Softw Syst Model (2016) 15:219–255
DOI 10.1007/s10270-014-0398-0

REGULAR PAPER

Evaluating the appropriateness of the BPMN 2.0 standard for modeling service choreographies: using an extended quality framework

Mario Cortes-Cornax · Sophie Dupuy-Chessa · Dominique Rieu · Nadine Mandran

Received: 26 June 2012 / Revised: 2 December 2013 / Accepted: 24 January 2014 / Published online: 1 March 2014

Abstract The concept of choreography has emerged over the past years as a foundational concept for capturing and managing collaborative business processes. The latest version of the Business Process Modeling Notation (BPMN 2.0) adopts choreography as a first-class citizen. However, it remains an open question whether BPMN 2.0 is actually appropriate for capturing this concept. In this paper, we extend an existing language evaluation framework in order to evaluate the support for choreographies in BPMN 2.0. The framework provides a means of identifying the strengths and weaknesses of BPMN 2.0 for choreographies. We also give potential solutions that may be taken into account in future extensions or improvements to BPMN 2.0.

Keywords Service choreographies · Language quality framework · BPMN 2.0 · Evaluation

1 Introduction

1.1 Problem statement

Today, organizations trend to work together to achieve common goals. Therefore, collaborative business processes between different organizations are enacted. Processes coordination becomes critical during the collaboration and underlines the need for communication rules and information exchanges that can be captured by choreography languages.

A *choreography* is an overall representation of the interactions between multiple organizations or organizational units involved in a common business process [46]. Choreographies provide analysts with a basis for understanding, analyzing and optimizing cross-organizational business processes. The choreography can also be seen as a "business contract" between several organizations. Figure 1 shows an example of a choreography where the interactions between several participants in a manufacturing process are represented. In addition to a graphical representation, many technical constraints have to be taken into account to define the choreography precisely so that each individual participant can develop his individual process without ambiguity and therefore achieve interoperability with the other participants. It is clear that there is a need for choreography languages that support technical configuration [17].

Over the past years, several research projects have proposed different languages for capturing choreographies such as Let's Dance [61], BPEL4Chor [17] or multi-agent protocols (MAP) [2]. Popular languages such as the Message Sequence Charts (MSC) [29] have also been proposed to capture cross-organizational interactions. W3C's efforts in the context of the Web Service Choreography Description Language (WS-CDL [56]) also presented a choreography language. Even if WS-CDL is one of the most popular languages in the choreography field, major deficiencies were detected by Barros et al. [6].

Industry initiatives such as RosettaNet[1] are aimed at standardizing choreographies in the value chain domain. How-

Communicated by Prof. Robert France.

M. Cortes-Cornax (✉) · S. Dupuy-Chessa · D. Rieu
SIGMA Team, Laboratoire d'Informatique de Grenoble (LIG),
Université Grenoble-Alpes, 220 Rue de la Chimie, BP 53,
38041 Grenoble Cedex 9, France
e-mail: Mario.Cortes-Cornax@imag.fr

N. Mandran
PIMLIG Team, Laboratoire d'Informatique de Grenoble (LIG),
Université Grenoble-Alpes, 220 Rue de la Chimie, BP 53,
38041 Grenoble Cedex 9, France

[1] http://www.rosettanet.org/.

ever, these initiatives mostly involve textual descriptions of the overall choreographies and are focused on providing a detailed message format description [17]. Moreover, they are usually restricted to a particular domain.

In early 2011, the Object Management Group [44] released the latest version of the Business Process Model and Notation language (BPMN version 2.0 [45]), which includes ideas from the aforementioned choreography languages WS-CDL, BPEL4Chor and Let's Dance [7]. In previous versions of BPMN, the only way to represent choreographies was via *collaboration diagrams*. In this new version, the *choreography diagram* and the *conversation diagram* were introduced in addition to the collaboration diagram. Collaborations give the participants' view, whereas choreography diagrams represent an overall view of the interaction sequence. Conversation diagrams represent a "bird's eye" perspective of collaborations abstracting several sequence flows in conversation nodes. These three diagrams will be considered in the context of modeling choreographies in BPMN 2.0. These extensions for choreographies rely on the previous BPMN version diagrams, which are widely used in the industry. These facts make this new version of BPMN 2.0 extremely important not just for modeling individual processes or collaborations but also for capturing choreographies.

To help standardization in the choreography domain, a complete set of requirements in this area is needed together with the evaluation of BPMN 2.0, which is major candidate for becoming a choreography standard.

1.2 Contribution

The main goal of this paper is:

- To *evaluate the adequacy of the constructs for choreography modeling introduced in BPMN 2.0*. A critical overview of the choreography proposal is given.

Other contributions of this paper are the following:

- The *identification of limitations and areas for improvements* in the language. *Potential solutions* are proposed that may be taken into account in future extensions or improvements to BPMN 2.0.
- The catalog of requirements identified to perform the evaluation that could be used as a starting point for further (empirical) choreography evaluations. It also represents a clear panorama of possible *criteria for evaluating the quality of different choreography languages*.
- A detailed understanding of the BPMN 2.0 meta-model dedicated to choreographies. Complex modeling languages such as BPMN 2.0 should always be evaluated. In addition to evaluating the language, they also enable readers to *acquire a better understanding of the language*.

1.3 Structure of the paper

This paper is structured as follows. Firstly an overview of the choreography, subset of BPMN 2.0 is given in Sect. 2. Section 3 gives details of the research methodology followed to carry out this evaluation. This section also presents the language quality framework upon which our evaluation is based. This framework is very general and can be adapted/extended to fit the requirements of specific domains. In this respect, the requirements for choreography modeling are identified and analyzed, leading to an extended language quality framework tailored to the evaluation of choreography modeling languages. Sections 4, 5 and 6 explain the evaluation of the choreography subset of BPMN 2.0. The evaluation relies on the requirements that extend the quality framework according to three different axes. Finally, Sect. 7 puts the results into a broader perspective.

2 Choreographies in BPMN 2.0

A choreography in BPMN 2.0 formalizes the way business *participants* coordinate their *interactions*. In a choreography, the focus is not on the work performed internally by each participant, but rather on the *exchange of information* (messages) between participants. Another way to look at choreography in BPMN 2.0 is to view it as a type of *business contract* between two or more organizations.

In this section, the two choreography modeling styles are introduced: first of all (Sect. 2.1). Then, the three diagrams that can be used in BPMN 2.0 to capture different views in a choreography are briefly described. The conversation diagram is described in Sect. 2.2, the choreography diagram in Sect. 2.3 and the collaboration diagram in Sect. 2.4.

The BPMN 2.0 standard provides a formal meta-model shown through UML class diagrams. Not all of the concepts of the meta-model are represented graphically in the diagrams. But the XML Schemas [8] fully implement the meta-model. An example of XML Schema and part of the meta-model in BPMN 2.0 are presented in Sect. 2.5.

2.1 The two choreography modeling styles

Currently, there are two choreography modeling styles: *interaction models* and *interconnection models* [19]. In the interaction models, the interactions (message exchanges between two participants) are the basic building blocks. In the case of interconnection models, messaging activities of different processes are interconnected (e.g., a "send activity" and the corresponding "receive activity").

In previous versions of BPMN, the only way to represent choreographies was with the interconnection point of view given by the *collaboration diagrams* (see Fig. 1c). Capturing

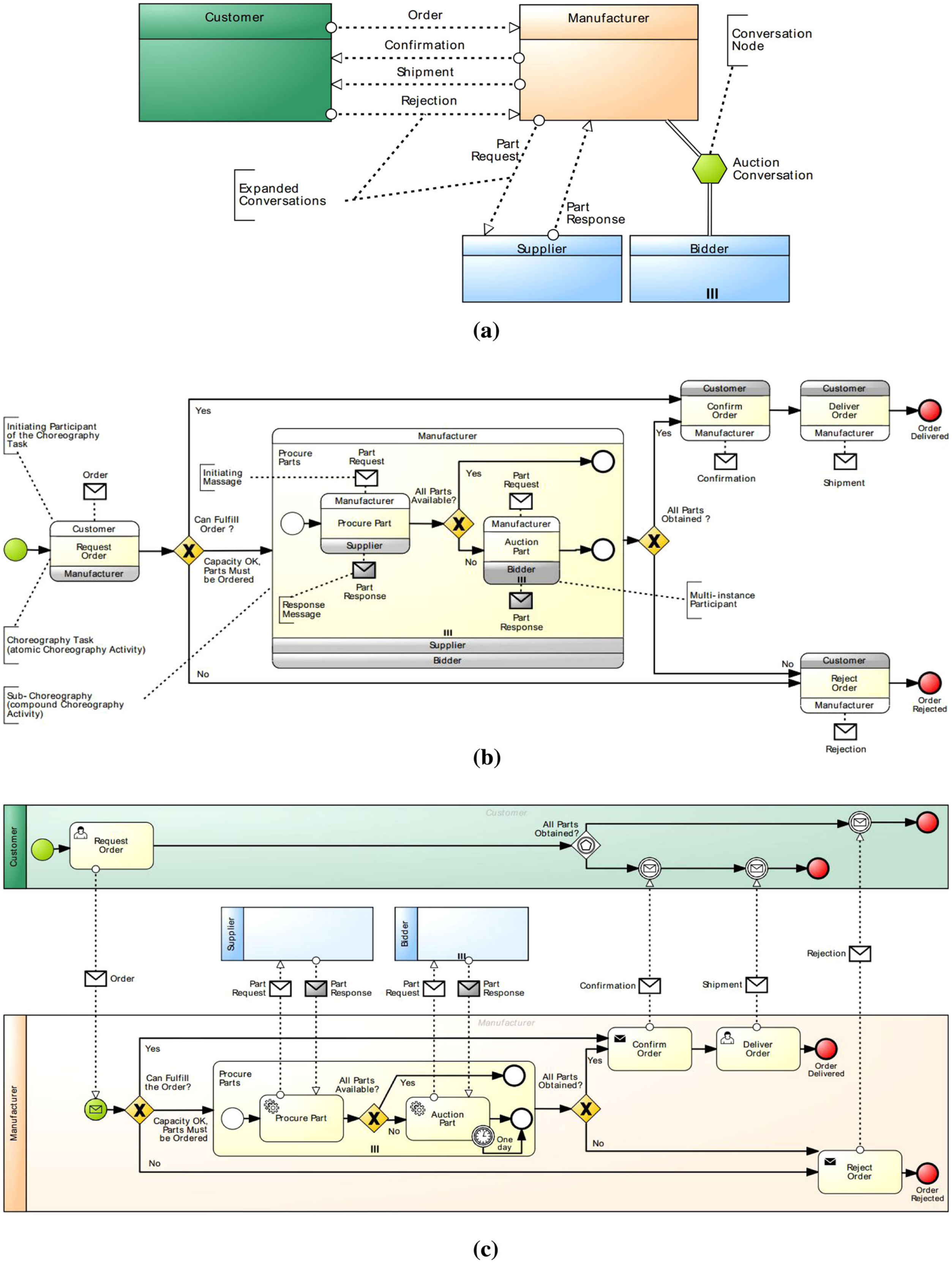

Fig. 1 Choreographies in BPMN 2.0. **a** Conversation diagram with two expanded nodes. **b** Choreography diagram (interaction model). **c** Collaboration diagram (interconnection model)

choreographies just by using collaboration diagrams could lead to ambiguities and deadlocks especially when the complexity increases [19]. In the new version of BPMN 2.0, modelers can describe choreographies using the *choreography diagram* corresponding to the interaction model. As shown in Fig. 1b, the overall view of the sequence of interactions in a choreography is represented in addition to the participants' view given by collaborations. These two diagrams can be represented together or individually to capture the choreography in BPMN 2.0. The latter diagrams are complemented with the new *conversation diagram* that gives a logical relation of the participants' communication (see Fig. 1a). In practice, this logical relation often concerns one or several business objects of interest, e.g., "Order," "Shipment and Delivery," or "Invoice."

Figure 1 illustrates the different views of the choreography given by the three diagrams in BPMN 2.0. This figure, inspired by one of the examples given in the BPMN 2.0 standard, captures a manufacturing process. A brief description of the three diagrams is given below.

2.2 The conversation diagram in BPMN 2.0

The conversation diagram in Fig. 1a gives a structural view of the participants' topology in the choreography. The *Conversation Nodes* indicate in an abstract manner how participants relate. For instance, the *Auction Conversation* indicates that several message exchanges between the *Manufacturer* and the *Bidder* for an auction are produced. Conversation nodes can be expanded into message exchanges. For instance, message exchanges between the *Customer* and the *Manufacturer* could be merged into an "Order" conversation node. Note that conversation diagrams are not necessary for tool developers to achieve "Choreography Conformance" [45]. However, as discussed in Sect. 4, conversations might be an important view for modeling choreographies in BPMN 2.0 because they are a good illustration of the structural view of the choreography.

2.3 The choreography diagram in BPMN 2.0

In the choreography diagram (Fig. 1b), interactions between participants are explicitly represented by means of *choreography tasks* (e.g., *Request Order*). The fact of having explicit interaction sequences avoids possible misunderstandings and deadlock errors. For each interaction, two *participants* are involved. One of them (the white band) is the *initiating participant* of the interaction (the one who sends the *initiating message*). The shaded one is the receiver. In Fig. 1b, the interaction *Procure Part* (*Choreography Task*) is initiated by the *Manufacturer* with the initiating message *Part Request*. The *Supplier*, who is the receiver, responds with a *Part Response* (the shaded message). The *control flow arrows* and *gateways* determine the sequence of the choreography elements. In the example shown, if the *Manufacturer* can fulfill the order, he acknowledges with a confirmation message. The *Customer* waits for the delivery of the order when the *Manufacturer* makes the shipment. If the *Manufacturer* cannot fulfill the order, then he sends an order rejection to the *Customer*. If the order cannot be fulfilled immediately but the *Manufacturer* has enough capacity, he contacts a *Supplier* to procure the missing parts.

2.4 The collaboration diagram in BPMN 2.0

In the collaboration diagram in Fig. 1c, the model simply captures the so-called *public process*. "A public process represents the interactions between a private business process and another process or participant" [45]. Only activities that are used to interact with other *Participant*(*s*), as well as the sequence of these activities, are included in this public process. It thus defines the interface of a participant's process that is accessible to other participants. In the example, only the public process of the *Manufacturer* and a part of the *Customer's* process are shown. The other participants are represented as black boxes to simplify the diagram. It is assumed that the achievement the role of these black boxes corresponds to some sort of service or is automatized. Therefore, *Procure Part* and *Auction Part* are represented as *Service Tasks* (gear icon). On the other hand, *Request Order* and *Deliver Order* are "human activities" represented with the so-called *User Tasks* (user icon). Finally, *Confirm Order* and *Reject Order* are represented as *Send Tasks* (black envelope icon).

In the evaluation, the collaboration diagram, the choreography diagram and the conversation diagram are taken into account. As already seen, the three diagrams form different but complementary views of the same choreography notion. However, note that the graphical notation is only a partial representation of the meta-model. The standard also provides a textual representation in XML that implements the complete meta-model. This textual notation is explained below.

2.5 The XML Schemas in BPMN 2.0

The XML Schemas in BPMN 2.0 implement all the constructs defined in the meta-model. All the conceptual information in the diagrams is formalized in the XML Schemas. Therefore, certain concepts and properties of the meta-model are not represented graphically but only implemented textually in the XML Schemas.

