

# BPMN 2.0 Choreography Language: interface or business contract?

Greta ADAMO<sup>a,c</sup> and Stefano BORGIO<sup>b</sup> and Chiara DI FRANCESCOMARINO<sup>a</sup> and  
Chiara GHIDINI<sup>a</sup> and Marco ROSPOCHER<sup>a</sup>

<sup>a</sup> *FBK-IRST, Via Sommarive 18, 38050 Trento, Italy*

<sup>b</sup> *ISTC-CNR Laboratory for Applied Ontology, Trento, Italy*

<sup>c</sup> *University of Genova, DIBRIS, via Dodecaneso 35, 16146 Italy*

**Abstract.** Choreography diagrams have been introduced in the Business Process Model and Notation language 2.0 (BPMN 2.0), one among the most used languages for modelling and analyzing business processes in industry, in order to provide a view on the *interaction* between participants. Besides the intuitive definition of choreographies as *interfaces* among participants, the BPMN 2.0 specifications also define choreographies as *business contracts* among the parties. However, the adoption and the diffusion of the business contract nature of choreography diagrams seem to be hindered by the underspecification of the notation, which does not allow to model and formalize constraints and relationships among choreography entities, which would need to be specified in a business contract. In this paper we provide a preliminary investigation of some of the open issues characterizing BPMN 2.0 choreography diagrams when looking at the business contract nature of the notation, by focusing on those related to messages and participants.

## 1. Introduction

The Business Process Modelling Notation<sup>1</sup> (BPMN) is one of the most popular languages for business process modelling largely used in industry. Few works in the literature [9,6,10] focused on the ontological formalization of the language. For instance, [9] proposed the “BPMN Ontology” for formalizing *business process diagrams* of the BPMN 1.1 specifications. A similar approach is followed in [6] to formalize the structure of process diagrams expressed according to BPMN 2.0. A different approach is taken in [10], where the authors focus on the analysis of behavioral aspects of process models (activities and events) in order to investigate whether these constructs of BPMN commit to an ontological theory of the domain entities at hand. However, so far no initiatives have considered the ontological formalization of the additional typologies of diagrams introduced with the BPMN 2.0 specifications, namely *collaborations* and *choreographies*. Differently than process diagrams, collaborations focus on representing the interactions between two or more processes, while choreographies enable capturing the coordination between different business participants. In the quest for providing an ontological formalization of the whole BPMN specifications, we started tackling the analysis of choreography diagrams.

---

<sup>1</sup><http://www.omg.org/spec/BPMN>

The BPMN 2.0 specifications [7] provide several *definitions* of choreographies, each one capturing a different aspect. A first definition of choreography is “the way business participants *coordinate their interactions*” ([7, p. 345]). Starting from this assumption, it seems that the purpose of a choreography is to hook the *interaction* behavior between two or more participants, thus “making possible to derive the process interfaces of each partner”([7, p. 345]). In addition, the interaction is focused on the exchange of *information* conceived as *messages*. Another definition considers choreographies as “a type of *business contract* between two or more organizations” ([7, p. 345]).

Over the past years, different works have proposed languages and evaluation models for extending choreographies with business contract (deontic) constructs [1] and for capturing and assessing the value of choreographies [2,3]. Choreographies in BPMN 2.0 seem not to satisfy models as the Web Service Choreography Description Language proposal (WS-CDL [8]) and, at least not completely, the main evaluation model for the adequacy of choreographies based on a semiotic quality framework [11]. The conclusion of the evaluation is, indeed, that BPMN 2.0 choreographies present some issues [2].

In the attempt to better understand choreography diagrams for their formalization, we also clashed against a mismatch between the business contract nature of the BPMN 2.0 choreography language and the underspecification of the language when coming to define constraints and relationships at a business contract level.

In this preliminary work we focus on a few ontological elements (participants, roles, information) to highlight and discuss, by means of an example, some of the constraints and the relationships which a business contract could require to model, but that the language does not allow to explicitly represent. In detail, we focus on:

- message-related issues: how to constrain a message and its content?
- participant-related issues: how to specify relationships among participants?

