

## Towards a Unified Formal Model for Service Orchestration and Choreography

Diana Allam, Hervé Grall, Jean-Claude Royer

► **To cite this version:**

Diana Allam, Hervé Grall, Jean-Claude Royer. Towards a Unified Formal Model for Service Orchestration and Choreography. Yves Ledru, Anne-Françoise Le Meur, Olivier Caron. Actes des troisièmes journées nationales du Groupement De Recherche CNRS du Génie de la Programmation et du Logiciel (Poster), Jun 2011, Lille, France. 2011. <hal-00700737>

**HAL Id: hal-00700737**

**<https://hal.inria.fr/hal-00700737>**

Submitted on 23 May 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.

# Towards a unified formal model for service orchestration and choreography

Diana Allam - PhD Student  
Supervised by: Herve Grall and Jean-Claude Royer

ASCOLA group; EMN-INRIA, LINA  
4 rue Alfred Kastler, 44307 NANTES Cedex 3, France  
{diana.allam, herve.grall, jean-claude.royer}@emn.fr

The growth of Internet has extended the scope of software applications, leading to network-based architectures. The main characteristic of these architectures is that they restrict the communication between remote components to message passing. Service-oriented computing is a solution to organise the exchange of messages in a network-based architecture, by using services as primitive components. Thus, each component can be a client, a server or both. Since a service-oriented application typically spans a number of different organizations, its executions is subject to stringent security requirements. That is the reason why the partners involved generally define a contract at the global level in order to enforce some security policy. From the contract, each partner deduces by projection a specification of the security functionalities that it must locally implement. Of course, in order to be useful, all these projections must ensure that the local functionalities effectively collaborate to realize the global contract.

The ANR project CESSA <sup>1</sup> essentially aims at tackling this problem: in the context of service-oriented computing, it will provide means to locally enforce a security policy globally defined, by resorting to aspect-oriented programming. The core of the project is a formal model. Generalizing the top-down approach described above for security, the formal model is based on two stages.

**Description stage** Aiming at expressing global properties and their local projections, it will lead to a choreography language and an interface language for local processes.

**Realization stage** Aiming at expressing local processes and their collaboration, it will lead to a process orchestration language and a collaboration infrastructure.

The stages are linked, since the formal model must satisfy the fundamental projection property: given a choreography, if each local process realizes the corresponding projection of the choreography, then their collaboration realizes the choreography.

In this abstract, we will focus on the realization stage, already achieved. We will also shortly describe some future steps for fulfilling the description stage and for linking the two stages.

**Context: choreography and orchestration of web services** A choreography defines a contract between partners that collaborate by exchanging messages: it corresponds to the protocol used to exchange messages. Locally, each partner controls processes that invoke or provide services. At the process level, an orchestration describes the interactions of one service with other services. Contrary to the choreography where no central coordinator exists, an orchestration makes explicit a central coordinator which is responsible for invoking and combining the services realizing the composition.

There are languages not only for the orchestration of web services like the standard business process execution language (BPEL), but also for choreography like the web services choreography description language (WS-CDL) [7]. There are also languages that offer the possibility of modeling these two levels like the standard business process model and notation (BPMN 2.0).

The formal model that we now present can be considered as an abstraction of BPEL and of BPMN 2.0.

**Collaboration infrastructure** A collaboration involves a set of processes that execute in a concurrent way and interact each other via message exchanges in a completely asynchronous way. Each process has an address, like a uniform resource locator. When a process executes, it can receive messages from other

---

<sup>1</sup> CESSA (Compositional Evolution of Secure Services using Aspects) is an ANR project in partnership with SAP labs France, EURECOM and IS2T, <http://cessa.gforge.inria.fr>

processes that knows its address and send messages to processes the address of which it knows. The topology of the collaboration network can evolve as addresses can be exchanged. For example, in a simple client-server interaction, the client can send a request because he knows the server address. The server's knowledge of the network increases at the reception of the request message which contains the client location. By this way, the server can reply the client while he did not know its address before the request.