Listing 1 shows an example of the choreography task schema given in the standard. Note that the properties *minOccurs* and *maxOccurs* that indicate the minimum and maximum number of occurrences of the task, respectively, are captured. However, these two properties are not represented explicitly in the choreography diagram.

```
<xsd:element name="choreographyTask"
    type="tChoreographyTask"
    substitutionGroup="flowElement"/>
<xsd:complexType name="tChoreographyTask">
    <xsd:complexContent>
        <xsd:extension base="tChoreographyActivity">
            <xsd:sequence>
                <xsd:element name="messageFlowRef"
                    type="xsd:QName"
                    minOccurs ="1"
                    maxOccurs ="2"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

Listing 1 ChoreographyTask XML Schema

Figure 2 shows part of the choreography meta-model. The brightness of the classes differentiates between constructs that are graphically captured or not. The white classes such as the *Participant* or the *Collaboration* are constructs that are explicitly represented in the graphical notation. The clear gray classes are those that are represented, but there is no specific graphical construct to represent them in an unambiguous way. For example, the standard says: “Potentially, both the *PartnerEntity* name and *PartnerRole* name can be displayed for the *Participant*.” As a result, when a participant is represented graphically, it is not possible to determine whether the name refers to a role (e.g., *Manufacturer*) or a concrete entity (e.g., IBM) without the domain knowledge. Finally, the dark gray classes are only implemented in the XML Schema. This means that no graphical construct is defined for these classes. The complete description of the constructs and the complete meta-model are found in the standard.

The difference between what is represented graphically in the diagrams and what is actually implemented in the XML Schema motivates the evaluation of BPMN 2.0 in both levels separately. This separation gives an overview of the support for the requirements in both the graphical representation and in the overall concepts presented in the meta-model. The analysis of the different supports leads to interesting conclusions.

3 Research methodology

This section explains the different phases were performed to carry out the evaluation. First, some related work aimed at evaluating BPMN (Sect. 3.1) is introduced. Section 3.2 gives an overview of the evaluation approach. Section 3.3 explains and briefly describes the motivation to choose the language quality framework. Section 3.4 details the extension of the framework. Section 3.5 briefly describes the empirical studies that were performed. Finally, Sect. 3.6 summarizes the research methodology.

3.1 Looking at previous evaluations of BPMN

There have been other previous evaluations of BPMN. Wohed et al. [60] evaluated BPMN (v.1.1) in terms of the workflow patterns [55]. Recker et al. [47] evaluated BPMN (also v.1.1) in terms of the Bunge-Wand-Weber (BWW) ontology. The latter analysis highlighted a certain degree of construct redundancy and overload. Meanwhile, Wahl & Sindre [57] and Nysetvold et al. [42] evaluated BPMN using a Semiotic Quality Framework [33]—the same framework that has been used (with a number of extensions) as the basis of our evaluation. Like Recker et al., Wahl and Sindre’s conclusions highlight a certain degree of construct overload in BPMN that could potentially have a negative effect on comprehensibility. These previous evaluations focused on earlier versions of BPMN, which did not include the choreography subset. Hence, BPMN 2.0 evaluations such as ours complement the others.

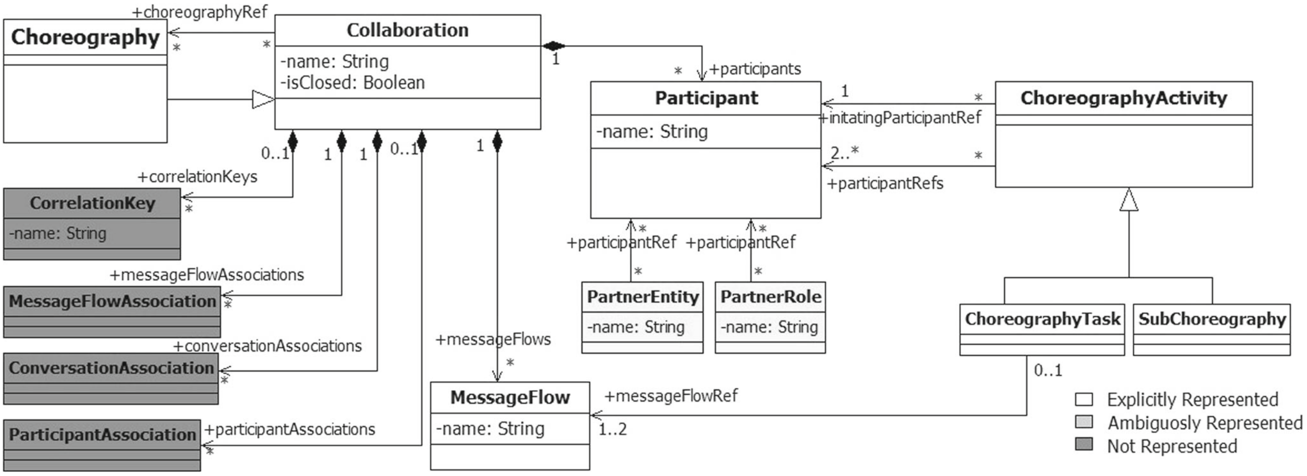

Fig. 2 Part of the Choreography Meta-model in BPMN 2.0

More recently, Oliver Kopp et al. [32] used the requirements identified by Decker et al. [17] to evaluate the choreography capabilities of BPMN 2.0. This is one of the most detailed prior evaluations of choreography definition languages, which is based on the service interaction patterns [5]. But these patterns only cover a perspective of the requirements for choreography definition languages (the domain perspective). Consequently, this pattern-based evaluation framework was complemented with other perspectives, namely the comprehensibility and technical perspectives.

Genon et al. [27] analyzed in detail the cognitive effectiveness of the graphical notation in BPMN 2.0, relying on the nine *Principles of Graphical Notation* proposed by Moody [40]. However, they did not consider the choreography subset. Their work is mainly focused on graphical notation and does not address the meta-model. This paper considers graphical notation as an important evaluation axis for comprehensibility, but it is completed by the meta-model evaluation as they are closely related.

In a previous article, choreographies in BPMN 2.0 were evaluated by extending a quality framework [13]. Several major issues in the latter evaluation are discussed in [12]. This paper reviews and extends the previous evaluation giving more precision to the evaluation in terms of requirements and analysis. Further details regarding this extension of our previous publication are given later in Sect. 3.6.

3.2 Overview of the evaluation approach

We adopted the quality framework approach, which has been used in a number of recent evaluations of BPMN [42,57]. This approach was embraced because it allows us to assess a broad and relevant spectrum of characteristics of the language. The semiotic quality framework [35] was chosen as it is more precise and addresses both the human and technical viewpoints of a language. The high level of generality of the framework is overcome by refining the axes with specific choreography requirements. The evaluation of each axis is based on requirements given by the literature. It distinguishes between two levels of abstractions corresponding to the graphical notation used for analysis and the textual notation captured in XML Schemas for design. The first axis evaluates how BPMN 2.0 supports the choreography key domain concepts (Sect. 4); the second is aimed at analyzing the comprehensibility of the language (Sect. 5); the third evaluates how well the language can be supported by tools (Sect. 6).

Several tables provide the name and description of each choreography requirement. The tables also provide the evaluation rating based on the analysis performed and on discussions with other experts in the field.

At a given level (Graphical or XML Schema), in BPMN 2.0, a requirement may be rated as:

- A **0** if it is *Not supported*. There is no construct in the language for capturing the requirement.
- A **1** if it is *Partly supported*. The requirement could be captured to a certain degree, but not in an overall and explicit way, so different interpretations might be induced.
- A **2** if it is *Fully supported*. There is an explicit construct in the language for capturing the requirement without ambiguity.

Although our evaluation is primarily *expert centered*, we nevertheless performed *experimental studies* for the comprehensibility and domain areas to help evaluate certain requirements. These experimental evaluations are described in greater detail in Sect. 3.5.

When a flaw in BPMN 2.0 is identified, an attempt is made systematically to give some leads to extend the standard in a way that will fix it. In the evaluation tables, a reference is added that relates the requirement and the text describing the proposed suggestion, and tables are included to summarize suggestions.

At the end of the evaluation of each axis, the strong points and weak points of the language, together with the corresponding suggestions, are summarized.

3.3 Choosing the evaluation approach: the language quality framework

As can be noted from related work, two evaluation approaches are generally used: *qualitative* and *quantitative* [9].

On the one hand, in a **quantitative approach**, specific metrics are needed to convert into numbers the evaluations made. But looking at certain aspects of the language, such as graphical notation or meta-model quality, measurable metrics are difficult to identify [36]. Rossi and Brinkkemper [49] proposed a method for calculating the conceptual complexity. In this case, the assumption (not proven) is that there is an intrinsic dependence between the meta-models and the ease of learning the language: "the more complex a meta-model, the harder the method is to learn." Siau and Cao [53] arrived to similar results. However, as the BPMN 2.0 meta-model is UML-based, research in the domain of software quality where the UML class diagram quality is surveyed can be applied [26].

On the other hand, a common **qualitative approach** for evaluating languages is by means of empirical studies (e.g., controlled experiments or surveys) [40,47]. Although such evaluations provide valuable insights, they are time-consuming and only allow one or two specific aspects of a language to be evaluated (e.g., understandability or readability). Other qualitative application is to study the language meta-model by experts in the domain. For example, it has been also shown by this approach that an intrinsic dependency exist between the meta-model and the learnability of a language [28]. BWW ontology-based evaluations are also focused on the conceptual basis of modeling languages. But these evaluations are limited as a language cannot be reduced to its meta-model. For example, when it comes to understandability, concrete syntax plays a significant role [40]. Therefore, a more general approach is needed that can be used to evaluate an entire language without requiring the systematic use of metrics.

A complementary approach is provided by **quality evaluation frameworks** such as the language quality framework presented by Krogstie [35] which extends the framework of Lindland, Sindre, and Sølvberg (LSS) [37]. This framework assesses a broad spectrum of characteristics of a language so it can be seen as a "guide" to evaluate the quality of languages. The Language Quality Framework has already been used in previous language evaluations. As previously mentioned, Wahl and Sindre's [57] and Nysetvold et al. [42] have evaluated BPMN this framework. Krosgtie evaluated the Unified Modeling Language (UML) in [33] as well as enterprise modeling languages [34].It was also used in our previous evaluation of BPMN 2.0 [13]. More recently, Nelson et al. [41] combined the aforementioned LSS and BWW frameworks to propose a conceptual modeling quality framework (CMQF).

Another well-known quality framework is the ISO/ IEC-9126 [30]. Mancioppi et al. [38] relied on the latter to propose a model quality framework for choreographies. They focus on models, while the accent is placed on language's characteristics.

Among the quality frameworks [24], Krogstie's framework was chosen because it is widely used, it is specific for modeling languages and it is sufficiently generic to be easily extended. In addition, it accommodates both qualitative and quantitative methods. The framework identifies five characteristics of modeling languages:

Domain Appropriateness (D). This relates to the quality of language semantics. The key underpinning idea is that the conceptual basis of the language must be powerful enough to express anything in the domain but no more.

Participant Language Knowledge Appropriateness (P). The language must be appropriate for the participants (modelers). It is best to base a language on experience with previously used languages.

Knowledge Externalizability Appropriateness (K). There should be no statement on the explicit knowledge of the participants that cannot be expressed in the language. This is also related to the quality of the language semantics.

Comprehensibility Appropriateness (C). This relates the language to the social actor's interpretation. It considers the concepts of the language and its notation. The language should be cognitively manageable (not too many concepts) and the concepts must be comprehensible and distinguishable from each other, etc.

Technical Actor Interpretation Appropriateness (T). This relates the language to the tools. The syntax and the semantics of the language must be sufficiently formal so that the tools can automate certain treatment processes on models.

In our evaluation, consideration was given to characteristics that relate the language to the *modeler's explicit knowledge* and to *social actor* in order to form a common "human" characterization (see Fig. 3). The focus was placed on *Comprehensibility Appropriateness* to include the requirements that relate the language and the "human." *Participant Language Knowledge Appropriateness* and *Knowledge Externalizability Appropriateness* rely on the empirical knowledge of modelers, which is difficult to categorize. Modelers' knowledge may vary over time and may differ depending on the level of expertise. However, looking at *Participant Language Knowledge Appropriateness*, it can be argued that for those who are familiar with BPMN's process diagrams it should not represent any difficulty to learn this new representation. The knowledge and experience acquired in using one diagramming technique can be transferred to another diagramming technique in BPMN 2.0.

The evaluation focuses on axes that are as objective, as possible namely *Domain Appropriateness*, *Comprehensibility Appropriateness* and *Technical Actor Interpretation appropriateness*. Krogstie has already evaluated UML by focusing primarily on these three axes of the framework [33]. They are highlighted in Fig. 3.

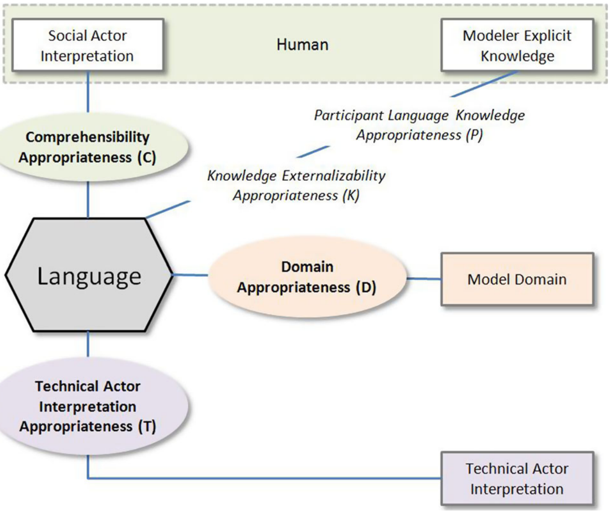
Social Actor Interpretation
Human
Modeler Explicit Knowledge
Participant Language Knowledge Appropriateness (P)
Comprehensibility Appropriateness (C)
Knowledge Externalizability Appropriateness (K)
Language
Domain Appropriateness (D)
Model Domain
Technical Actor Interpretation Appropriateness (T)
Technical Actor Interpretation

Fig. 3 Krogstie's language quality framework [33] highlighting the three main axes of the evaluation: comprehensibility, domain and technical appropriateness

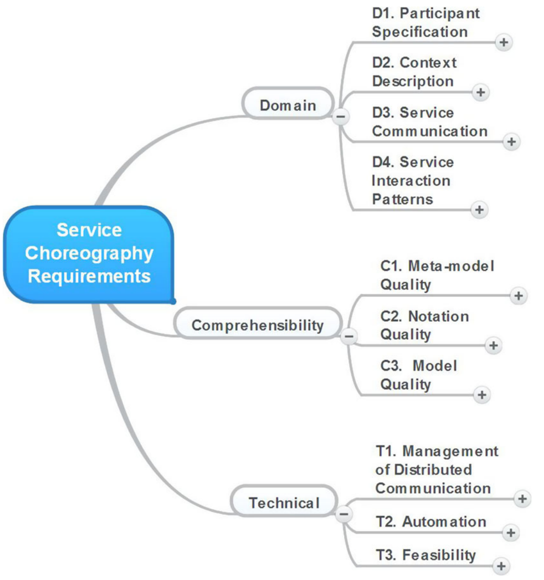
Service Choreography Requirements
Domain
D1. Participant Specification
D2. Context Description
D3. Service Communication
D4. Service Interaction Patterns
Comprehensibility
C1. Meta-model Quality
C2. Notation Quality
C3. Model Quality
Technical
T1. Management of Distributed Communication
T2. Automation
T3. Feasibility

Fig. 4 The requirements axes extending the language quality framework

3.4 Extending the quality framework with the choreography requirements

This Section describes how to extend the three axes of the quality framework, namely *Domain*, *Comprehensibility* and *Technical* appropriateness. Figure 4 illustrates an overall view of the extended quality framework for choreographies. The requirements that are identified in the domain, comprehensibility and technical areas are broken down even further to achieve a finer analysis.

3.4.1 Domain appropriateness requirements

Domain requirements rely on the service choreography requirements identified by Decker et al. [17]. These requirements had already been used to evaluate choreography languages such as BPMN, BPEL4Chor or WS-CDL in different studies, such as [18,32].

This axis also includes the requirements' framework proposed by Schönberger et al. [52]. In this technical report, the authors performed a very detailed analysis of the Business-to-Business integration (B2Bi) requirements relying on different abstraction levels. In our evaluation, the requirements identified for the following B2Bi levels were included: (1) the *business process model* that "specifies the flow of information among the partners in an abstract way," (2) *choreography*, which "captures the overall message exchange between partners in a more technical way to be processed by machines" and (3) *public orchestrations* that "specify interactions from the point of view of individual partners." Note that the authors separate *choreography* requirements from *public orchestration* requirements. Nevertheless, as explained in Sect. 2.1 above, *choreography* and *public orchestrations* may be considered together as two different views of the same collaborative process. Therefore, all the requirements that are identified in these three levels are considered in our evaluation.

B2Bi requirements are also filtered thanks to the consideration ratings given by the authors in the paper. The scale [%,−,0,1,2] to represent the degree of consideration of B2Bi requirements is proposed, varying from *not applicable (%)*, *should not be considered (–)*, *could be considered (0)* and *should be considered (1)* to *is strongly recommended to be considered (2)*. The requirements are filtered following these simple rules:

1. If a requirement is evaluated as *not applicable (%)* or *should not be considered (–)* or *could be considered (0)*, then it is **not considered** in our evaluation.
2. If a requirement is evaluated as *should be considered (1)* or *strongly recommended (2)*, then it is **considered** in our evaluation.

Consider now the list of requirements for *Domain* appropriateness. Figure 4 shows the analysis of how the participants are specified in choreographies (D1). For instance, the capacity of the language to support multi-party collaborations or role modeling is evaluated. The manner in which the context is captured (D2) is also evaluated taking into account, for example, time constraints or exceptions. The manner in which partners communicate is performed is evaluated by paying special attention to message exchanges (D3). The inclusion of the service interaction patterns [5] as an explicit point of evaluation is also an important point (D4). Prior evaluations of choreography languages such as WS-CDL [6], Let's Dance [61], BPEL4Chor [17], MAP [2] or even BPMN extensions to support choreographies [18] are mainly based on these patterns. But service interaction patterns are considered to cover only part of the domain's perspective for choreography definition languages. Accordingly, this pattern-based evaluation is complemented by other key domain requirements that influence their support.

Domain requirements are described and evaluated in Sect. 4.

3.4.2 Comprehensibility appropriateness requirements

Three main points are considered when dealing with comprehensibility. The first establishes that the meta-model of a choreography modeling language must allow stakeholders to gradually understand and design choreographies (C1). For instance, different views and levels of abstraction may improve the meta-model and consequently the language [11]. The second point is focused on the notation quality (C2), which establishes that the concrete syntax of a language should be cognitively effective (optimized for human communication and problem solving). This can be achieved by supporting the *Principles of Graphical Notations* [40]. These principles give theoretical and empirical foundations related to visual aspects of notations in our evaluation. Finally, the evaluation deals with points such as the model's ease of explanation [1]: It focuses on the necessity to produce effective models (C3).

Comprehensibility requirements are described and evaluated in Sect. 5.

3.4.3 Technical appropriateness requirements

To elaborate technical requirements, consideration was given to industry initiatives and previous choreography languages. In particular, B2Bi requirements [52], which are more focused on the industry initiative RosettaNet and the ebXML language [23], were analyzed. Other choreography technical requirements to complete this axis are identified by Barros et al. [6] evaluating WS-CDL, Decker et al. [17] analyzing BPEL4Chor and Barker et al. [2] looking at MAP.

This study is focused on how the choreography subset of BPMN 2.0 is supported by tools. the first step is to look at how the distributed communication between partners may be managed (T1). For instance, the use of standards, support for message formats or quality of service are evaluated in this axis. Several criteria that are necessary for a given language to achieve automation (T2) are also evaluated, such as formalism or traceability between different models. Feasibility (T3) corresponds to the agreement needs that may be established to support interoperability between different tools and technologies. For instance, the portability or the language flexibility in this area was analyzed.

Technical requirements are described and evaluated in Sect. 6.

3.5 Performing empirical studies

Two empirical studies were conducted and are included in the user-centered analysis methodology illustrated in Fig. 5. The user-centered design method is described in ISO standard 9241 [21]. It gives the requirements and the recommendations that must be followed by computer science projects in order to obtain a ”human-centered” characterization.

With a view to making this method more pragmatic and easier to deploy, Mandran et al. [39] proposed to use a different terminology from the ISO standard. This terminology, used in Fig. 5, refers to the goals of each step:

- **Exploration**, instead of analysis, to form an in-depth “views of users” expectations and practices. Our expert-centered analysis, which is the main contribution of this paper, is placed in this step.

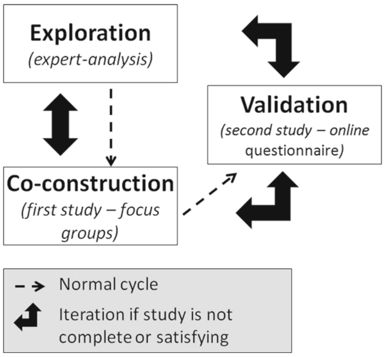
Exploration