By analyzing these issues we advocate the need of a finer-grained specification of the language able to capture these aspects or, alternatively, the adoption of semantic annotation mechanisms enabling the formalization of constraints and relationships among choreography entities.

After a brief introduction of the main BPMN 2.0 choreography constructs (Section 2), we illustrate through an example the issues of the language in formalizing contract level constraints and relationships (Section 3). Finally, conclusions and future work are reported in Section 4.

## 2. The BPMN 2.0 choreography modeling

The BPMN 2.0, which is the *de-facto* standard notation for business processes proposed by the Object Management Group (OMG), captures the coordination between different business participants through *choreography diagrams*.

Figure 1 shows an example of a choreography diagram in BPMN 2.0. The interaction behaviour among two or more participants is described by means of *choreography activities* i.e., *choreography tasks* (atomic activities) or *sub-choreographies* (compound activities), connected via *sequence flows*. Choreography activities are depicted as rectangles (decorated with a “+” symbol in case of sub-choreographies), while participants as bands on their top and bottom. For instance, the choreography diagram in Figure 1

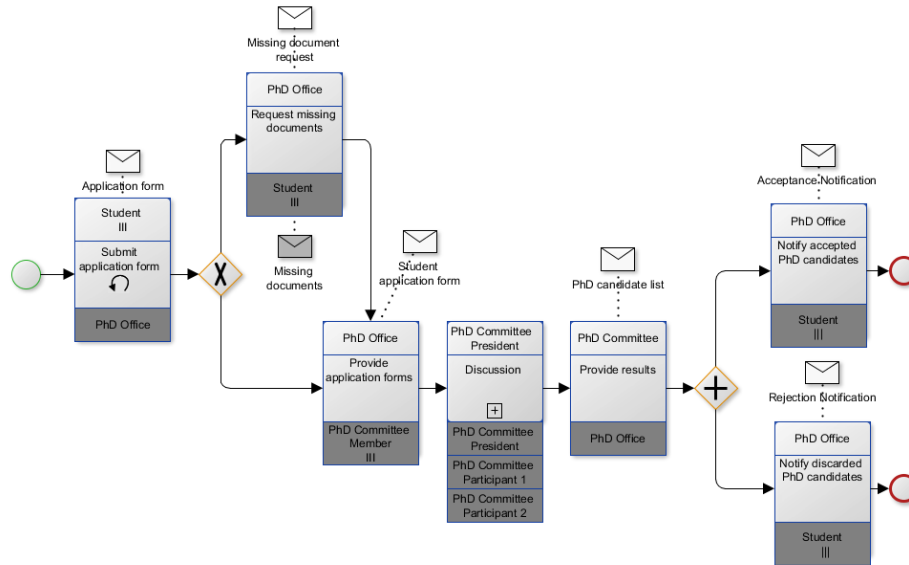


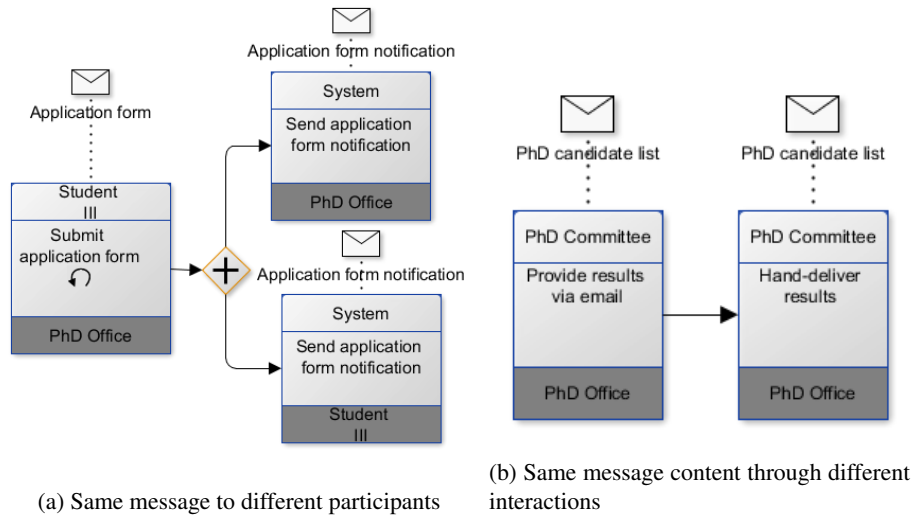
Figure 1. Choreography example: the PhD selection process