**Process orchestration language** A wide variety of formal models of web services exists for the orchestration level. Amongst the candidate models we consider [1,5,6]. These models are essentially based on the  $\pi$ -calculus, since this process calculus allows channels to be communicated, a fundamental feature as seen above and in [2]. [5] introduces an extension of the  $\pi$ -calculus with a notion of transaction. Viroli [6] introduces also a  $\pi$ -calculus based semantics but focusing on correlation sets. It details the session management in BPEL. Another notable work is the one from Abouzaid and Mullins [1]. The author proposes a formal BP-language close to the  $\pi$ -calculus and inspired from the work of Lucchi and Mazzara for handlers and of Viroli for correlation sets. Following and extending these studies, we also propose a process language as an extension of the  $\pi$ -calculus. The three most important differences between our model and the  $\pi$ -calculus are (i) the language has state variables and expressions to denote the values assigned to the variables while in the  $\pi$ -calculus, there are no state variables (but they can be encoded) (ii) whereas a channel is represented in our process model by an ordered pair, a location and a name for the message, there is no way to define a new location, contrary to the  $\pi$ -calculus, and (iii) channels are completely asynchronous to be closer to real network (it is not the case for synchronous channels as they are defined in the  $\pi$ -calculus). One more important contribution in our model is the use of correlation sets in a more general form than the one defined in BPEL or in its formalization. Correlation allows a process to hold conversations with partners involved in a collaboration. It is equivalent to the creation of sessions to maintain a connection between two partners. In the standard semantics of the  $\pi$ -calculus, when different processes are waiting for an incoming message, the process fired is randomly chosen. With correlation, it is the process whose state is correlated with the incoming message.

**Extensions and future work** As an extension of the formal model, we expect to add exceptions and handlers so to have a common abstraction of what is defined in BPEL and BPMN. Our goal is to reduce the complexity and the ambiguity that we see in the BPEL standard and to have a more understandable and simpler model closer to BPMN 2.0. Beyond the extension, we will complete the formal model by the description stage. We aim to follow the top-down approach, from the global choreography to the local orchestration. Starting from a choreography described as collaboration types, inspired from session types [3], we will deduce its projection over each process involved. Then each partner will develop its processes realizing the projections, in order to finally produce a collaboration realizing the global choreography. We are more particularly interested in projection theories [4], specially for analyzing and enforcing some security properties. We aim to extend these theories to other interaction modes, not only to the synchronous mode. At the present, there is no study of the purely asynchronous case for collaborations, as they are defined in our framework.

## References

1. Faisal Abouzaid and John Mullins. Formal specification of correlation in WS orchestrations using BP-calculus. *Electr. Notes Theor. Comput. Sci.*, pages 3–24, 2010.
2. Michele Boreale and al. SCC: A service centered calculus. In *Web Services and Formal Methods, Third International Workshop, WS-FM, Proceedings*, Lecture Notes in Computer Science, pages 38–57. Springer, 2006.
3. Marco Carbone, Kohei Honda, and Nobuko Yoshida. A calculus of global interaction based on session types. *Electronical Notes in Theoretical Computer Science*, 171:127–151, June 2007.
4. Marco Carbone, Kohei Honda, and Nobuko Yoshida. Structured communication-centred programming for web services. pages 2–17. Springer, 2007.
5. Roberto Lucchi and Manuel Mazzara. A pi-calculus based semantics for WS-BPEL. *J. Log. Algebr. Program*, pages 96–118, 2007.
6. Mirko Viroli. A core calculus for correlation in orchestration languages. *J. Log. Algebr. Program*, 2007.
7. Michael zur Muehlen, Jeffrey V. Nickerson, and Keith D. Swenson. Developing web services choreography standards - the case of REST vs. SOAP. *Decision Support Systems*, 40:9–29, 2005.