(expert-analysis)
Validation
(second study – online questionnaire)
Co-construction
(first study – focus groups)
Normal cycle
Iteration if study is not complete or satisfying

Fig. 5 User-centered methodology including the empirical studies

- **Co-construction**, instead of conception, where the work is conducted in order to involve all the different users concerned by the product. An initial study was carried out in this step, relying on *focus groups* [50]. The aim of this study was to explore new ideas concerning the abstract syntax of the language (choreography meta-model) and the concrete syntax (graphical notation).
- **Validation**, instead of evaluation, because the evaluation can be conducted during each step, while the validation takes place at the end of the processes. In fact, the evaluation can be conducted on the exploration results or at the end of the co-conception. In this case, the evaluation enables the strengths and weaknesses of the results to be identified at each step. The validation indicates the last step to test the product with the user. Obviously, an evaluation can also be conducted in this step implying potential iterations. The second study performed, using an online questionnaire, corresponds to this step. It then seemed advantageous to find a more quantitative way of validating several points that emerged in the previous study.

The *first study* has already been presented in a previous article [10] which relies on the scenario illustrated in Fig. 1b. Eight service and/or business process specialists participated in the study. A facilitation guide was used to schedule the evaluation, with several questionnaires, and an evaluation protocol was followed.

The study was based on two hypotheses. First, it was necessary to check whether *“our own choreography meta-model correctly represents the semantics for choreographies.”* The evaluation of this hypothesis was based on identifying the meta-model concepts within a textual scenario (where a choreography could be applied). The subjects identified concepts in the meta-model in the proposed scenario and looked for missing or superfluous concepts. This first part helped refine our choreography meta-model and therefore compare it with the one proposed by BPMN 2.0. The second hypothesis, namely *“our graphical notation (based on UML sequence diagrams) is appropriate for describing* *choreographies,"* was then checked. The evaluation of the second hypothesis involved comparing our notation with the choreography notation proposed by BPMN 2.0. This helped to evaluate the notation quality and some points of the model quality.

Therefore, this first study helped identify several improvement points in BPMN 2.0 which are discussed in Sect. 5. The group discussions allowed new ideas to emerge.

The *second study* was focused on several points that emerged in the previous study such as the use of a multilayered and multi-view meta-model and some concrete graphical notation difficulties in BPMN 2.0 compared to UML. This stage involved a greater number of subjects in order to obtain a quantitative evaluation of the proposals. The evaluation method chosen is an online questionnaire.[2] This questionnaire looked for points that did not need an in-deep knowledge of the choreography modeling domain. Every effort was made to minimize the time needed to answer the questionnaire (about 15 min) in order to motivate respondents.

The questionnaire was answered by 40 people. The targeted subjects were computer scientists with different skills. Their modeling habits and their knowledge of business processes (e.g., Business Process Management, BPMN, etc.) were assessed. Three different profiles were identified: (1) the minority (5 out of 40) were familiar neither with modeling languages such as UML nor business processes; (2) around a half of the subjects (19 out of 20) were familiar with modeling languages but not with business processes; and (3) finally, 15 out 40 declared to be familiar with both. Therefore, it was an heterogeneous group of subjects.

The questionnaire comprises four main stages:

1. In the first step, subjects are asked to pick the best representation of a simple scenario: "A Participant A sends multiple messages to a Participant B." Subjects may choose between the BPMN 2.0 option and other similar representations that were proposed.
2. The second question is similar to the first. The subjects have to choose the best representation for a scenario where an interaction involves several Participants of type B.
3. The third step deals with the BPMN 2.0 meta-model presentation, focusing on the choreography and the service part (class diagrams taken from the standard [45]). The subjects were asked to describe the strong and weak points of its presentation though open-ended questions. They were also asked to identify the points that make it difficult to understand. Afterward, the contribution of applying abstraction levels, view separation and introducing several visual variables [40] to the same meta-model was analyzed.
4. Finally, subjects were asked about their opinion on naming conventions.

This second study helped validate important ideas that emerged from the first study concerning both the meta-model and how certain concepts are captured in the graphical notation. The findings of this questionnaire are discussed later in Sects. 4 and 5.

Thanks to these two experimental studies and the preceding expert analysis, a complete cycle of the user-centered methodology was performed. However, for the time being, the methodology has yet to be applied to every point of this BPMN 2.0 evaluation. This will imply several iterations in the cycle. At the end of the paper (Sect. 7.1) the limits of the evaluation are discussed.

3.6 Summary of the research methodology

This paper evaluates the subset of choreography constructs of BPMN 2.0. The general quality framework has been extended to refine the different axes with the choreography requirements. These requirements arise from other already well known and accepted requirements, the analysis of previous choreography language proposals, bibliographical sources as well as practical studies.

Major changes were introduced with respect to the previous evaluation performed [13] to extend our work:

- In this paper, the three diagrams for capturing choreographies as they are different views of the same model are considered. In the previous evaluation, only the choreography diagram was considered.
- *Service interaction patterns* were not included in the previous evaluation framework. Here, they are considered explicitly in the domain axis.
- This extension provides greater precision especially concerning the *Technical Appropriateness* and the *Domain Appropriateness* requirements. Previously, industrial proposals such as RosettaNet that provide valuable information on industry requirements in the choreography field were not considered.
- The framework has been enriched by including the work performed by Schönberger et al. [52] in the B2Bi requirements.
- The graphical constructs and the XML Schemas must be evaluated separately. This separation gives a clear vision of the information under the graphical diagrams in BPMN 2.0. It can also show issues that may arise as result of the gap between both levels.
- Several findings from two empirical studies that we performed were included in order to validate various points in the *Domain* and the *Comprehensibility* axes.

[2] http://goo.gl/AsvO7.

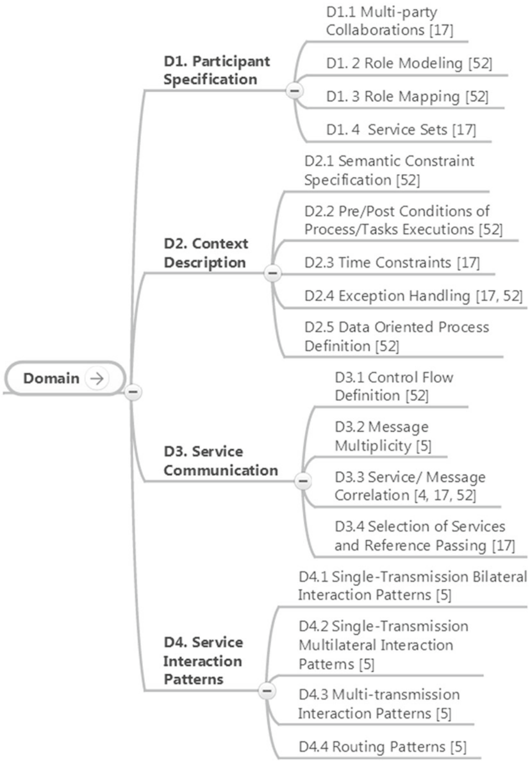
Domain
D1. Participant Specification
D1.1 Multi-party Collaborations [17]
D1. 2 Role Modeling [52]
D1. 3 Role Mapping [52]
D1. 4 Service Sets [17]
D2. Context Description
D2.1 Semantic Constraint Specification [52]
D2.2 Pre/Post Conditions of Process/Tasks Executions [52]
D2.3 Time Constraints [17]
D2.4 Exception Handling [17, 52]
D2.5 Data Oriented Process Definition [52]
D3. Service Communication
D3.1 Control Flow Definition [52]
D3.2 Message Multiplicity [5]
D3.3 Service/ Message Correlation [4, 17, 52]
D3.4 Selection of Services and Reference Passing [17]
D4. Service Interaction Patterns
D4.1 Single-Transmission Bilateral Interaction Patterns [5]
D4.2 Single-Transmission Multilateral Interaction Patterns [5]
D4.3 Multi-transmission Interaction Patterns [5]
D4.4 Routing Patterns [5]

Fig. 6 Overview of domain requirements

In conclusion, the catalog of identified requirements needed to perform the evaluation represents a clear overview of possible criteria for evaluating the quality of different choreography languages. It may be a starting point for evaluating other choreography languages or future versions of BPMN. The evaluation also enables the reader to acquire a better understanding of the language.

Valuable ideas are also given on the empirical studies that can be performed in order to implement a user-centered methodology for this kind of evaluation.

The next section provides an analysis of BPMN 2.0 for choreographies. Section 4 evaluates *Domain* requirements. *Comprehensibility* requirements are assessed in Sect. 5. Finally, Sect. 6 looks at *Technical* requirements.

4 Domain requirements evaluation

In this section, the *Domain Requirements* previously presented in Sect. 3.4 are described and evaluated in detail for BPMN 2.0. Figure 6 gives an overview of the evaluated requirements. The four general requirements, namely *D1. Participant Specification*, *D2. Context Description*, *D3. Service Communication* and *D4. Service Interaction Patterns*, are further divided into sub-requirements. The description and the evaluation of each sub-requirement can be found in Tables 1, 2, 3 and 4, respectively. A discussion on the evaluations for each sub-requirement is presented below:

4.1 D1. Participant Specification

Figure 7b shows the part of BPMN 2.0 meta-model related to participants. The different classes in the meta-model are explained with regard to the different requirements. Table 1 describes the requirements and gives the rating values.

D1.1 Multi-party collaborations. The example on Fig. 1 shows that more than two participants can be involved in the choreography model. In the example, the participants are the *Customer*, the *Manufacturer*, the *Supplier* and the *Bidder*. This requirement is critical because this point is part of the definition of the choreography. There is full support in both the graphical and the XML Schema layers as the participant's representation is a core concept in BPMN 2.0 meta-model and participants are graphically captured. In a collaboration and therefore also in the choreography (inherited from the collaboration) one or many participants can be defined (see

Table 1 Participant specification requirements

Refinement	Description	Graphical	XML Schema	Suggestion
D1.1 Multi-party Collaborations	More than two participants (services) might be involved in a choreography	2	2	
D1.2 Role Modeling	A functionality might be specified in an abstract manner that then can be bound to concrete integration partners	1	2	S-D1.2
D1.3 Role Mapping	Denotes the mapping of an abstract functionality to concrete implementations/instances as well as the mapping of roles to other roles	1	2	S-D1.3
D1.4 Service Sets	Several participants (services) of the same type might be involved in a choreography	1	2	S-D1.4
		5/8 (62.5 %)	8/8 (100 %)	

D1. Participant Specification: A choreography language should support the representation of more than one participant (service) and more than one instance of the same participant. It should also give the possibility of defining "abstract participants" as roles and map them to concrete instances or implementations

Table 2 Context description requirements

Refinement	Description	Graphical	XML Schema	Suggestion
D2.1 Semantic Constraint Specification	For validating, enacting and managing constraints. Constraints can be used during the initial definition of a choreography, but also when choreographies have to be modified during the design stage or even during run time	1	1	S-D2.1
D2.2 Pre-/Post-Conditions of Process/Task Executions	The definition of pre-/post-conditions of process/task execution targets during automatic evaluation of constraints particularly regarding task requirements	0	0	S-D2.2,S-D2.2′
D2.3 Time Constraints	A choreography language should allow participants to agree upon the maximum allocated time for receipt of a given expected message and the time allocated to a participant to transmit a message	2	2	
D2.4 Exception Handling	Messages may not be sent within the allocated time or may contain faults. Modelers should be able to define paths in a choreography that handle such negative scenarios	2	2	S-D2.4
D2.5 Data-Oriented Process Definition	It should be possible to define routing rules based on the data being exchanged and to define how data are transferred or where they are visible.	2	2	
		7/10 (70 %)	7/10 (70 %)	

D2. Context Description: A choreography language should support an accurate definition of the context also taking into account exceptions and time constraints

Table 3 Service communication requirements

Refinement	Description	Graphical	XML Schema	Suggestion
D3.1 Control Flow Definition	Control flow definition using typical control flow constructs like AND/OR-Fork and AND/OR-join is inevitable and has been extensively investigated by the scientific community	2	2	
D3.2 Message Multiplicity	An undefined number of messages might be sent or received by the participants. The language should support the dispatch and the receipt of several messages by a participant	1	1	S-D3.2
D3.3 Service/ Message Correlation	Possibility of distinguishing different conversations as well as relating an incoming message to the previously sent or received message must be supported	0	2	S-D3.3
D3.4 Selection of Services and Reference Passing	Selection (of a service) can be made in design, deployment or execution time. A reference of a selected participant (service) may be passed to another participant as there is no central coordinator in choreography	0	1	S-D3.4
		3/8 (37.5 %)	6/8 (75 %)	

D3. Service Communication: a choreography language should capture precisely the communication between participants considering that no central coordinator in choreographies might be defined

Fig. 7). *Sequence Flows* and *Choreography Activities* capture the interactions between participants. This requirement is therefore rated with full support in both levels.

D1.2 Role Modeling. In BPMN 2.0, a participant plays a role in a choreography. As Fig 7b shows, BPMN 2.0 uses the term *PartnerRole* linked to the construct *Participant* to support this. However, there is no strict rule about how the role definition should be represented graphically. The meta-model part defining the *Participant* leads to ambiguity. Firstly, because it is not mandatory to link a *PartnerRole* to a *Participant* in a choreography; and secondly, because there is no need to assign a name to the participant to achieve a valid model and finally, because the *PartnerEntity* or the *PartnerRole* name can be displayed for the *Participant*. This could easily lead to confusion. Ideally, the label appearing in the participant's band should have only a single meaning. This

Table 4 Service interaction patterns requirements—**Evaluation**

Refinement	Description	Graphical	XML Schema	Suggestion
D4.1 Single-Transmission Bilateral Interaction Patterns	Elementary interactions where a party sends or receives a message and as a result expects a reply (sends a reply)	2	2	
D4.2 Single-Transmission Multilateral Interaction Patterns	A party may send or receive multiple messages but as part of different interaction threads dedicated to different parties	2	2	
D4.3 Multi-transmission Interaction Patterns	Multi-transmission (non-routed) interactions, where a party sends (receives) more than one message to (from) the same logical party	1	1	S-D4.3
D4.4 Routing Patterns	Dedicated to routed interactions where communication between two participants might be established through an intermediate participant	0	1	S-D4.4
		5/8 (62.5 %)	6/8 (75 %)	

D4. Service Interaction Patterns: A choreography language should support the 13 interaction patterns [5]

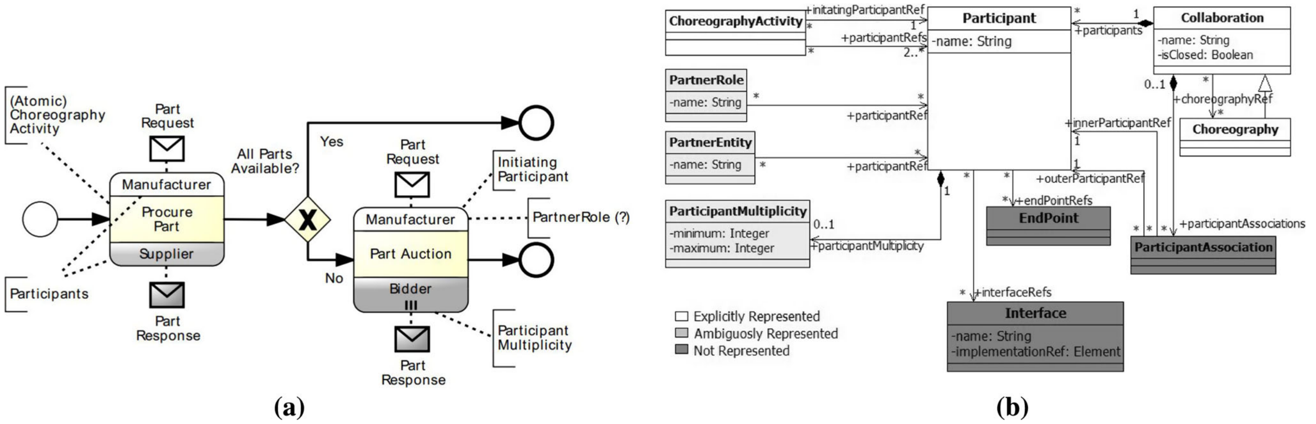

Fig. 7 Difference of Support between graphical and XML Schema Level in BPMN 2.0 for Participants. **a** Example of participant (graphical). **b** Participant meta-model (XML Schema level)

would follow the principle of semiotic clarity [40] that recommends a one-to-one relation between graphical elements and meta-model constructs. *PartnerRole*, *PartnerEntity* and *Participant* should be differentiated explicitly in a graphical way to avoid misunderstandings [S-D1.2]. This differentiation can be observed in Let's Dance [61] where the terms *Role* and *Actor Reference* are defined, or in BPEL4Chor [17] that defines the terms *participantType* and *participant*. As this requirement is explicitly captured in the meta-model but not graphically in BPMN 2.0, there is full support in the XML level, and only partial support in the graphical level.

D1.3 Role Mapping. When analyzing the capacity to map roles, a distinction is made between (1) the technical specification (XML), where mapping is focused on concrete implementations or instances. and (2) the graphical layer, where the possibility of mapping roles to other roles is assessed. In the first case, the BPMN 2.0 meta-model supports the assignment of one or more *EndPoints* to a *Participant* as can be seen in Fig. 7b. The *ParticipantAssociation* construct is used to associate different participants but who actually play the same role in a specific choreography. At the XML level, this requirement is fully supported. On the other hand, the latter construct is not captured graphically. A possible solution may be to label a participant with a textual annotation, a reference in the name's label or an icon to clarify the *ParticipantAssociation* concept (what is called in the standard the *inner* and the *outer* element) and therefore also support this requirement in the diagram [S-D1.3]. So role mapping is not fully supported graphically and fully supported in the XML Schema.

D1.4 Service Sets. BPMN 2.0 also supports the description of service sets. The example in Fig. 7 shows that several instances of the participant *Bidder* might be involved in the choreography. A multi-instance participant is indicated with three vertical lines. In the context of BPMN 2.0, it is possible to define how many participants of the same type are involved in a choreography using the *ParticipantMultiplicity* construct. However, this multiplicity is only supported in the

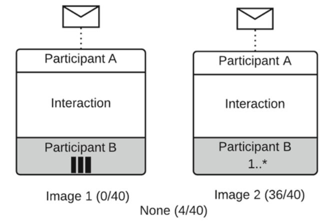

Fig. 8 EVAL. [Service Sets]—which graphical construct captures best the fact that an interaction may involve several participants of type B?

XML level and not represented graphically. This graphical failing could easily be clarified allowing modelers to explicit the number of participants taking part in the choreography [S-D1.4]. Therefore, the requirement is only partially supported graphically but fully supported in the XML Schema. In the online questionnaire, 36 out of 40 subjects pointed out that an UML-type notation (e.g., 1..*) where the number of participants is explicit, is more appropriate than the three vertical lines proposed by BPMN 2.0 (0 out of 40) to indicate participant multiplicity (see Fig. 8). Four out 40 subjects answered that neither of them where appropriate.

4.2 D2. Context Description

This section collects several requirements dedicated to defining the context in choreography. Table 2 gives the requirements' description and the corresponding rating values.

D2.1 Semantic Constraint Specification. There is no prescriptive way of defining semantic constraint specification in BPMN 2.0 for a choreography. It could be defined with *Textual Annotations*, but this implies that constraints cannot be automatically verified. However, these textual notations are helpful, especially when diagrams are used for communication between humans. This requirement is of much greater relevance at a graphical level. This construct does not appear explicitly in the BPMN 2.0 meta-model but could be partially supported by adapting the *Expression* or *Text Annotation* constructs. Therefore, this requirement is partly supported in both layers. If a *Text Annotation* was somehow related to the *Expression* construct or *Formal Expression* construct (inherits from *Expression*), a modeler could easily give formalism to semantic constraints and then fully support the requirement [S-D2.1].

D2.2 Pre-/Post-Conditions of Process/Task Executions. BPMN 2.0 does not directly support pre- and post-condition definitions. This requirement targets an automatic evaluation of constraints regarding task requirements. The meta-model does not define any construct referring to pre- and post-conditions. This requirement could be partially supported by *dataInput* and *dataOutput*, but these constructs are not considered for "choreography conformance" in the standard. Data input represents information needed to start an activity, and each data input can be defined as required or optional. Data output represents information that may be output from an activity. In BPMN 2.0, data inputs and outputs are usually omitted from the diagram and specified in the XML. The set of data inputs and outputs defines the *ioSpecification* for an activity or process. Choreography activities allow data inputs and outputs. Consequently, *dataObjet* constructs could potentially support the requirement, but they may have to be adapted for choreographies and not restricted to processes [S-D2.2]. Also, a link to goal-oriented languages such as KAOS [31], where formal verifications on pre- and post-conditions are made, could be a possible way of supporting this requirement [S-D2.2′]. Currently, this requirement is not supported in BPMN 2.0 for choreographies.