is modelled involving seven participants: the *student*, the *PhD Office*, the *PhD Committee*, the generic *PhD Committee Member* and the single members of the committee (*PhD Committee Participant 1*, *PhD Committee Participant 2* and the *PhD Committee President*).

Each choreography activity has only one initiator (depicted in white) sending the initial message and one or more receivers (darker bands associated to the activity). An envelope represents a message sent by the sender while return message envelopes of a two-way interaction are darkened. For instance, the atomic choreography task *Provide results* represents the interaction between the *PhD Committee* (sender) and the *PhD Office* (receiver): the *PhD candidate list* is sent from the *PhD Committee* to the *PhD Office*.

As for the control flow, the BPMN 2.0 choreography language inherits the gateways of the BPMN 2.0 language: the XOR, OR and AND gateways. For example, in Figure 1, the first XOR gateway models a decision point so that, if the application forms sent by the students are complete, they are directly passed to the PhD Committee; otherwise, the students are asked to send the office the missing documentation.

Moreover, BPMN 2.0 choreography diagrams allow for representing more than two participants (when dealing with sub-choreographies) and *multi-instance* participants. An example of choreography activity with several participants is the *Discussion* sub-choreography in Figure 1. This choreography activity involves three participants (one as initiator and three as receivers of the message). *Multi-instance* participation is denoted by three vertical lines and can refer both to the choreography activity and the participants. For the sake of this paper we provide here only an example of *multi-instance* participant: *Student* is used at the start of the diagram of Figure 1 for representing several instances of the role ‘Student’ (as several students can apply for PhD). Note that, in this case, the *PhD Office* will loop on the receipt of the application forms by each of the students



**Figure 2.** Message-related issues

until the PhD call deadline is over. This is denoted by the looping arrow symbol in the choreography activity *Submit application form*.

### 3. Open issues

In this section we describe a scenario we are interested to model as a choreography diagram and we use it for discussing the aspects of the BPMN 2.0 choreography language that could potentially hinder the interpretation of a choreography as a business contract.

*Ph.D Selection Scenario* The *Ph.D selection* scenario deals with the application, evaluation and applicants' notification in a PhD selection. In detail, students apply for a PhD through an online application form, composed of three parts: personal information details, CV, and motivation letter. The online application form is filled and submitted by the student to the PhD Office and to the student herself for notification. If documents are missing, the PhD Office requests them to the students. Once all the needed documents are available, the PhD Office sends the received application forms to the PhD Committee. The committee members discuss about each of the applicants and, at the end, the committee returns the final list of accepted candidates to the PhD Office both via e-mail and through hand-delivery. The PhD Office will finally notify both the accepted and the rejected PhD candidate applicants.

Figure 1 shows a choreography diagram modelling the PhD Selection Scenario at the level of abstraction that is generally used with choreography diagrams. However, by keeping in mind the definition of a choreography as a business contract and looking at the description of the scenario, we would like to be able to specify and formalize other constraints and relationships among choreography entities. We detail in the following the modelling issues we found in the scenario when detailing messages and participants, respectively.

*Message-related issues* For instance, we would like to specify that the application form is composed of three parts: the personal details, the CV and the motivation letter. As only a single message (per direction) can be exchanged in a choreography task, the only viable solution for expressing this kind of information in BPMN 2.0 choreographies seems to be the Message attribute - *item description* - which allows for specifying the structure of the exchanged message, i.e., the *form* of the message. However such attribute provides only an informal means to specify the structure of the document.

