Automated derivation of executable business processes from choreographies in virtual organisations. *International Journal of Business Process Integration and Management*, 3(2):85–95, 2008.