D2.3 Time Constraints. Time constraints can be defined in BPMN 2.0. Timers are represented in choreography diagrams graphically as well as in collaboration diagrams (inside a participant's pool). Note that time is not precise in choreography diagrams—there will be no exact synchronization between participants [45]. For absolute timers, for example, all participants have to know the time. In the example of Fig. 1, the timer is represented only in the collaboration diagram (Fig. 1c) and not in the choreography diagram (Fig. 1b) because this absolute timer does not necessarily have to be considered by all the participants such as the *Customer*. This requirement is therefore fully supported in both levels.

D2.4 Exception Handling (Fault Paths). Exceptions or fault paths are captured in collaboration diagrams by means of exception events which are not present in the choreography diagrams. Choreography exceptions might be transmitted as messages (no error message is defined). Exception constructs are present in the BPMN meta-model, and they are also graphically represented, so there is a full support at both levels. A possible improvement is to graphically differentiate the choreography messages caused by an exception from the rest of the messages [S-D2.4].

D2.5 Data-Oriented Process Definition. Data-oriented process definition is supported. BPMN 2.0 defines the *Data Objects* which are the primary constructs for modeling data within the process flow. In choreography, the focus is placed on the data that are exchanged between participants (i.e., *Messages*). These messages are explicitly represented in both collaboration and choreography diagrams. For example, in Fig. 1, the *order* represents a message sent by the customer to the manufacturer which initializes the manufacturer's process. The message represents a connection point between the choreography and the collaboration diagram. Data-based guards for XOR-Splits based on the exchanged messages could be defined. The standard defines the limits of the data used for gateway conditions. Data used in guard conditions must have been used in a message that was sent before reaching the gateway. This requirement is therefore rated as fully supported in both levels.

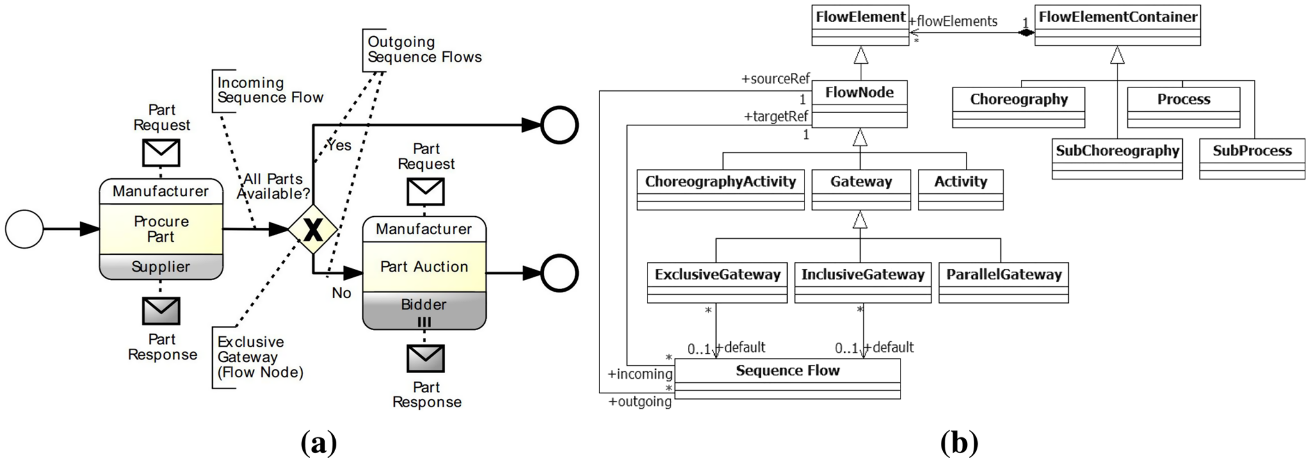

Fig. 9 Control flow definition through Gateways in BPMN 2.0. **a** Example of Gateway in BPMN 2.0. **b** Gateway Meta-model in BPMN 2.0

4.3 D3. Service Communication

This section concerns requirements that are needed for the proper definition of service communication in the choreography. Basic requirements, such as control flow definition and more technical requirements like reference passing or message correlation are evaluated. Table 3 gives the requirements' description and the rating values.

D3.1 Control Flow Definition. BPMN 2.0 supports control flow definition for choreographies. This requirement is represented by sequence flows and the basic types of gateways (e.g., the AND/OR-Fork and the AND/OR-Join) similar to process models. Figure 9 shows how these gateways are defined in the BPMN 2.0 meta-model and gives an example. These control flow constructs are considered to be fully supported as they are all captured in the BPMN 2.0 meta-model and graphically represented.

D3.2 Message Multiplicity. Looking at the message exchanges, several scenarios should be taken into account, as listed below.

1. A participant A sends one message to a participant B.
2. A participant A sends multiple messages to a participant B.
3. A participant A sends one message to multiple participants B.
4. A participant A sends multiple messages to each participant B.

Other scenarios might also be possible by reversing the direction of message flow. This distinction is important for a strict definition of choreographies and to avoid inconsistencies when participant processes are implemented. The point here is that BPMN 2.0 captures the concept of *Participant Multiplicity*, but does not capture the concept of *Message Multiplicity*. This requirement is implicitly supported by combining multi-instance activities (*MultiInstanceLoopCharacteristics*) and multi-instance participants (*Participant Multiplicity*) in collaboration and choreography diagrams. Figure 10 illustrates the different scenarios that can be supported where multi-instance is indicated with three parallel lines. The figure also shows the local enforceability [61] from the choreography diagram to the collaboration diagram. For example, in Scenario 2, the multi-instance choreography activity enforces a multi-instance "send" activity defined for *Participant A* and a multi-instance "receive" activity defined for *Participant B*.

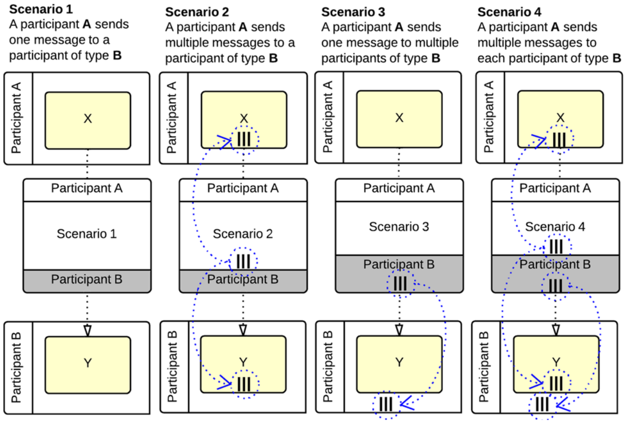
Scenario 1
A participant A sends one message to a participant of type B
Scenario 2
A participant A sends multiple messages to a participant of type B
Scenario 3
A participant A sends one message to multiple participants of type B
Scenario 4
A participant A sends multiple messages to each participant of type B
Participant A
X
Participant A
Scenario 1
Participant B
Participant B
Y
Scenario 2
Scenario 3
Scenario 4

Fig. 10 Message exchange scenarios

The problem here is that support for this requirement should be related to the concept of *Message*. Perhaps the concept of *Message Flow* should be extended, so that the message multiplicity can be made visible in the diagram [S-D3.2]. Our second evaluation shows that just by introducing a UML-like notation defining the cardinality of the message flow, the support for this requirement could be substantially increased. This may be more appropriate than depending on

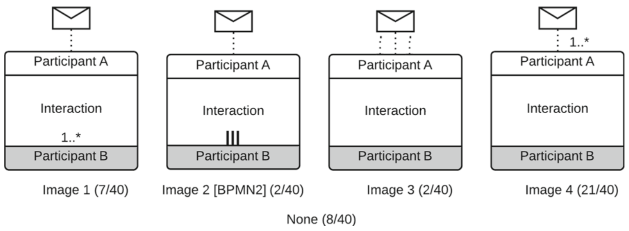

Fig. 11 EVAL. [Message Multiplicity]—Which representations do you think best capture the scenario: "Participant A sends multiple messages to Participant B"

ParticipantMultiplicity and *MultiInstanceLoopCharacteristics*. In the evaluation, 21 out of 40 subjects considered that including UML-like multiplicity attached to message flows represents better "message multiplicity" than three vertical lines (2 out of 40 subjects) (see Fig. 11).

It may be concluded that this requirement is only partially supported in the XML and therefore graphically as well.

D3.3 Service / Message Correlation. BPMN 2.0 uses the mechanism of *Correlation Key* to link messages with process instances. A *Conversation* (Fig. 1a) represents a set of *Message Flows* grouped together based on a concept and/or a correlation key. Correlation can be applied to *Message Flows* to associate a particular *Conversation* with a particular *Message* instance. Figure 12 shows that basically, a *CorrelationKey* represents a composite key consisting of one or many *CorrelationProperties* (partial keys). For each *Message* exchanged as part of a *Conversation*, the *CorrelationProperty* provides a *CorrelationPropertyRetrievalExpression* which refers to a *FormalExpression* defining how to extract the *CorrelationProperty* from the *Message*. Both the *CorrelationKey* of the *Conversation* and the *Message* should match in order to be considered associated. It is, however, not clearly specified in the standard how this correlation is graphically represented. Semantically, several correlation keys could be related to a choreography, but explicit representation is not considered. Nevertheless, *Function-Based* correlation patterns [4] (e.g., property-based correlation) could be explicitly represented in choreography diagrams so as to clarify possible labeling or special treatment of a group of messages [S-D3.3]. This requirement is rated as fully supported at the XML level but not supported graphically.

D3.4 Selection of Services and Reference Passing. One potential deficiency in BPMN relates to the absence of a *channel* concept as in WS-CDL [56]. A channel is a reference that allows one participant to communicate with another participant. Importantly, channels can be created by one participant and passed to other participants, thereby enabling dynamic message destinations [56]. For example, in Fig. 1 a *Customer* could create a channel to exchange delivery information. In this scenario, the *Customer* sends the channel to the *Manufacturer* who forward it to the *Seller*. The *Seller* would then be able to send delivery information directly to the *Customer* using the channel originally created by the *Customer*. Potentially, the customer could communicate with multiple sellers and give a separate response channel to each of them.

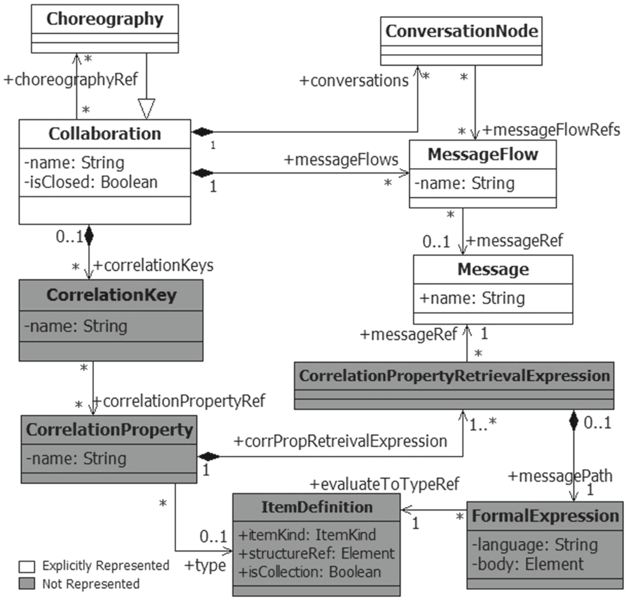

Fig. 12 Key-based Correlation in BPMN 2.0

The concepts of *Correlation Key* and *Participant References* in BPMN 2.0 address this need to some extent on the XML level. However, these concepts constitute one particular way of implementing a channel, based on the idea that every message contains an explicit identifier. A possible way of addressing this deficiency in BPMN might be to replace *Correlation Key* and *Participant References* with a single concept of channel [S-D3.4]. In this extension, interactions in a BPMN choreography diagram would have channels attached to them in order to capture channel creation, passing and use. Additionally, channel details could be captured using textual annotations, following the principle of *Dual Coding* [40]. Reference passing could be supported graphically in BPMN collaboration diagrams as shown in [18]. An informal way of indicating that a reference is passed through a *Message* is by natural language descriptions in *Documentation* elements.

4.4 D4. Service Interaction Patterns

The service interaction patterns [5] are a catalog of common scenarios in choreographies. They are commonly used as a benchmark for assessing choreography languages [2, 17, 18, 61]. Although they are an important guide, they are limited to the choreography domain perspective. They take into account neither important technical requirements such as formalism nor comprehensibility aspects such as the graphical notation and the model quality considered in this evaluation. The 13 patterns presented in [5] are classified in four general types. Table 4 presents the four categories of patterns.

It is not a prior aim of this article to analyze in detail the BPMN 2.0 support for the 13 patterns. The support for most of these patterns depends on the combination of several requirements previously evaluated. For example, a dependence exists between the support for *Multi-transmission Interaction Patterns (D4.3)* and support for previously analyzed requirements such as *Multi-party Collaboration(D1.1)*, *Service Sets (D1.4)*, *Control Flow Definition (D3.1)* and *Message Multiplicity(D3.2)* among others.

D4.1 Single-Transmission Bilateral Interaction Patterns and D4.2 Single-Transmission Multilateral Interaction Patterns. The first two group of patterns are quite basic scenarios such as a send or receive between different partners. They both depend on well supported requirements in BPMN 2.0 such as *D1. Multi-party Collaboration*, *D1.4 Service Sets* and *D3.1 Control Flow Definition* so they are evaluated as fully supported at both graphical and XML Schema levels.

D4.3 Multi-Transmission Interaction Patterns. Poor support was previously detected for *D3.2 Message Multiplicity*, used to capture the definition of the number of messages sent from one (or more) participant(s) to other(s). This deficiency will avoid full support for the so-called *Multi-transmission Interaction Patterns*. Concretely, *contingent requests* and *atomic-multicast notifications* (Pattern 9 and 10 [5]) are not supported in BPMN 2.0. A better support for message multiplicity (D3.2) should help capture these patterns [S-D4.3].

D4.4 Routing Patterns. Partial support for *D3.4 Reference Passing* was also discussed, where participant A allows participant C to communicate with participant B by passing the reference of B to C. This will avoid fulfilling the so-called *Routing Patterns*. Concretely, *dynamic routing* (Pattern 13 [5]) is not supported. A better support for reference passing (D3.4) should help capture these patterns [S-D4.4].

4.5 Summary: domain requirements

Table 5 summarizes the ratings for each general requirement, and Table 6 summarizes the suggestions given for each requirement. The global support of the axis is calculated as a percentage average of the sub-requirements. Note that there is a great difference between support at the XML level compared to the graphical level. It could be argued that this trend is logical as a more detailed specification is obtained. However, the support for these domain requirements is also considered critical in higher levels of abstraction because a precise definition at the graphical level could avoid major problems of interoperability between participants. In particular, critical might be when further refining to obtain a more technical specification of the choreography is the target.

Table 5 Domain requirements recap

Requirement	Graphical (%)	XML Schema (%)
D1. Participant Specification	62.5	100
D2. Context Description	70	70
D3. Service Communication	37.5	75
D4. Service Interaction Patterns	62.5	75
Avg. percentage	58.13	80

4.5.1 Strong points

Note that there is remarkably good support for **Participant Specification (D1)** requirements. The problem here is that this good support is somehow blurred in the graphical representation. There is also a good support for **Context Description (D2)** requirements (70 %). In this case, graphical support is well balanced with the XML Schema support. Note that there is full support for *Time Constraints (D2.3)*, *Exception Handling (D2.4)* and *Data-Oriented Process Definition (D2.5)*. **Service Communication (D3)** is much better supported at the XML Schema level. The major differences concern *Message Correlation (D3.3)* and *Reference Passing (D3.4)*. The first groups of **Service interaction patterns (D4)** namely *Single-Transmission Bilateral Interaction Patterns (D4.1)* and *Single-Transmission Multilateral Interaction Patterns (D4.2)* are fully supported.

4.5.2 Weak points

D1. Participant Specification. The major problem of BPMN 2.0 in this axis is that the XML Schema implements useful information that should also be represented graphically. Therefore, modeling choreographies with the aim of raising a technical level becomes tedious and error prone. To improve the *Role Modeling (D1.2)* requirement, the *Role* and the *Participant* constructs should be differentiated explicitly. For a better support of *Role Mapping (D1.3)*, the *Participant Association* construct may be explicitly illustrated in the diagram by means of a text annotation for example. To achieve full support of *Service Sets (D1.4)*, the language should allow modelers to specify graphically the number of participants that take part in the choreography.

D2. Context Description. In the context description area, *Semantic Constraint Specification (D2.1)* may be improved extending the constructs *Expression* or *Textual Annotations*.

Table 6 Summary of suggestions for the domain appropriateness requirements

Requirement	Suggestion	Suggestion description
D1. Participant Specification		
D1.1 Multi-party Collaborations		*Good Support—No Suggestion*
D1.2 Role Modeling	S-D1.2	Ideally, the label appearing in the participant's band should have only a single meaning. *PartnerRole*, *PartnerEntity* and *Participant* should be differentiated explicitly in a graphical way to avoid misunderstandings
D1.3 Role Mapping	S-D1.3	A possible solution may be to label a participant with a textual annotation, a reference in the name's label or an icon to clarify the *ParticipantAssociation* and then support this requirement also in the diagram
D1.4 Service Sets	S-D1.4	Perhaps modelers could be permitted to clarify the number of participants taking part in the choreography graphically
D2. Context Description		
D2.1 Semantic Constraint Specification	S-D2.1	If a *Text Annotation* construct was related in some way to the *Expression* construct or *Formal Expression* construct (inherits from *Expression*), a modeler could easily give formalism to semantic constraints and then better support this requirement
D2.2 Pre-/Post-Conditions of Process/Task Executions	S-D2.2, S-D2.2′	*DataObjets* constructs could potentially support the requirement, but they may have to be adapted for choreographies and not restricted to process diagrams. Also, a link to goal-oriented languages such as KAOS [31], where formal verifications on pre- and post-conditions are made, could be a possible way to help support this requirement
D2.3 Time Constraints		*Good Support—No Suggestion*
D2.4 Exception Handling	S-D2.4	A possible improvement to better support for this requirement is to graphically differentiate the choreography messages caused by an exception from the rest of the messages. Currently, no distinction is made
D2.5 Data-Oriented Process Definition		*Good Support—No Suggestion*
D3. Service Communication		
D3.1 Control Flow Definition		*Good Support—No Suggestion*
D3.2 Message Multiplicity	S-D3.2	The concept of *Message Flow* may be extended, so that the message multiplicity is visible in the diagram. Graphically, a single message flow line or a triple line (to be coherent with the multi-instance marker) may help to differenciate between a single or a multiple message exchange
D3.3 Service/Message Correlation	S-D3.3	*Function-based* correlation patterns [4] (e.g., property-based correlation) could be explicitly represented in choreography diagrams so as to clarify possible labeling or give special treatment to a group of messages
D3.4 Selection of Services and Reference Passing	S-D3.4	A possible way of providing better support for this requirement might be to replace *Participant References/ Correlation Keys* with a single concept of channel similar to WS-CDL [56]
D4. Service interaction patterns		
D4.1 Single-Transmission Bilateral Interaction Patterns		*Good Support—No Suggestion*
D4.2 Single-Transmission Multilateral Interaction Patterns		*Good Support—No Suggestion*
D4.3 Multi-transmission Interaction Patterns	S-D4.3	Better support for message multiplicity (D3.2) should help capture these patterns
D4.4 Routing Patterns	S-D4.4	Better support for reference passing (D3.4) should help capture these patterns

Two suggestions to support *Pre-/Post-Conditions (D2.2)* are proposed: First, *dataObjects* constructs to should be adapted to the choreography; and secondly, BPMN 2.0 should provide a means of linking up to a goal-oriented language such as KAOS [31]. *Exception Handling (D2.4)* could be better supported if messages caused by an exception are differentiated graphically.

D3. Service Communication. Requirements related to service communication may be much improved in the graphical aspect. Extending the *Message Flow* construct to visualize message multiplicity may help support *Message Multiplicity (D3.2)*. *Service/ Message Correlation (D3.3)* support may be improved by labeling or giving special treatment to correlated messages taking into account correlation patterns [4]. For the *Selection of Services and Reference Passing (D3.4)*, BPMN 2.0 may adopt the concept of *Channel* used in WS-CDL.

D4. Service Interaction Patterns. To achieve better support for service interaction patterns, BPMN 2.0 needs to improve the support for *Message Multiplicity (D3.2)*. This may imply better support for *Multi-Transmission Interaction Patterns (D4.3)*. *Routing Patterns (D4.4)* may be better supported if better support for the *Reference Passing (D3.4)* requirement requirement is managed.

Suggestions for the domain appropriateness requirements are summarized in Table 6.

5 Comprehensibility requirements evaluation

Figure 13 gives an overview of the evaluated requirements. Comprehensibility is analyzed separately when regarding semantics and syntactics. Language semantics focuses on the choreography meta-model. Management of complexity and presentation are analyzed as well as some important style guidelines taken from Ambler's work [1]. The analysis of language syntactics relies on the *Graphical Notation Principles* [40].

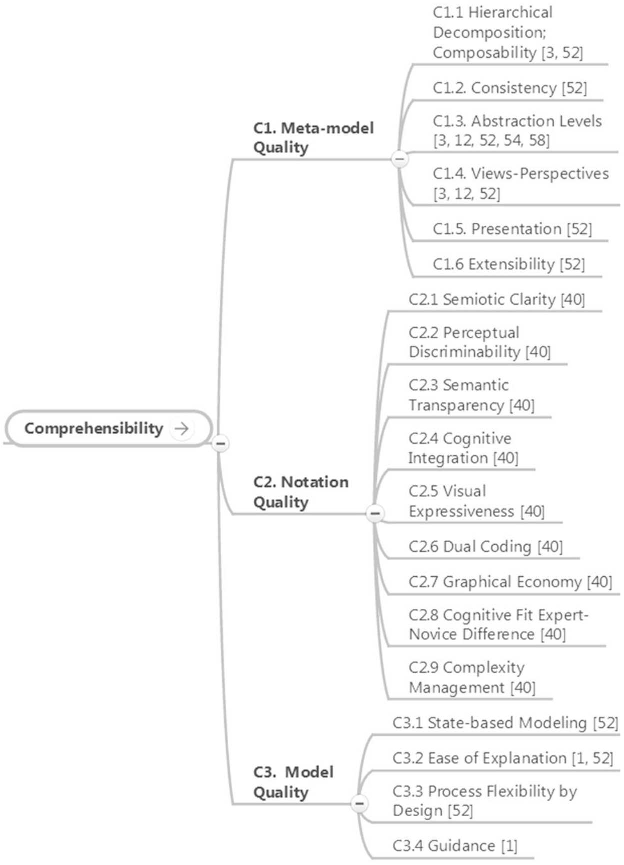
Comprehensibility
C1. Meta-model Quality
C1.1 Hierarchical Decomposition; Composability [3, 52]
C1.2. Consistency [52]
C1.3. Abstraction Levels [3, 12, 52, 54, 58]
C1.4. Views-Perspectives [3, 12, 52]
C1.5. Presentation [52]
C1.6 Extensibility [52]
C2. Notation Quality
C2.1 Semiotic Clarity [40]
C2.2 Perceptual Discriminability [40]
C2.3 Semantic Transparency [40]
C2.4 Cognitive Integration [40]
C2.5 Visual Expressiveness [40]
C2.6 Dual Coding [40]
C2.7 Graphical Economy [40]
C2.8 Cognitive Fit Expert-Novice Difference [40]
C2.9 Complexity Management [40]
C3. Model Quality
C3.1 State-based Modeling [52]
C3.2 Ease of Explanation [1, 52]
C3.3 Process Flexibility by Design [52]
C3.4 Guidance [1]

Fig. 13 Overview of comprehensibility requirements