If the form of the message can be, at least informally, specified, this is not possible at all for the (expected) *content of a message*. For instance, if we wish to explicitly model the fact that the applicant student receives, as submission confirmation, the submitted form, we would probably need to explicitly model a new participant - the *system* - sending an *Application form notification* not only to the PhD Office, but also to the students themselves. In other terms we would need to model the *sending of the very same message to two different participants*. Unfortunately, however, BPMN 2.0 choreography specifications limit the number of participants of a choreography task to two. We hence need to resort to graphically modelling the notification messages from the system to the *PhD Office* and to the *Student* as different messages (and choreography tasks), only implicitly bound by the same name, as shown in Figure 2a. However, this does not enable us to formally specify that the two *Application form notification* messages are exactly the same message, as we wish.

A similar, although slightly different situation, is related to the inability of the language to provide a mechanism for *sending the same message content through different participant interactions*. For instance, in the selection scenario, we would like to model the fact that the committee returns to the PhD Office the list of PhD candidates both via email and through hand-delivery. To this aim we would like to specify that a message with the same content (the list of PhD candidates) is delivered twice to the PhD Office - first via email and then through hand-delivery (see Figure 2b). The BPMN 2.0 choreography language, however, does not allow us to specify that the content of a message is exactly the same of another.

In both the last two examples, we can observe that the same message or message content, respectively, can have different effects on the participant(s) receiving it. Instead of focusing on a general message content, we can capture this difference by separating the generic content in *information* and *knowledge*. Roughly speaking, here we use the term *information* to refer to the meaning of a message in isolation. That is, when the message is seen as a (possibly complex) statement taken out of any particular context. For example, consider a message composed of the statement “snow is white”. This message conveys the information that the object snow has color white. The fact that snow is an object and white is a value of the color-quality is ensured by the language itself. Thus, information is here seen as the ‘linguistic’ meaning of the statement. The same message has a different meaning, which here we call *knowledge*, if we consider the meaning of the statement in a specific context. Briefly, in this paper the knowledge of a message is the change in the epistemic state of the receiver as caused by the acquisition of the message information in the context provided by the choreography model. The “snow is white” message has a different impact on the receiver depending on whether this is represented within a choreography model relative to an agent that is moving from, say, Cuba to Norway, compared to the impact it has on an agent that lives in Norway and is learning English. In the first case this sentence conveys new knowledge about the object

snow; in the second case it conveys new knowledge about the English grammar or about the meaning of some English terms.

In the first example of the selection scenario - *sending the very same message to two different participants* - there is only one message (and thus one message information) and yet the knowledge it conveys to the PhD Office and to the student is different. Indeed the PhD Office acquires knowledge on the fact that an application form has been submitted and a new student has applied for PhD. The student, on the contrary, acquires knowledge on the fact that the application form has been successfully received by the specific University. In the second example - *sending the same message information through different participant interactions* - there are two message instances with the same message information sent to the same participant and still the knowledge they convey differs: while the first message conveys knowledge to the PhD Office about the outcome of the evaluation, the second message provides no knowledge as the PhD Office's context at this point has already been updated on this issue.<sup>2</sup>

This discussion suggests a further issue of the choreography language: lack of guidelines on the usage of message labels. The use of message labels seems to be a natural means for informally constraining the messages' information. When the language already provides a mechanism to specify that two messages have the same information, the label of the message could be used for providing information about knowledge. For instance, in the first case we could label *confirmation receipt* the application form notification received by the student and *submission request* the very same message received by the PhD Office. Similarly, in the second example, the PhD Office could have received first a *PhD candidate list* and then a *Copy of the PhD candidate list*, where here the term 'copy' specifies that no new knowledge is added to the PhD Office at this point.

*Participant-related issues* A second group of issues is instead related to the participants and their relationships. In the choreography diagram in Figure 1, for instance, the *PhD Committee* and its members have been modelled in different ways: as a single entity (for representing that a unique message is sent by the committee to the PhD Office), as a multi-instance participant (for denoting that each member of the committee received the student application forms by the PhD Office), or by means of different roles (for specifying the role of each of the members of the committee in the discussion). However, these participants are completely unrelated in the choreography diagram, as BPMN does not allow to specify any relationship between participants, while we would like to be able to state and specify such relationships.