As already mentioned in Sect. 3.5, an empirical study was carried out and explained in a previous article [11]. This helped evaluate meta-model quality (C1), notation quality (C2) and various points of the model quality (C3). Points such as readability, completeness, complexity and ease of explanation were analyzed by subjects in a concrete scenario modeled with both graphical notations. This evaluation was complemented by an online questionnaire that helped us validate certain improvement proposals presented and discussed in this section.

5.1 C1. Meta-model Quality

Here, the BPMN 2.0 meta-model quality is analyzed according to several criteria, which are summarized in Table 7. Note that, in this respect, the focus is placed only on the meta-model and support for the different requirements for the graphical part in not therefore examined. Table 7 defines the analyzed column as "Meta-model" and not "XML-Schema" as in the previous tables. This is motivated by the fact that this table looks at the meta-model itself and not the way it is captured.

C1.1 Hierarchical Decomposition; Composability. A mechanism to break down an overall complex model into more manageable parts is supported in BPMN 2.0 through the concept of *sub-choreography* and *sub-process*. This mechanism can be used break down models into smaller parts. This is a powerful mechanism that should be used to improve comprehensibility. Therefore, there is full support for this requirement.

C1.2 Consistency. Uniformity between the different models representing the system is achieved in BPMN 2.0. Choreographies in BPMN 2.0 inherit from collaboration diagrams as can be seen in the meta-model illustrated in Fig. 2. This fact facilitates integration as they share most of their constructs (e.g., *Participant* or *Message*). Conversation diagrams are also consistent with collaborations given that conversation nodes represent a set of message flows. Therefore, this requirement is also fully supported in the XML Schema level. Here the aim is to look at the consistency from the meta-model point of view. Later, the requirement *C2.4 Cognitive Integration* in Sect. 5.2 looks at consistency from a graphical point of view.

C1.3 Abstraction Levels. BPMN 2.0 does not specify different dialects or levels of abstractions when defining choreographies. Subclasses (Descriptive, Analytic and Common Execution) are defined, but they only concern process modeling [45]. These subclasses correspond to the three-level models presented by Silver in [54], and they should be present in the standard to help construct choreographies. In a previous paper [11], the benefits of presenting the choreography concept (the meta-model) in three abstraction levels are discussed. In the latter paper, it is proposed to start from a con-

Table 7 Meta-model Quality Requirements

Refinement	Description	Meta-model	Suggestion
C1.1 Hierarchical Decomposition; Composability	A mechanism to break down an overall complex model into more manageable parts should be supported	2	
C1.2. Consistency	Uniformity between the different models that represent the system must be achieved	2	
C1.3. Abstraction Levels	Different levels of abstraction might be presented in the meta-model that may influence different abstraction levels in the notation. This gradual presentation would improve comprehensibility and understanding of the meta-model	1	S-C1.3
C1.4. Views-Perspectives	Different perspectives are identified and shown in different diagrams (e.g., structural and behavioral views)	1	S-C1.4
C1.5. Presentation	Meta-model diagrams (e.g., UML class diagrams) must be cognitive effective tools	1	S-C1.5
C1.6. Extensibility	The meta-model should be extensible so that new concepts may be introduced to maintain compliance with the standard	2	
		9/12 (75 %)	

C1. Meta-model Quality: A choreography language must allow modelers to understand and design choreographies gradually through their meta-models

ceptual level leading to an implementation (technical) level through an analysis level. This approach might help bridge the so-called Business-IT gap [58] when capturing service choreographies in a gradual way. It might be argued that the graphical notation and the XML Schema are two different abstraction levels in BPMN 2.0. Different levels of abstraction may be considered, providing more or less detail in the diagrams. But it is considered critical to define the levels of abstraction in the meta-model [S-C1.3]. The graphical notation would therefore be adapted to these levels. Therefore, this requirement is partially supported.

C1.4 Views-Perspectives. The support for different perspectives should also be a main concern when defining choreographies. In [3,17,20], the authors argue that different perspectives are required when modeling. In the standard, the term "perspective" is used to describe different diagrams (e.g., choreography and collaboration), but this concept is not clearly explained. The new *conversation diagram* could be used when defining choreographies. Figure 1a shows how it represents the manufacturing process of the example. The conversation diagram may be used since the *structural view* as it models relationships between participants. The choreography diagram and the collaboration diagram may be the *behavioral view*. Currently, the conversation diagram is not required for choreography conformance [45] but is recommend. In addition, a *stepwise refinement* [7] should be supported [S-C1.4]. This technique requires the modeler to define correspondence between structural and behavioral aspects within different abstraction levels. Therefore, this requirement is only partially supported.

C1.5 Presentation. When analyzing the meta-model, several guidelines described in [1,54] as well as the WS-CDL critical reviews [6] are taken into account. In the standard, numerous complex and ineffective meta-models are found. For example, the choreography meta-model has 18 elements and 25 relationships. The large number of elements in the diagram may produce a state of cognitive overload, where comprehension deteriorates rapidly [40]. In fact, it is not easy to understand the choreography concept just by regarding the meta-model despite the fact this should be one of its goals. The "style" of this meta-model class diagram focuses on symmetry, precise multiplicity definition, coherent sized elements and avoiding crossing lines. However, no attention is given to the use of color and packages, placing subclasses below super classes or avoiding redundant information. Such measures would simplify the user's understanding of the meta-model and make it much more effective [1,25]. Modularization [40], with a greater use of UML packages, might improve the meta-model presentation by effectively managing its complexity [S-C1.5]. Therefore, this requirement is again only partially supported.

In the questionnaire, the two main issues of the BPMN 2.0 meta-model (only the choreography and service meta-model were presented) were: too many relationships (28 out of 40) and too many classes (27 out of 40). This result highlights the meta-model's complexity. The approach presented in the authors' previous article [11] was followed to propose a three-level meta-model separating the structural and behavioral views to the subjects. All the concepts from the choreography and the service meta-models were retained. Packages were introduced (e.g., Service package), subclasses were placed below super classes, color was introduced and the brightness of classes changed depending on their level of abstraction. The subjects were then asked what they considered to be the changes that might enhance the user's understanding of the "new" meta-model. Color was considered the most helpful (23 out of 40), although the absence of a legend makes it difficult to understand the semantics. Sepa-

Table 8 Notation Quality Requirements

Refinement	Description	Graphical	Suggestion
C2.1 Semiotic Clarity	One-to-one correspondence between symbols and concepts	1	S-C2.1
C2.2 Perceptual Discriminability	The ease and accuracy with which different symbols are clearly distinguishable from each other	1	S-C2.2, S-C2.2′
C2.3 Semantic Transparency	Visual representation whose appearance suggests their meaning	1	S-C2.3
C2.4 Cognitive Integration	Inclusion of explicit mechanisms to support integration of information from different diagrams	1	S-C2.4, S-C2.4′
C2.5 Visual Expressiveness	Use the full range and capacities of visual variables (shape, texture, brightness, size, color and orientation)	1	S-C2.5
C2.6 Dual Coding	Use text to complement graphics. Textual encoding is most effective when it is used in a supporting role	2	
C2.7 Graphical Economy	The number of different graphical symbols has to be cognitively manageable	2	S-C2.7
C2.8 Cognitive Fit—Expert-Novice Difference	Take into account the difference between practitioners and novices	1	S-C2.8
C2.9 Complexity Management	Include explicit mechanisms for dealing with complexity	2	
		12/18 (66.67 %)	

C2. Notation Quality: The graphical notation must be optimized for human communication and problem solving. A scientific background should guide graphical choices

ration of views (21 out of 40), inheritance orientation (21 out of 40), packages (17 out of 40) and levels of abstraction (17 out of 40) were also appreciated. Nevertheless, the general view given by the choreography meta-model in BPMN 2.0 was missed in the modified meta-model. The use of bigger classes for major concepts such as *Choreography*, *Participant* or *Choreography Activity* was also highlighted as a good point in the BPMN 2.0 meta-model.

C1.6 Extensibility. The BPMN meta-model is intended to be extensible. This allows BPMN adopters to add new constructs to the specified meta-model. In addition, this property allows these new constructs to remain BPMN-compliant. The standard provides a set of extension elements, allowing modelers to attach additional attributes and elements to the standard BPMN elements. Therefore, full support is provided.

5.2 C2. Notation Quality

This section gives a detailed evaluation of the different notation principles [40] in order to analyze the notation quality of BPMN 2.0. However, the large number of graphical constructs makes the work difficult. Genon et al. [27] evaluated the cognitive effectiveness of BPMN 2.0 process models. This work can also be applied in our evaluation because choreography models and process models have common constructs. However, it is still difficult to give objective results. Valuable metrics could be defined to compare the BPMN 2.0 features with other languages. One of the major problems detected in [27] is the lack of a scientific background in notation design. This problem also applies to the choreography subset of BPMN 2.0.

Table 8 shows the descriptions and ratings for the different requirements. Note that only the graphical part of BPMN 2.0 has been considered.

C2.1 Semiotic Clarity. As seen in previous sections, there is not always one-to-one correspondence between symbols and concepts. For example, the service correlation construct or the participant's multiplicity construct is simply declared in an implementation level. In other cases, such as timer event, this requirement is fulfilled. On the other hand, there is the example of *Participant* or the *Exclusive Gateway* where more than one graphical construct is defined. Figure 1 shows three different graphical representations of *Participant* depending on the diagram. Therefore, this requirement is partially supported. To reduce the semantic gap between the meta-model and the diagrams, the meta-model could be simplified or a more refined notation for choreographies could be proposed. This difficult task could be facilitated by defining of several abstraction levels (requirement C1.3). An easier task might be to avoid redundancy, defining at most one graphical construct for each meta-class [S-C2.1].

C2.2 Perceptual Discriminability. The perceptual discriminability can also be criticized. The difference between call choreographies and sub-choreographies by line width is an example of this deficiency. Different visual variables such as shape, texture, brightness, size, color and orientation should be used [40]. However, BPMN 2.0 relies on shape, texture and on a few occasions color (e.g., to distinguish between the initiator participant and the receiver participant in an interaction).

Another problem arises when looking at the concept of message flow, which in BPMN is overloaded. Specifically, if one participant creates an instance of another participant ("instantiation"), a message flow will be used. But if one participant wants to send a message to an existing instance of another participant, a message flow will be also used. So in one case, the concept of message flow allows the user to create an instance of a service, while in the other case it allows the user to send a message to an existing instance of a service. This issue is compounded by the fact that the *Conversation Node* symbol (the hexagon in Fig. 1a) does not indicate which participant initiates a conversation. One possible way to address this issue is to introduce a special annotation that can be attached to message flows to indicate that they correspond to an "instantiation." This lack of perceptual discrimination in BPMN 2.0 might lead to errors, particularly in the case of choreographies that involve more than two participants, since it will not be clear then which participants come first in the choreography, and which ones come later. Taking for example the conversation diagram of Fig. 1a, how can one tell which participant initiates the overall conversation? Given our knowledge of the domain, intuition tell us that the collaboration was started by the *Customer*, but this is not apparent from the diagram. Therefore, this requirement is only partially supported. More visual variables (shape, texture, brightness, size, color and orientation) should be used as BPMN 2.0 relies basically on shape and texture and occasionally on in color [S-C2.2]. The distinction between graphical constructs should not be based on the stakeholders' assumed knowledge of the domain [S-C2.2'].

C2.3 Semantic Transparency. Visual representation should suggest their meaning. In some cases, the notation is intuitive, e.g., envelopes are used for symbolizing message events and clocks are used for symbolizing timer events. In other cases, especially those regarding minimal differences between process activities and choreography activities, less transparency is achieved. Also, there is no distinction between error messages and other messages. As can be seen in Fig. 1, *Participant* is another clear example of lack of transparency as it is impossible to deduce the meaning of the graphical construct by just looking at the representation. This requirement is therefore only partially supported. More intuitive symbols should be provided, especially to distinguish between *process activities* and *choreography activities* and to represent *Participants* [S-C2.3].

C2.4 Cognitive Integration. The integration between the different BPMN diagrams is not always intuitive, in particular between collaboration and choreography diagrams (Fig 1). In choreography diagrams, the sending participant band is shaded in white and the receiving participant band is shaded in gray. In collaboration diagrams, the sending event is marked with a black envelope, whereas the receiving event is marked with a white envelope. Thus, gray corresponds to white and white corresponds to black. It is not always evident how to match send tasks in collaborations to response messages in choreography tasks.

A difficulty was also noted when labeling choreography tasks. BPMN 2.0 does not impose limits on the modeler, but this freedom may lead to inconsistencies between the choreography and the collaboration diagrams. In Fig. 1 note that the choreography task is named in the same way as the send activity in the collaboration diagram. This or other conventions might improve semantic consistency between both diagrams. In the questionnaire, a scale was proposed so that subjects could mark the appropriateness to use this naming method for choreography tasks. The scale ranges from 1 (Not at all) to 5 (Absolutely). In the evaluation, 27 out 40 subjects considered it appropriate (4 and 5 in the scale). Also, 33 out of 40 subjects considered it important to use some sort of naming convention when modeling (standard conventions, organizational conventions or own conventions). However, different habits were identified concerning naming conventions so it was not possible to state that they should be included in the standard. It seems to be positive to use naming conventions, but it is not clear where the limits should be defined. Perhaps just by defining the structure (for example:"interactions should be named with a verb + noun") may help modelers and stakeholders. A clear position on interaction naming in the standard examples may be desirable to clarify the cognitive integration between collaboration [S-C2.4]. A better consistency between colors of the send and receive activity icons and their corresponding message objects may also be introduced [S-C2.4']. It is concluded that only partial support is achieved for this requirement.

C2.5 Visual Expressiveness. This requirement is strongly related to *C2.2 Perceptual Discriminability*. Visual expressiveness relies on visual variables (shape, texture, brightness, size, color and orientation). The standard mainly focuses on shape, texture and occasionally color. The other variables depend on implementers as the standard suggests flexibility in this aspect. In general, tools add colors and different textures to their palette of constructs to improve the expressiveness of the language. A very common convention is to represent the process *Start Event* in green and the *End Event* in red inspired by traffic lights [27]. It is recommended that common tool conventions be grouped so they could be standardized. More attention to color should be given, as it is one of the most cognitively effective of all visual variables [40][S-C2.5]. This requirement was therefore considered to be partially supported. A better interoperability between tools might be reached if all the tools will use the same visual variables.

C2.6 Dual Coding. Dual coding is fully supported and recommended. It will also depend on modelers and implementations. This can be supported by adding textual annotations to the diagram.

C2.7 Graphical Economy. The number of graphical symbols is cognitively manageable as the use of event and gateway symbols is considerably reduced with respect to process diagrams. As already mentioned, in previous evaluation of BPMN, Wohed et al. [60] and Recker et al. [47] highlighted a number of construct redundancies and overloads. This issue is reduced if the focus is put choreography. This requirement is considered to be fully supported. However, it is recommended that the limit between the constructs that might be used in *process diagrams* and *collaboration diagrams* [S-C2.7] be explicitly defined. This may reduce ambiguity.

C2.8 Cognitive Fit (Expert-Novice Fit). The choreography notation is simpler than the process notation. Therefore, there is perhaps no need for multiple dialects as proposed by Moody [40]. However, the use and integration of several diagrams might be difficult when dealing with complex choreographies. Hence, this requirement is considered partially supported. A more conceptual layer should be defined where just the basic elements of the choreography could be represented [S-C2.8]. For instance, just the conversation diagram and the choreography diagram would be used. This need is discussed in previous studies [10].

C2.9 Complexity Management. BPMN 2.0 includes explicit mechanisms for dealing with complexity. It allows the user to declare sub-choreographies that can be expanded or collapsed. Figure 14 shows how the choreography activity *Procure Parts* is collapsed (Fig. 14a) and could also be expanded (Fig. 14b). Another method that could be applied to simplify models is to use *Link Events*, which is a way of connecting two sections of a process. Paired *Link Events* can also be used to connect a printed diagram across multiple pages. This requirement is therefore rated as fully supported.

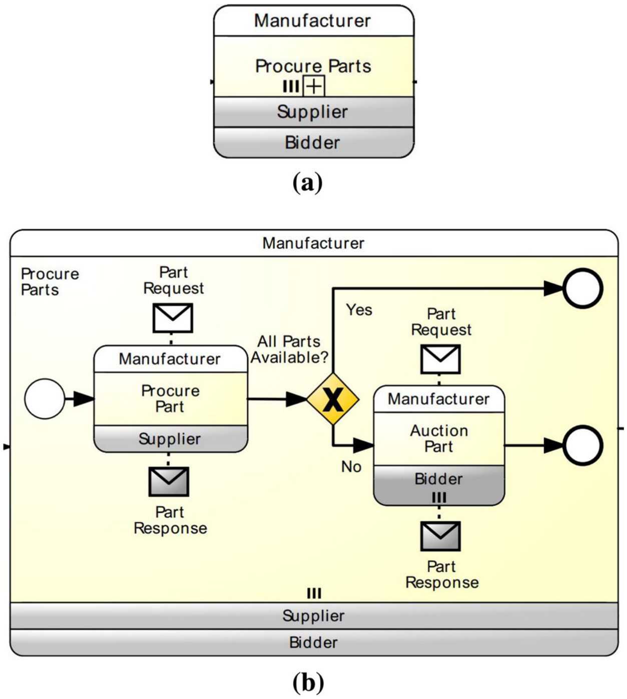

Fig. 14 Complexity Management with Sub-choreographies in BPMN 2.0. **a** Choreography activity collapsed. **b** Choreography activity expanded

5.3 C3. Model Quality

This section analyzes graphical models as well as the XML Schema models in BPMN 2.0. Table 9 provides the descriptions, and ratings for the different criteria are used to evaluate model quality support.

C3.1 State-based Modeling. The relation between message exchanges and state changes is not captured explicitly in BPMN 2.0. But thanks to the language permissiveness, this model can be partially supported using a (valid but aberrant) process diagram comprising only *Events*. These models may represent the sequence of states to be achieved by the process. An example of this approach is given by the so-called high-level behavioral diagrams in [16]. A possible solution to this failing is to explicitly define a diagram to support state modeling based on events as Fig. 15 shows [S-C3.1].

C3.2 Ease of Explanation. BPMN 2.0 proposes a graphical notation whose aim is to achieve communication between different audiences and skills. However, more technical details captured in the XML Schema would be much harder to understand for a non-specialist. This requirement may depend on the readability and the simplicity of models [1]. Note that this characteristics will mostly rely on the modeler's skills as well as the tools. However, the standard provides mechanisms such as *Sub-choreographies* or *Link events* that may facilitate the explanation [45]. A more refined graphical notation such as that presented in [51] might be a solution to make the technical specifications more comprehensible. This refined notation may capture more technical constructs such as services, operations, correlations, etc., in a visual way [S-C3.2].

C3.3 Process Flexibility by Design. All possible flows of a collaborative process can be explicitly represented in BPMN 2.0 as well as exceptional paths. *Gateways* in BPMN 2.0 can be used to capture all the possible paths in the choreography. For instance, AND/OR forks as well as AND/OR joins are supported graphically. Therefore, this requirement is fully supported in BPMN 2.0.

Table 9 Model Quality Requirements

Refinement	Description	Graphical	XML Schema	Suggestion
C3.1 State-based Modeling	The relation between message exchanges and state changes (as representation of collaboration progress) should be modeled explicitly	1	1	S-C3.1
C3.2 Ease of Explanation	The need for communication among unequal personnel is an important feature especially in high-level layers to achieve agreement among business personnel of different organizations	2	0	S-C3.2
C3.3 Process Flexibility by Design	All possible flows of a process model explicitly using typost-condition definition elements provided by the modeling language like AND/OR forks, AND/OR joins and derived constructs should be supported. Exceptional paths might also be captured	2	2	
C3.4 Guidance	Guidance should be given in order to encourage useful models	0	0	S-C3.4
		5/8 (62.5 %)	3/8 (37.5 %)	

C3. Model Quality: A language must permit modelers to produce valid and effective models

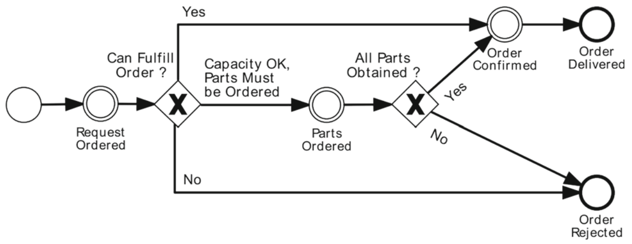

Fig. 15 Potential state-based model in BPMN 2.0

Table 10 Comprehensibility requirements recap

Requirement	Graphical (%)	XML Schema (%)
C1. Meta-model Quality	–	75
C2. Notation Quality	66.67	–
C3. Model Quality	62.5	37.5
Avg. Percentage	64.58	56.25

C3.4 Guidance. The standard lacks guidelines to produce "good" models. All diagrams can be combined and fulfilled at different levels of detail. Very complex as well as excessively simple diagrams can be produced. This flexibility and the lack of orientation could be disturbing factors for modelers especially for novices. More guidance, examples and methodology are needed in the standard [S-C3.4].

5.4 Summary: comprehensibility requirements

Despite the fact that BPMN 2.0 is aimed at being human centered, with a rich graphical notation, major deficiencies were found that may lead to misunderstandings. Table 10 summarizes the support for comprehensibility requirements in BPMN 2.0, and Table 11 summarizes the suggestions given for each requirement.

5.4.1 Strong points

Looking at **Meta-model Quality (C1)**, the *Composability (C1.1)*, the *Consistency (C1.2)* between the main elements for modeling choreographies, as well as the *Extensibility Mechanisms (C3.6)* are the best supported points. The analysis of the **Notation Quality (C2)** indicated good support for *Dual Coding (C2.6)* thanks to textual annotations, a good *Graphical Economy (C2.7)*, and effective mechanisms for *Complexity Management (C2.9)* such as sub-choreographies. In the empirical evaluation described at the beginning of this section, the subjects highlighted the *Ease of Explanation (C3.2)* of BPMN 2.0 choreography diagrams. This characteristic and the *Flexibility by Design (C3.3)* are remarkable features of BPMN 2.0 with respect to **Model Quality (C3)**.

5.4.2 Weak points

C1. Meta-model Quality. A major deficiency was detected in the meta-model quality. A meta-model should be a useful tool for communication in addition to a technical description of a language. The way in which meta-models are introduced in BPMN 2.0 hinders the understanding of choreographies because they are presented in a highly technical manner. A more abstract meta-model level capturing a more conceptual layer would drastically facilitate the understanding

Table 11 Summary of suggestions for the comprehensibility requirements

Requirement	Suggestion	Suggestion description
C1. Meta-model Quality		
C1.1 Hierarchical Decomposition; Composability		*Good Support—No Suggestion*
C1.2. Consistency		*Good Support—No Suggestion*
C1.3. Abstraction Levels	S-C1.3	It is of critical importance to explain the levels of abstraction in the meta-model. The graphical notation would therefore be adapted to these levels
C1.4. Views-Perspectives	S-C1.4	Currently, the conversation diagram is not required for choreography conformance [45], but it is recommended. In addition, a *stepwise refinement* [7] should be supported
C1.5. Presentation	S-C1.5	The use of style guidelines and visual variables (shape, texture, brightness, size, color and orientation) could make the meta-model much more effective [1,25]. Modularization [40] (using UML packages) might help manage its complexity
C1.6. Extensibility		*Good Support—No Suggestion*
C2. Notation Quality		
C2.1 Semiotic Clarity	S-C2.1	The semantic gap between the meta-model and the diagrams should be reduced by simplifying the meta-model or proposing a more refined notation that may capture technical concepts. Redundancy should be avoided, defining at most one graphical construct for each meta-class
C2.2 Perceptual Discriminability	S-C2.2, S-C2.2’	More visual variables should be used to distinguish BPMN 2.0 constructs which rely basically on shape, texture and occasionally color. Also, the distinction between graphical constructs should not be based on the stakeholders’ assumed knowledge of the domain
C2.3 Semantic Transparency	S-C2.3	More intuitive symbols should be provided, especially to distinguish between process and choreography activities and to represent *Participants*
C2.4 Cognitive Integration	S-C2.4, S-C2.4’	A clear position on the interaction naming within the standard examples may be desirable to clarify the cognitive integration between collaboration and choreography diagrams. Better consistency between the color of the send and receive activity icons and their corresponding message object may also be introduced
C2.5 Visual Expressiveness	S-C2.5	It is assumed that common tool conventions be grouped together so that they can be standardized. More attention to color should be given, as it is one of the most cognitively effective of all visual variables
C2.6 Dual Coding		*Good Support—No Suggestion*
C2.7 Graphical Economy	S-C2.7	It is recommended that limit between the constructs that might be used in *process diagrams* and *collaboration diagrams* be clearly defined. This may reduce ambiguity
C2.8 Cognitive Fit—Expert-Novice Difference	S-C2.8	It is recommended that a more conceptual layer be defined where just the basic elements of the choreography might be represented. For instance, just the conversation diagram and the choreography diagram would be used
C2.9 Complexity Management		*Good Support—No Suggestion*
C3. Model Quality		
C3.1 State-based Modeling	S-C3.1	A diagram should be defined to support state modeling based on events as Fig. 15 shows
C3.2 Ease of Explanation	S-C3.2	A more refined graphical notation such that presented in [51] might be a solution to make the technical specifications more comprehensible
C3.3 Process Flexibility by Design		*Good Support—No Suggestion*
C3.4 Guidance	S-C3.4	BPEL4Chor is an example of a choreography language that gives support for this requirement [17]. More guidance, examples and methodology are needed in the standard

of the meta-model and therefore the language. By presenting the meta-model at different *Abstraction Levels (C1.3)*, it would be possible to adapt the language to different audiences and formalize gradual refinements for the graphical notation. Also, the addition of a refined graphical notation may improve the ease of explanation of technical (XML) models. Another important failing in this area is the fact that the standard does not force the use of conversation diagrams to reach choreography conformance. As a result, the static view of choreography is underestimated. In a previous publication [11], the need for a clear separation of the structural and behavioral views in choreographies (*Views-Perspectives (C1.4)*) was identified. BPMN 2.0 may adopt conversation diagrams as the structural view for choreography, leaving choreography and collaboration diagrams as the behavioral views. A stepwise refinement [7] could then be used to define the correspondence between abstraction layers and views. To improve the *Presentation (C1.5)* of the meta-model, it is recommended to use color, to place subclasses below subclasses and avoid redundant elements.

C2. Graphical notation. Looking at the graphical notation design the lack of a scientific background in many choices is confirmed. Hence, most of the principles are only partially supported. To achieve better support for *Semiotic Clarity (C2.1)*, the standard should avoid redundant constructs and reduce the semantic gap between the XML Schema and the graphical notation by refining the graphical notation (potentially adding an explicit level of abstraction). *Perceptual Discriminability (C2.2)* could be improved by using more visual variables (shape, color, texture, size, brightness, orientation) and avoiding the dependence on domain knowledge to distinguish graphical constructs. *Semantic Transparency (C2.3)* could be attained if the distinction between process and choreography activities and the participant's representation is enhanced. To achieve better support for *Cognitive Integration (C2.4)*, two main recommendations are made: force the way modelers should label elements such as choreography tasks; and improve the graphical coherence between send and receive tasks with their respective input or output messages. The *Visual Expressiveness (C2.5)* can be improved by gathering common style conventions used in tools to standardize element texture and color in the notation. Looking at *Graphical Economy (C2.6)*, even full support is considered, it may be improved by clearly delimiting the constructs used in process and collaboration diagrams. Finally, BPMN 2.0 should support *Cognitive Fit – Expert-Novice Difference (C2.8)*. To achieve this requirement, a conceptual level should be introduced represented only by the conversation and choreography diagrams.

C3. Model Quality. Major failings were also noted in the model quality area. Concerning *State-based Modeling (C3.1)*, a model consisting only of events should be defined. This model gives a high-level behavioral view of the choreography illustrating the states to be achieved. To improve the *Ease of Explanation (C3.2)*, especially of technical choreographies, a more refined graphical notation need to be defined in order to make the technical specification, currently implemented using XML Schemas, more comprehensible. For the *Guidance (C3.4)* aspect, a certain degree of under specification and a lack of examples concerning choreographies were also noted. As a result, effective use of the language becomes more difficult (e.g., the *ChoreographyLoopType*[3]). The introduction of guidance and examples in the standard should be improved. This may encourage the production of effective models.

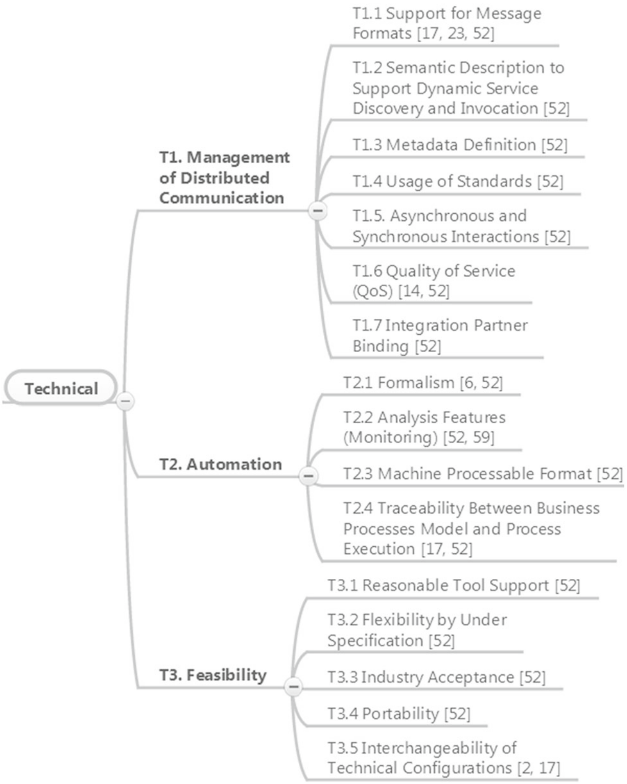
Technical
T1. Management of Distributed Communication
T1.1 Support for Message Formats [17, 23, 52]
T1.2 Semantic Description to Support Dynamic Service Discovery and Invocation [52]
T1.3 Metadata Definition [52]
T1.4 Usage of Standards [52]
T1.5. Asynchronous and Synchronous Interactions [52]
T1.6 Quality of Service (QoS) [14, 52]
T1.7 Integration Partner Binding [52]
T2. Automation
T2.1 Formalism [6, 52]
T2.2 Analysis Features (Monitoring) [52, 59]
T2.3 Machine Processable Format [52]
T2.4 Traceability Between Business Processes Model and Process Execution [17, 52]
T3. Feasibility
T3.1 Reasonable Tool Support [52]
T3.2 Flexibility by Under Specification [52]
T3.3 Industry Acceptance [52]
T3.4 Portability [52]
T3.5 Interchangeability of Technical Configurations [2, 17]

Fig. 16 Overview of technical requirements

6 Technical requirements evaluation

In the technical evaluation, three main axes are analyzed, namely *Management of Distributed Communication (T1)*, *Automation (T2)* and *Feasibility (T3)*. Figure 16 gives an overview of the evaluated requirements.This section evaluates the capacity of the language to go toward an implementation tool. Several meta-model constructs not captured in the graphical notation need to be illustrated to minimize potential inconsistencies between diagrams and textual models.

6.1 T1. Management of Distributed Communication

The management of distributed communication in the context of BPMN 2.0 is evaluated here. Table 12 summarizes the different sub-requirements and provides the ratings. The various points of the evaluation are then analyzed.

T1.1 Support for Message Formats. Message formats are critical to achieve interoperability between participants as

[3] http://www.omg.org/issues/bpmn2-rtf.open.html#Issue16554.

Table 12 Management of distributed communication requirements

Refinement	Description	Graphical	XML Schema	Suggestion
T1.1 Support for Message Formats	The definition of message formats for business documents and sensibly dealing with the meaning of their content are vital to achieve interoperability between partners	0	2	S-T1.1
T1.2 Semantic Description to Support Dynamic Service Discovery and Invocation	Dynamic discovery and invocation of unknown services according to semantic description that captures the functionality needed and QoS parameters should be supported	0	1	S-T1.2
T1.3 Meta-data Definition	Detailed information for implementation must be supported in addition to the proper code	2	2	
T1.4 Usage of Standards	Standards define common formats, protocols, concepts and terminology. This facilitates communication among participants and reduces the maintenance cost for information systems	2	2	
T1.5. Asynchronous and Synchronous Interactions	Support for both asynchronous and synchronous interactions should be attained as both scenarios can be found in the choreography domain	1	2	S-T1.5
T1.6 Quality of Service	QoS is an important axis to support "all non-functional aspects of a service which may be used by clients to judge service quality." [22]	0	0	S-T1.6
T1.7 Integration Partner Binding	Concepts for binding a contract to a concrete partner may be helpful	0	2	S-T1.7
		5/14 (35.71 %)	11/14 (78.57 %)	

T1. Management of Distributed Communication: A choreography language should provide a means of implementing a distributed communication within a system. A choreography language must facilitate associations between partners

shown by industrial initiatives such as RosettaNet. In BPMN 2.0, a message may have optionally an *itemRef* attribute (not visible in the diagrams) implemented in the XML level by an *itemDefinition*, which contains a reference to a concrete data schema [32]. It could be argued that this requirement should be much more clearly specified and explained in the standard. Explicitly attaching a format construct to the message term is recommended. Different message formats could be distinguished graphically [S-T1.1]. Consequently, this requirement is supported in the XML Schema but not graphically.

T1.2 Semantic Description to Support Dynamic Service Discovery and Invocation. Services could be identified in design time, but they should also be identified in a more dynamic way, in run time. Dynamic discovery and invocation of unknown services according to semantic description that captures the functionality needed is necessary in order to describe highly dynamic environments. The semantic description to support this requirement is outside the scope of the standard. A clearer separation between the *Participant*, *PartnerRole*, *PartnerEntity* and *Interface* constructs might help distinguish the desired functionalities for an expected type of service and the potential providers [S-T1.2]. In the graphical notation, the four constructs are hidden under the *Participant* element and this may be confusing. Therefore, the XML Schema partially supports this requirement, and there is no graphical support.

T1.3 Meta-data Definition. In general, meta-data may be intended for human users, but requirement engineering sources such as KAOS methodology also suggest a more technical use [31]. Meta-data definition may serve various purposes ranging from detailed information for implementation to the dynamic discovery of processes/services. The latter application has already been treated in requirement T1.2. Here, the focus is placed on the extra information given to both users or modelers and tools. Good support for meta-data definition is found in BPMN 2.0 in both graphical and XML specification. *Text Annotations* as well as *Expressions* might help complete the model semantics. It would be even be possible to define *FormalExpressions* with a concrete data structure. BPMN 2.0 also provides a set of extension elements, which allows BPMN adopters to attach additional attributes and elements to standard BPMN elements.

T1.4 Usage of Standards. The standard uses other different standards to define data types, expressions and service operations such as XML Schema, XPath and WSDL, respectively [45]. This characteristic is a favorable point for BPMN 2.0 for promoting interoperability between partners.

Table 13 Automation Requirements

Refinement	Description	Graphical	XML Schema	Suggestion
T2.1 Formalism	Precise/formal definition or mapping into to a well-established formalism	1	1	S-T2.1
T2.2 Analysis Features (Monitoring)	Monitoring mechanisms can be introduced in the language to facilitate the analysis of the choreography	0	1	S-T2.2
T2.3 Machine-processable Format	Models should be processable by machines	2	2	
T2.4 Traceability Between Business Process Model and Process Execution	Assessing the correctness of process executions requires comparing the result of process runs with process definitions	1	2	S-T2.4, S-T2.4′
		4/8 (50 %)	6/8 (75 %)	

T2. Automation: Automation is essential to minimize human error and to automate time-consuming tasks

T1.5 Asynchronous and Synchronous Interactions. Both asynchronous and synchronous interactions could be modeled in BPMN 2.0. These different ways of communication should be explicitly captured by the choreography language. Fig. 1 shows the distinction between both communication styles. On the one hand, when the *Manufacturer* receives the request order by the *Customer*, the latter is not blocked while waiting for the answer (asynchronous communication). Another interaction is performed later to indicate that the request was correctly treated (confirmed) or not (rejected). On the other hand, the interactions between the *Manufacturer* and both the *Supplier* and the *Bidder* are performed synchronously. In the collaboration diagram, both request and answer messages are linked to the same activity. Silver [54] recommends representing synchronous service calls as *Service Tasks* and asynchronous ones as separate *Send* and *Receive* tasks (or their *Message* event equivalents). Using this convention, synchronous interactions are represented as request–response choreography tasks (e.g., *Procure Part* and *Auction Part*) and asynchronous service calls as separated choreography tasks in the choreography diagram. Note that the standard gives no information on this point, as the difference between these two ways of communication is not specified in this way.

A comprehension problem might arise when synchronous communication is defined in BPMN 2.0 where other interactions are produced between the request and the answer of an operation. Imagine that the *Customer* and the *Manufacturer* on Figure 1 now communicate synchronously. No explicit relationship between the two interactions that might be part of the same synchronous operation is explicitly represented. The latter point is discussed in more detail in Req. *T2.4 Traceability Between Business Process Model and Process Execution*. It may be concluded that this requirement partially supported graphically and fully supported in the XML Schema. A method of explicitly representing synchronous and asynchronous communications should be introduced, especially in choreography diagrams [S-T1.5].

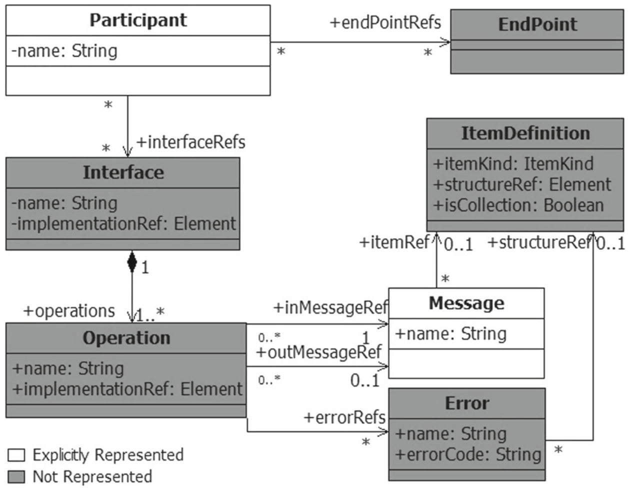

Fig. 17 Service Meta-model in BPMN 2.0

T1.6 Quality of Service (QoS). QoS is a major axis to support "all non-functional aspects of a service which may be used by clients to judge service quality" [22]. This important requirement to be considered in service architectures is outside scope of the standard. But QoS may be a critical aspect in a multi-party collaboration such as choreographies. The recommended approach to QoS (that may be defined in a separated standard to avoid introducing more complexity in the meta-model) would be to take into account proposals such as the Q-WSDL [14] where authors depict how to extend the WSDL meta-model in order to capture non-functional requirements (QoS). A QoS construct may therefore be defined as an extension point, the same as the *Monitoring* or the *Auditing* constructs, that are mentioned in the standard [S-T1.6] (Table 13).