On the other hand, we are aware of the potential complexity introduced by allowing to specify these relationships, not only in terms of choreography diagram but also in terms of process diagram. Indeed, since each participant could implement her process in her own pool, reconciling the processes of the *PhD Committee* and a generic *PhD Committee Member*, or even worse the process of a generic *PhD Committee Member* with the one of a specific *PhD Committee Participant 1*, could be far from trivial.

---

<sup>2</sup>Note that the knowledge acquired by the PhD Office on the fact that these activities have been executed due to the reception of the two messages is not knowledge obtained from the information in the messages themselves but from the fact that they (both) exist.

#### 4. Conclusions and Future Work

The preliminary analysis carried out on the BPMN 2.0 choreography language revealed that the language presents some issues with respect to its capability to describe constraints and relationships among its entities. These issues open the way to the possibility to enrich the language with the appropriate means for expressing properties on message form and content, as well as participant relationships. An alternative possibility could be enriching the diagram with semantic annotations, as already proposed for the BPMN process diagram [4], in order to be able to formalize and specify constraints on messages and relations among participants.

In the future, we plan, besides proceeding with the ontology-based choreography diagram formalization, to further investigate these open issues and to suggest possible solutions to support and guide users in the exploitation of choreographies as a means for specifying business contracts among parties.

*Acknowledgments* This research has been partially carried out within the Euregio IPN12 KAOS, which is funded by the “European Region Tyrol-South Tyrol-Trentino” (EGTC) under the first call for basic research projects.

#### References

- [1] Andrew Berry and Zoran Milosevic. Extending choreography with business contract constraints. *International Journal of Cooperative Information Systems*, 14(02n03):131–179, 2005.
- [2] Mario Cortes-Cornax, Sophie Dupuy-Chessa, Dominique Rieu, and Marlon Dumas. *Evaluating Choreographies in BPMN 2.0 Using an Extended Quality Framework*, pages 103–117. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [3] Gero Decker, Oliver Kopp, Frank Leymann, and Mathias Weske. Bpel4chor: Extending BPEL for modeling choreographies. In *2007 IEEE International Conference on Web Services (ICWS 2007), July 9-13, 2007, Salt Lake City, Utah, USA*, pages 296–303, 2007.
- [4] Chiara Di Francescomarino, Chiara Ghidini, Marco Rospocher, Luciano Serafini, and Paolo Tonella. A framework for the collaborative specification of semantically annotated business processes. *Journal of Software Maintenance*, 23(4):261–295, 2011.
- [5] Pawel Garbacz and Oliver Kutz, editors. *Formal Ontology in Information Systems - Proceedings of the Eighth International Conference, FOIS 2014, September, 22-25, 2014, Rio de Janeiro, Brazil*, volume 267 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2014.
- [6] Christine Natschläger. *Towards a BPMN 2.0 Ontology*, pages 1–15. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [7] OMG. Business Process Model and Notation (BPMN), Version 2.0, January 2011.
- [8] Zongyan Qiu, Xiangpeng Zhao, Chao Cai, and Hongli Yang. Towards the theoretical foundation of choreography. In *Proceedings of the 16th International Conference on World Wide Web, WWW '07*, pages 973–982, New York, NY, USA, 2007. ACM.
- [9] Marco Rospocher, Chiara Ghidini, and Luciano Serafini. An ontology for the business process modelling notation. In Garbacz and Kutz [5], pages 133–146.
- [10] Emilio M. Sanfilippo, Stefano Borgo, and Claudio Masolo. Events and activities: Is there an ontology behind bpmn? In Garbacz and Kutz [5], pages 147–156.
- [11] Terje Wahl and Guttorm Sindre. An analytical evaluation of bpmn using a semiotic quality framework. *Advanced topics in database research*, 5:94–105, 2006.