T1.7 Integration Partner Binding. Concepts for binding a contract to a concrete partner may be helpful. BPMN 2.0 proposes *Interfaces* to define the contract of a specific participant (service). The standard also provides a mean of attaching a *Participant* to one or more *Endpoints*. As can be observed in the service meta-model of Fig. 17, both *Interface* and *End-*

point elements are not represented graphically (dark gray). A graphical construct for service interfaces linked to each participant might improve support for this requirement and clarify the relation between participants and WSDL contracts [S-T1.7]. Again, the requirement is fully supported in the XML Schema but not supported graphically.

6.2 T2. Automation

T2.1 Formalism. The main criterion for automation is *Formalism*. Both syntax and semantics must be formalized [6]. Accordingly, a choreography diagram should have precise semantics, which can be specified, for example by mapping to a formal notation (e.g., colored petri nets or pi-calculus) [S-T2.1]. Although substantial efforts have been made to clearly define the execution semantics of process diagrams, the semantics of choreography diagrams are not well defined. This hinders the local enforceability rules from choreography diagrams to collaboration diagrams and may lead to ambiguities when combining both diagrams. Local enforceability means that the sequencing in the global choreography model has to be coherent with the sequencing of the related message exchanges in individual partner processes [7]. This mapping is not totally defined in the standard; therefore, this requirement is only partially supported at both levels.

T2.2 Analysis Features (Monitoring). The work carried out by Wetzstein et al. [59] highlighted the usefulness of monitoring choreographies. Thus, a choreography language should be defined so that choreography monitoring is made possible. Analysis features such as monitoring are considered in the standard meta-model but not developed in detail. The standard only indicates an extension start point in the meta-model. This point may be developed in more detail in order to be able to define, for example what parts of the choreography need to be monitored and how [S-T2.2]. Hence, this requirement is partially supported in the XML Schema, but there is no possible way to support this graphically.

T2.3 Machine-processable Format. One of the good qualities in the standard that helps automation is the use of XML to serialize diagrams. This makes the models readable by machines. The diagrams' XML serialization represented in BPMN Diagram Interchange format (BPMN DI) [45] helps define the concrete graphical syntax of the language. Note that distinction is made between XML serialization of the graphical diagrams (i.e., forms, sizes, position, etc.) and the XML Schema, which implements the meta-model concepts. Therefore, this requirement is fully supported.

T2.4 Traceability Between Business Process Model and Process Execution. Adequate alignment between choreographies and process execution is necessary in order to bridge the "Business-IT gap" and to avoid inconsistencies between process modeling and execution. This can be achieved by aligning (though not necessarily mapping) the choreography language to an executable process definition language such as the Business Process Execution Language (BPEL) [43] (cf. the approach taken in BPEL4Chor [17]) or by aligning choreographies with executable BPMN models.

The possibility of creating a skeleton of the participants' process starting from the collaboration diagram facilitates this integration in BPM 2.0. However, shown in Fig. 18, the link between choreography diagrams and technical services in the graphical perspective can be very confusing (see the service meta-model in Fig. 17). Imagine that a manufacturer interface has to be defined related to the *Manufacturer* participant. An obvious operation that may be declared in the interface might be *Order Goods* or a similar operation. In the example, a synchronous operation is considered. This operation might have the *Order* message as input, a *Confirmation* as output message and a *Rejection* as error message. The diagram illustrates two important problems: (1) First, messages that potentially take part of the same service operation cannot be grouped together and (2) it is not possible to distinguish between messages that may be output messages of a service operation and messages that may be input in cases (like in the example) where interactions are produced between the request and the response of an operation.

It could be argued that each initiating message (the white message) always corresponds to an input message in a service operation. But this constraint is too strong and restrictive and may not be adapted to scenarios such as the one in Fig. 18 synchronous communication is considered between the *Customer* and the *Manufacturer*. For instance, defining a *Rejection* message as an input message for an operation implemented by the *Customer* will make no sense (it should be represented as a response message and not as initiator message). However, in an asynchronous communication, this model will fit well (the confirmation may be a message within a callback operation implemented by the *Customer* service). Here again, some technical aspects should be graphically represented to be more precise when capturing choreographies. In the first place, messages that take part of the same service operation should be graphically linked [S-T2.4]. Secondly, an abstract view of the participants' interfaces should be represented together with operations attached to the participant construct in order to increase technical precision [S-T2.4'].

6.3 T3. Feasibility

In order to analyze *Feasibility*, the focus is placed on points such as flexibility, adaptability and portability. BPMN is considered to be an implementation independent language [15]. In addition, it provides an easy connection to WSDL allowing a simple transition to web service technology in the XML implementation (Table 14).

T3.1 Reasonable Tool Support. There is still not great tool support for the choreography subset of BPMN 2.0.

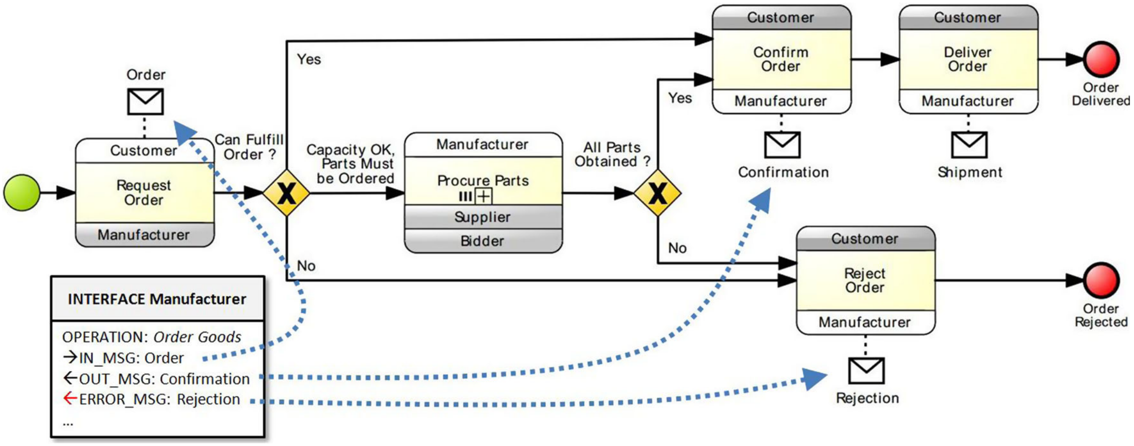

Fig. 18 Potential traceability problem between technical services and choreography diagram

Table 14 Feasibility Requirements

Refinement	Description	Graphical	XML Schema	Suggestion
T3.1 Reasonable Tool Support	Reasonable tool support is a cross-cutting concern throughout all phases of any project. The distributed computing setting and the management of complex associations require special support	1	1	S-T3.1
T3.2 Flexibility by Underspecification	This requirement calls for the provision of the means for putting into production process models that do not describe the control flow in full detail	0	2	S-T3.2
T3.3 Industry Acceptance	Formats, protocols and technologies should be aligned with the needs of the integration partners and the integration partners should not have to adapt their needs to the available technology	1	1	
T3.4 Portability	Model interchange support. Models might be opened by different tools	2	2	
T3.5 Interchangeability of Technical Configurations	The language might support changes in the interfaces which may not affect the global definition of the choreography	0	1	S-T3.5
		4/10 (40 %)	7/10 (70 %)	

T3. Feasibility: This requirement is important for agreement needs. Special attention must be taken when considering communication between heterogeneous people and heterogeneous computing resources

Tools are for the moment focused on model and execute processes diagrams. However, there is an increasing interest in choreographies, and tools are starting to implement this subset of the meta-model. Proposals such as SAVARA[4] or CHOREOS[5] are some examples of projects that have adopted choreography as a first-class citizen. BPMN 2.0 is the chosen language to capture choreographies in both cases. As indicated in Sect. 2.1, BPMN 2.0 collaboration diagrams and choreography diagrams are complementary and can be represented together. Nevertheless, tests were conducted on some BPMN 2.0 tools that implement the choreography subset such as Signavio,[6] Eclipse BPMN 2.0 Modeler,[7] EasierCos[8] and Cameo Business Modeler[9] and it was noted that none of them integrated the choreography and the collaboration diagrams together. A more precise mapping between the choreography diagram and the collaboration diagram is required (now, only indicative mapping examples are provided). Efforts should be made to clarify the choreography diagram semantics so that tool adoption is made easier [S-T3.1]. Currently, this requirement is rated as partially supported in both levels.

[4] http://www.jboss.org/savara.

[5] http://www.choreos.eu/bin/Main/.

[6] http://www.signavio.com/.

[7] http://www.eclipse.org/bpmn2-modeler/.

[8] http://research.linagora.com/display/easiercos.

[9] http://www.nomagic.com/products/cameo-business-modeler.html.

Table 15 Technical Requirements Recap

Requirement	Graphical	XML Schema
T1. Management of Distributed Communication	5/14 (35.71 %)	11/14 (78.57 %)
T2. Automation	4/8 (50 %)	6/8 (75 %)
T3. Feasibility	4/10 (40 %)	7/10 (70 %)
Avg. Percentage	41.9 %	74.52 %

T3.2 Flexibility by Underspecification. This is required to provide the means for putting process models into production without describing the control flow in full detail. This is needed for scenarios that require extensive human involvement or that are too variable to justify fully specifying each detail in a process model. Constructs such as *isClosed* and *isImmediate* specify whether *Activities* or *Choreography Activities* not in the model containing the *Sequence Flow* can occur between the elements connected by the *Sequence Flow*. This increases flexibility and adaptability to the choreography models in BPMN 2.0. However, in the graphical view, there is no way of distinguishing these "flexible" models. Markers should be introduced that advertise graphically the use of this property [S-T3.2]. Therefore, BPMN 2.0 supports this facet in the XML Schema but not graphically.

T3.3 Industry Acceptance. Good industry acceptance is one of the main strengths of BPMN 2.0. Indeed, industry is more focused on the implementation of executable processes than on choreographies so it is too early to give conclusions on choreography conformance support [45]. Better tool support for the language may imply better industry acceptance for choreographies. The acceptance might not be hard in terms of costs if there are already BPMN tools in use (previous versions or other sub-parts of BPMN 2.0). However, it is difficult to predict whether BPMN 2.0 development will help choreography diagrams to be accepted. At least, industry initiatives such as RosettaNet show the advantages to be gained from using choreographies. Schönberger [51] pointed out the need for visual choreography models proposing a graphical notation inspired from BPMN 2.0 in order to support the RosettaNet methodology for creating choreographies. For the time being, this requirement is considered to be only partially supported. No suggestions are provided on this point.

T3.4 Portability. A choreography language should define an interchange format covering both the language itself as well as graphical data attached to the model elements. Accordingly, BPMN 2.0 standard specifies the meta-model and schema for BPMN 2.0 Diagram Interchange (BPMN DI). So portability of diagrams between tools is also supported. Therefore, this requirement is fully supported in both levels.

T3.5 Interchangeability of Technical Configurations. To enable an easy exchange of choreography implementations (such as concrete WSDL *interfaces* and WSDL operations) the technical configurations of a choreography may be interchangeable and should be separated from the choreography model. In BPMN collaborations, participants can be associated with interfaces. These abstract interfaces may contain references that point to a corresponding technical implementation, such as WSDL *interfaces*. This implies updating the reference if the *interface* name changes. Hence, BPMN 2.0 is not as flexible as might be desired. BPEL4Chor is an example of a choreography language that gives support for this requirement [17]. Possible mapping of this language may resolve this problem as proposed in [32] [S-T3.5].

6.4 Summary: technical requirements

Requirements in this axis are found to be much better supported in the XML level which is faily logical (41.9 vs. 74.52 %). However, it would be desirable to have a refined graphical notation where many of these requirements that are currently only implemented in a textual notation could be expressed. Here again, the same necessity arises as when the domain and comprehensibility axes were analyzed. The graphical notation is still not sufficiently expressive. Table 15 gives an overview of the requirement support in this axis and Table 16 summarizes the suggestions given for each requirement.

6.4.1 Strong points

It is clear that the OMG has made great efforts in the technical aspects of this new version. This may help to adopt choreographies into implementations and facilitate interoperability. Regarding the **Management of Distributed Communication (T1)**), the *Use of Standards (T1.4)* is a critical point, which is well supported. The evaluation also highlighted the good support for *Meta-data Definition*. In order to achieve **Automation (T2)**, good support for *Machine-processable Format (T2.3)*, with a XML diagram serialization is a good starting point in BPMN 2.0. In the **Feasibility (T3)** area, the introduction of interchange formats for abstract syntax in both XMI and XSD is a great improvement to achieve *Portability (T3.4)* between tools. This last point will surely help to improve *Industry Acceptance (T3.3)* of the choreography subset of BPMN 2.0. It is to be hoped that BPMN 2.0 implementations will take into account all these improvements presented in the standard.

6.4.2 Weak points

T1. Management of Dist. Communication. Major shortcomings have been detected in the requirements to support the management of distributed communication. First, *Mes-*

Table 16 Summary of Suggestions for the Technical Appropriateness Requirements

Requirement	Suggestion	Suggestion description
T1. Management of Distributed Communication		
T1.1 Support for Message Formats	S-T1.1	A format construct should be attached explicitly to the message term. Different message formats could even be distinguished graphically
T1.2 Semantic Description to Support Dynamic Service Discovery and Invocation	S-T1.2	A clearer separation between the *Participant*, *PartnerRole*, *PartnerEntity* and *Interface* constructs might help distinguish the desired functionalities for an expected type of service and the potential providers. In the graphical notation, these four constructs are hidden under the *Participant* element and may be confusing
T1.3 Meta-data Definition		*Good Support—No Suggestion*
T1.4 Usage of Standards		*Good Support—No Suggestion*
T1.5. Asynchronous and Synchronous Interactions	S-T1.5	It is recommended to provide a means of explicitly representing synchronous and asynchronous communications, especially in choreography diagrams
T1.6 Quality of Service	S-T1.6	A QoS construct may be defined as an extension point, the same as the *Monitoring* or the *Auditing* constructs that provide a hook for related properties
T1.7 Integration Partner Binding	S-T1.7	A graphical construct for service interfaces attached to each participant might improve support for this requirement and clarify the relation between participants and WSDL contracts
T2. Automation		
T2.1 Formalism	S-T2.1	A choreography diagram should have precise semantics, which can be specified, for example, by mapping to a formal notation (e.g., colored petri nets or pi-calculus)
T2.2 Analysis Features (Monitoring)	S-T2.2	The *Monitor* construct may be developed in more detail in order to define, for example, the parts of the choreography to be monitored and how to do this
T2.3 Machine-processable Format		*Good Support—No Suggestion*
T2.4 Traceability Between Business Process Model and Process Execution	S-T2.4, S-T2.4′	Messages that are part of the same service operation should, in the first instance, be graphically linked. In a second stage, an abstract view of the participants' interfaces and their operations attached to the participant construct should be represented in order to increase technical precision
T3. Feasibility		
T3.1 Reasonable Tool Support	S-T3.1	Efforts should be made to clarify the choreography diagram semantics so that tool adoption is easier. More precise mapping between the choreography diagram and the collaboration diagram is required (currently based on few examples)
T3.2 Flexibility by Underspecification	S-T3.2	There is no way in the graphical view to differentiate between "flexible" models. A marker should be introduced graphically advertises the use of this property
T3.3 Industry Acceptance		*No Suggestion*
T3.4 Portability		*Good Support—No Suggestion*
T3.5 Interchangeability of Technical Configurations	S-T3.5	BPEL4Chor is an example of a choreography language that gives support for this requirement [17]. A possible mapping to this language may resolve this problem as proposed in [32]

sage Formats (T1.1) should be adopted as first-class citizens because they are critical feature for interoperability between partners. Explicitly attaching a format construct to the message term may clarify the support for this requirement. *Semantic Description to Support Dynamic Service Discovery and Invocation (T1.2)* also has to be improved. The separation between the constructs *Participant*, *PartnerRole*, *PartnerEntity* and *Interface* should be clarified. All of them could be graphically represented instead of relying on a single *Participant* representation.

T2. Automation. In order to achieve automation, a mapping to a formal notation such as petri nets is desirable to achieve formal semantics and eliminate the current ambiguities of the choreography diagrams. *Analysis Features (T2.2)* should be developed to facilitate user analyses and improve choreography implementation. Good support for *Traceability between Process Models and Process Execution (T2.4)* is also critical. A more expressive notation capturing lower level requirements may be a good way of obtaining better support for this requirement. For a better support of *Asynchronous and Synchronous Interactions (T1.5)*, BPMN 2.0 should enable both communication forms to be represented without ambiguities especially in choreography diagrams. Currently, there is no way of distinguishing both communication methods. *Quality of Service (QoS) (T1.6)* is currently outside of the scope of the standard. A construct may be defined as an extension point provide to better support for this requirement. Finally, a graphical construct should be defined to represent participant's interfaces graphically so that *Integration Partner Binding (T1.7)* could be better supported.

T3. Feasibility. Other issues were also noted in the feasibility area. There is still no extensive *Tool Support (T3.1)* for the BPMN 2.0 choreography subset. The standard needs to clarify the choreography diagram semantics so that tool adoption is made easier. A more precise mapping between the choreography diagram and the collaboration diagram is required (currently based on few examples). To obtain a better support for the *Flexibility by Underspecification (T3.2)* requirement, markers should be graphically represented when under-specified choreographies are modeled. *Industry Acceptance (T3.3)* may be facilitated by improving tool support. But first of all BPMN 2.0 should eliminate semantics ambiguity. The mapping to BPEL4Chor could help improve the *Interchangeability of Technical Configurations (T3.5)* as pointed out by the authors in [32].

7 Summary and conclusion

The appropriateness of the latest version of BPMN (version 2.0) to capture service choreography has been evaluated. In its previous version, choreographies could only be represented using the *collaboration diagram*. In this new version, the collaboration diagram can be complemented by the *choreography diagram*, which captures interactions in a truly overall manner, and the *conversation diagram*, which provides an abstract way of representing message exchanges. BPMN 2.0 allows a combination of the three diagrams. Therefore, BPMN 2.0 provides a powerful graphical means of supporting choreography as the language is well adapted to the choreography domain. BPMN 2.0 also has widespread industrial support, and it interacts with standards such as XPath, XML Schema or WSDL. This promotes interoperability between participants. The definition of interchange formats to support portability is also a remarkable point.

Table 17 Overview of the Choreography Requirements' Support in the Three Axes in BPMN 2.0

Axis	Graphical (%)	XML Schema (%)
Domain Appropriateness Requirements (D)	58.13	80
Comprehensibility Appropriateness Requirements (C)	64.58	56.25
Technical Appropriateness Requirements (T)	41.9	74.52

Nevertheless, the graphical notation of BPMN 2.0 is still not sufficiently developed to express choreographies in a precise way. In the evaluation, significant difference in support difference in support between the graphical notation and the XML models was found. This gap may lead to errors and misunderstandings. The gap between graphical representation and the XML Schemas, which support the meta-model, is highlighted in Table 17 where an overview of the evaluation is shown. In the domain axis, more than 40% of the requirements are not reached in the graphical level. Even worse, in the technical axis, almost 60 % of the requirements are not supported. On the other hand, requirements are generally well supported at the XML Schema level. However, comprehensibility issues in this level have been highlighted. Table 17 clearly states that the domain and the technical requirements are well supported in the XML Schema, but there are major deficiencies in the graphical representation. In conclusion, it was found that BPMN 2.0 has great potential for capturing choreographies but is not yet a complete service choreography language as there are still substantial limitations, especially in the graphical part. These limitations have been discussed in the previous sections.

BPMN was not intended to be used to specify service choreographies but internal processes (orchestrations) [54]. Therefore, it is difficult for this language to be sufficiently expressive to properly model choreography with the same level of appropriateness as domain-specific language. However, it has to be highlighted that BPMN 2.0 gives the possibility of representing a business process with different perspectives such as the individual process or the choreography view. Only choreography support is considered in the evaluation, but the latter point is a major advantage of BPMN 2.0 compared to other choreography-specific languages.

Some of the *main shortcomings* detected in the evaluation are:

- The gap between the graphical notation and the XML Schema, implementing the BPMN 2.0 meta-model. The graphical notation is still not sufficiently expressive.
- The meta-model quality, which should be a tool to understand the choreography concept. Instead, it is presented in a very technical way.
- The lack of traceability between choreography models and technical services that may lead to errors when trying to specify choreographies in a more technical way.

– The semantics of the choreography diagrams, which are not well defined.

To resolve most of these deficiencies, it would be worth proposing a more refined graphical notation, where certain technical requirements could be introduced. For instance, Schönberger et al. [51] adapted the BPMN 2.0 choreography diagrams to graphically represent choreographies in a more precise manner than BPMN 2.0. Ideally, the choreography meta-model should be stratified in three different abstraction layers as discussed in [10]. Consequently, the graphical notation could be adapted to this three-leveled choreography meta-model. For example, there could be a more conceptual level where core elements of the chorography are represented (potentially, the current graphical notation). The conceptual level would be refined in an analysis level, where several key technical elements would also be represented graphically. The lower level could include the current XML Schema models, which maps well with executable BPMN. The main idea would be to introduce an intermediate graphical layer that bridges the gap between the current graphical models and the textual models. Then, traceability between "human-oriented" models and "machine-oriented" models would be more easily achieved. This three-level representation could be complemented by a well-defined structural and behavioral views. A stepwise refinement [7] would be an effective way of maintaining the consistency between levels and between views.

Empirical studies with service and business specialists have been considered for completing and validating our requirements proposal. It is also necessary to look into defining more measurable scales to automate the evaluation. These measures will also be useful in order to compare BPMN 2.0 with other choreography proposals.

It is worth emphasizing underscore that the main goal of this paper is to evaluate the adequacy of BPMN 2.0 to support service choreographies.

7.1 Limits of the evaluation

In this evaluation, the aim was to be objective as much as possible to allow other researchers to reproduce the same evaluation while using the same requirements. The language meta-model presented in the standard is systematically relied upon to analyze the support for the different requirements. However, this is not always sufficient, notably when looking at the comprehension axis where objectivity is much more difficult to reach. This assessment relied on empirical studies performed in previous work [10], which was performed to validate some of the proposals presented in this paper, and also in other evaluations performed in the field [27,40,48]. The main purpose of the numeric rating score given in the evaluation is to identify limitations and areas for improvement in the language.

A point of discussion might be the fact that not all B2Bi requirements are considered. Only those applicable to service choreographies. Discussions were held with the author to determine the B2Bi requirements that might be considered to evaluate service choreographies. Several requirements were rejected two main reasons, namely: (1) because they were not considered appropriate for evaluating service choreography languages. For example, *Intelligible feedback of analysis* (Req. 2 [52]) is specific to B2Bi and not to service choreography and (2) because they were focused on an architecture or tool and not on the language itself. For instance, *Repository Functionality* or *Documentation* (Req. 3 and Req. 31 in [52]) are requirements that are not applicable to the language but focused on support mechanisms and the architecture.

One major issue in BPMN 2.0 choreography diagrams is that their semantics are not well defined. The standard only provides an indicative idea of the semantics through local enforceability of different BPMN choreography constructs and modeling situations. So, even if support could be claimed for most of the service interaction patterns [5], the underspecified semantics of the choreography model make the analysis difficult.

Moody [40] states that some interactions among the notation principles are observed. Interactions may have a positive or a negative effect. For example, semiotic clarity (one-to-one correspondence between symbols and concepts) may have a negative effect in graphical economy within BPMN 2.0 (currently, not all the concepts are represented graphically). In the same way, it may be argued that the addition of proposed constructs or extensions may decrease comprehensibility criteria. Nevertheless, the fact of defining several abstraction layers should minimize this potential issue by promoting stepwise refinement.

Unfortunately, only a small part of all the points covered in this BPMN 2.0 evaluation are included in the empirical studies. The evaluation may be biased because it is mostly expert centered. Final users of the language are not systematically asked to evaluate the appropriateness for a given requirement and as its rating. The evaluation relies mainly on a literature review and the analysis of the BPMN 2.0 standard. Empirical studies are very time-consuming and difficult to implement. Several points were assessed concerning the comprehensibility and the domain axes, which were well suited to focus group techniques and sufficiently intuitive to be presented in an online questionnaire. However, the ideal scenario would be to integrate all the points of the evaluation and the proposals in the user-center methodology. The suggestions described in the paper have not yet been implemented. They are still in a theoretical state. A graphical editor inspired by the Eclipse BPMN 2.0 Modeler[10] is currently in progress.

It is recommended to review and improve the choreography requirements further so that they can be used as a source of inspiration and reference to evaluate choreography languages, as well as to achieve a better understanding of this currently emerging concept. However, it is to be hoped that this publication will help modelers understand the great potential of BPMN 2.0 for choreographies.

Acknowledgments We should like to thank Andreas Schönberger for valuable discussions on choreography integration requirements. We are also grateful to Marlon Dumas and Alistar Barros for useful feedback in the analysis of BPMN 2.0. This work was supported by the ANR MOANO project. All URLs were checked on November 15, 2013.

References

1. Ambler, S.: The Elements of UML 2.0 Style. Cambridge University Press, Cambridge (2005)
2. Barker, A., Walton, C., Robertson, D.: Choreographing web services. IEEE Trans. Serv. Comput. **2**(2), 152–166 (2009)
3. Barros, A., Decker, G., Dumas, M.: Multi-staged and multi-viewpoint service choreography modelling. In: Proceedings Workshop on Software Engineering Methods for Service Oriented Architecture (SEMSOA), CEUR Workshop Proceedings, vol. 244 (2007)
4. Barros, A., Decker, G., Dumas, M., Weber, F.: Correlation patterns in service-oriented architectures. In: Fundamental Approaches to Software Engineering, pp. 245–259 (2007)
5. Barros, A., Dumas, M., Hofstede, A.: Service interaction patterns. In: Aalst W., Benatallah, B., Casati, F., Curbera, F. (eds.) Business Process Management, 3rd InternationalConference, BPM 2005,. Lecture Notes in COmputer Science, vol. 3649, pp. 302–318. Springer, Heidelberg (2005). doi:10.1007/1153839420
6. Barros, A., Dumas, M., Oaks, P.: A critical overview of the web services choreography description language. BPTrends Newsl. **3** 1–24 (2005). http://www.pbtrends.com
7. Barros, A., Hettel, T., Flender, C.: Process choreography modeling. In: Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1, International Handbooks on Information Systems. pp. 257–277. Springer, Heidelberg (2010). doi:10.1007/978-3-642-00416-212
8. Biron, P., Malhotra, A., Consortium, W.W.W., et al.: Xml Schema Part 2: Datatypes. World Wide Web Consortium Recommendation REC-xmlschema-2-20041028 (2004)
9. Cook, T., Reichardt, C.: UML, vol. 1. Sage publications, London (1979)
10. Cortes-Cornax, M.: Service choreographies through a graphical notation based on abstraction layers and viewpoints. In: 2011 Fifth International Conference on Research Challenges in Information Science (RCIS), IEEE, pp. 1–12 (2011)
11. Cortes-Cornax, M., Dupuy-Chessa, S., Rieu, D.: Bridging the gap between business processes and service composition through service choreographies. In: Engineering Methods in the Service-Oriented Context, pp. 190–203 (2011)
12. Cortes-Cornax, M., Dupuy-Chessa, S., Rieu, D.: Choreographies in BPMN 2.0: new challenges and open questions. In: Schönberger, A., Kopp, O., Lohmann, N. (eds.) Proceedings of the 4th Central-European Workshop on Services and their Composition, vol. 847, pp. 50–57. CEUR (2012). http://ceur-ws.org/Vol-847/paper7.pdf
13. Cortes-Cornax, M., Dupuy-Chessa, S., Rieu, D., Dumas, M.: Evaluating choreographies in bpmn 2.0 using an extended quality framework. In: Dijkman, R., Hofstetter, J., Koehler, J. (eds.) Business Process Model and Notation - Third International Workshop, BPMN 2011, Lucerne, Switzerland. Lecture Notes in Business Information Processing, vol. 95, pp. 103–117. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25160-38
14. D'Ambrogio, A., Bocciarelli, P.: A model-driven approach to describe and predict the performance of composite services. In: Cortellessa, V., Uchitel, S., Yankelevich, D. (eds.) WOOSP 2007 Proceedings of the 6th International Workshop on Software and Performance, pp. 78 -89. ACM (2007)
15. Decker, G., Kopp, O., Barros, A.: An introduction to service choreographies. Inf. Technol. **50**(2), 122–127 (2008)
16. Decker, G., Kopp, O., Leymann, F., Pfitzner, K., Weske, M.: Modeling service choreographies using bpmn and bpel4chor. In: Advanced Information Systems Engineering, pp. 79–93. Springer, Berlin (2008)
17. Decker, G., Kopp, O., Leymann, F., Weske, M.: Interacting services: from specification to execution. Data & Knowl. Eng. **68**(10), 946–972 (2009)
18. Decker, G., Puhlmann, F.: Extending BPMN for modeling complex choreographies. In: Proceedings of the 2007 OTM Confederated international conference on On the Move to Meaningful Internet Systems: CoopIS, DOA, ODBASE, GADA, and IS-Volume Part I, pp. 24–40. Springer, Berlin (2007)
19. Decker, G., Weske, M.: Interaction-centric modeling of process choreographies. Inf. Syst. **36**, 292–312 (2011). doi:10.1016/j.is.2010.06.005
20. Dijkman, R., Dumas, M.: Service-oriented design: a multi-viewpoint approach. Int. J. Coop. Inf. Syst. **13**, 337–368 (2004)
21. DIS, I.: 9241–210: 2009. Ergonomics of human system interaction-part 210: human-centred design for interactive systems. International Organization for Standardization (ISO). Switzerland (2010)
22. Dobson, G., Lock, R., Sommerville, I.: QoSont: a QoS ontology for service-centric systems. In: 31st EUROMICRO Conference on Software Engineering and Advanced Applications, 2005, pp. 80–87. IEEE (2005)
23. Dubray, J., Amand, S., Martin, M.: ebxml Business process specification schema technical specification v2. 0.4. Committee specification, UN/CEFACT (2006)
24. Dupuy-Chessa, S.: Quality in ubiquitous information system design. In: Third International Conference on Research Challenges in Information Science, 2009, RCIS 2009. pp. 343–352. IEEE (2009)
25. Evitts, P.: A UML Pattern Language. New Riders Publishing, Thousand Oaks (2000)
26. Genero, M., Piattini, M., Calero, C.: A survey of metrics for uml class diagrams. J. Object Technol. **4**(9), 59–92 (2005)
27. Genon, N., Heymans, P., Amyot, D.: Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. Software Language, Engineering, pp. 377–396 (2011)
28. Hafner, M., Breu, R.: Realizing Model Driven Security for Inter-organizational Workflows with WS-CDL and UML 2.0. Model Driven Engineering Languages and Systems, pp. 39–53 (2005)
29. Harel, D., Thiagarajan, P.: Message Sequence Charts, UML for Real. Springer, Berlin (2004)
30. ISO/IEC: 9126–1:2001: Software Engineering—Product Quality. International Organization for Standardization (2001)
31. Koliadis, G., Ghose, A.: Relating business process models to goal-oriented requirements models in KAOS. In: Advances in Knowledge Acquisition and Management, pp. 25–39 (2006)
32. Kopp, O., Leymann, F., Wagner, S.: Modeling Choreographies: BPMN 2.0 Versus BPEL-based approaches. In: Nüttgens, M.,

[10] http://www.eclipse.org/bpmn2-modeler/.

Oliver, T., Weber, B. (eds.) Enterprise Modelling and Information Systems Architectures: Proceedings of the 4th International Workshop on Enterprise Modelling and Information Systems Architectures, EMISA 2011, pp. 225–230. GI LNI (2011)
33. Krogstie, J.: Evaluating UML using a generic quality framework. In: UML and the Unified Process (2003)
34. Krogstie, J., de Flon Arnesen, S.: Assessing enterprise modeling languages using a generic quality framework. In: Krogstie, J., Halpin, T.A., Siau, K. (eds.) Information modeling methods and methodologies (1537–9299), pp. 63–79. Idea Group Publishing (2005)
35. Krogstie, J., Sindre, G., Jørgensen, H.: Process models representing knowledge for action: a revised quality framework. Eur. J. Inf. Syst. **15**(1), 91–102 (2006)
36. Le Pallec, X., Dupuy-Chessa, S., Dupuy, S.: Support for quality metrics in metamodelling. In: GMLD (Graphical Modeling Language Definition) Workshop@ ECFMA 2013 (2013)
37. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. IEEE Softw. **11**(2), 42–49 (1994)
38. Mancioppi, M., Perepletchikov, M., Ryan, C., van den Heuvel, W.J., Papazoglou, M.P.: Towards a quality model for choreography. In: Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops, pp. 435–444. Springer, Berlin (2010)
39. Mandran, N., Dupuy-Chessa, S., Rieu, D., et al.: Démarche centrée utilisateur pour une ingénierie des langages de modélisation de qualité. Ingénierie des systèmes d'information **18**(3) (2013)
40. Moody, D.: The physics of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. **35**, 756–779 (2009)
41. Nelson, H., Poels, G., Genero, M., Piattini, M.: A conceptual modeling quality framework. Softw. Qual. J. **20**(1), 201–228 (2012)
42. Nysetvold, A.G., Krogstie, J.: Assessing business process modeling languages using a generic quality framework. In: Proceedings of the CAiSE'05 Workshops, pp. 545–556. Idea, Group (2005)
43. OASIS: Web services business process execution language v2.0 (ws-bpel 2.0). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel (2007)
44. OMG: Object Management Group. http://www.omg.org/ (1989)
45. OMG: Business Process Model and Notation (BPMN 2.0). http://www.omg.org/spec/BPMN/2.0/ (2011)
46. Peltz, C.: Web services orchestration and choreography. Computer pp. 46–52 (2003)
47. Recker, J., Indulska, M., Rosemann, M., Green, P.: Do process modelling techniques get better? A comparative ontological analysis of BPMN. In: Proceedings of the 16th Australasian Conference on Information Systems, Sydney, Australia. Citeseer (2005)
48. Recker, J.C., Indulska, M., Rosemann, M., Green, P.: How good is BPMN really? Insights from theory and practice. In: Ljungberg, J., Andersson, M. (eds.) 14th European Conference on Information Systems. Goeteborg, Sweden (2006). http://eprints.qut.edu.au/4636/
49. Rossi, M., Brinkkemper, S.: Complexity metrics for systems development methods and techniques. Inf. Syst. **21**(2), 209–227 (1996)
50. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131–164 (2009)
51. Schönberger, A.: Visualizing B2Bi choreographies. In: Proceedings of the 4th IEEE International Conference on Service-Oriented Computing and Applications (SOCA'11), Irvine, California, USA. IEEE (2011)
52. Schönberger, A., Wilms, C., Wirtz, G.: A requirements analysis of integration business to business integration. Technical Report, Otto-Friedrich-Universitt Bamberg, Bamberger Beitrge zur Wirtschaftsinformatik und Angewandten Informatik (2009)
53. Siau, K., Cao, Q.: Unified modeling language: a complexity analysis. J. Database Manag. (JDM) **12**(1), 26–34 (2001)
54. Silver, B.: BPMN Method and Style: A Levels-Based Methodology for BPM Process Modeling and Improvement Using BPMN 2.0. Cody-Cassidy Press, US (2009)
55. van der Aalst, W., Ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. Distrib. Parallel Databases **14**(1), 5–51 (2003)
56. W3C: Web services choreography description language version 1.0 (ws-cdl)—w3c candidate recommendation. http://www.w3.org/TR/2004/WD-ws-cdl-10-20041217/ (2005)
57. Wahl, T., Sindre, G.: An analytical evaluation of BPMN using a semiotic quality framework. In: Advanced Topics in Database Research, vol. 5 (2006)
58. Weidlich, M., Barros, A., Mendling, J., Weske, M.: Vertical alignment of process models - how can we get there? In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) Enterprise, Business-Process and Information Systems Modeling. 10th International Workshop, BPMDS 2009, and 14th International Conference, EMMSAD 2009, and 14th International Conference, EMMSAD 2009, CAiSE 2009. Lecture Notes in Business Information Processing, vol. 29, pp. 71–84. Springer, Heidelberg (2009)
59. Wetzstein, B., Karastoyanova, D., Kopp, O., Leymann, F., Zwink, D.: Cross-organizational process monitoring based on service choreographies. In: Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 2485–2490. ACM (2010)
60. Wohed, P., Aalst,W., Dumas, M., Hofstede, A., Russell, N.: On the suitability of BPMN for business process modelling. In: Dustdar, S. Fiadeiro, J.L., Sheth, A.P. (eds.) Business Process Management, 4th International Conference, BPM 2006. Lecture Notes in Computer Science, vol. 4102, pp 161–176, Springer, Heidelberg (2006). doi:10.1007/11841760_12
61. Zaha, J., Barros, A., Dumas, M., ter Hofstede, A.: Lets dance: a language for service behavior modeling. On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, pp. 145–162 (2006)

Mario Cortes-Cornax is a PhD student since 2010 at the University of Grenoble. He holds a double degree in computer science engineering from the "Universidad Politécnica of Madrid" (UPM) and the "École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble" (ENSIMAG). He got a research master in the domain of information systems and software engineering (Grenoble). His research interest concerns information system design, business process management and model-driven engineering.

Sophie Dupuy-Chessa is assistant professor at University of Grenoble since 2002. She got her PhD thesis in 2000 in the domain of software engineering, more precisely in software modeling. Then she has two postdoctoral positions: She was lecturer at University of Geneva and research scientist at Xerox Research Centre Europe. Currently, her research interest concerns model-driven engineering for human-computer interaction and information system design. She published papers in the software engineering, the information systems and the HCI domains.

Dominique Rieu is full professor at Pierre Mendès France University (UPMF) in Grenoble. She is member of Laboratory of Informatics of Grenoble (LIG). Her research interests concern process aware information systems engineering and method engineering. She is responsible of the theme ≪ engineering for and by models in information systems ≫ of GDR I3 (Information, Interaction, Intelligence), president of INFORSID, the French association of researchers and industrials in Information System.

Nadine Mandran is an engineer in the human and social sciences, especially in methodologies for the production and data analysis. Her research interest concerns the adaptation of methods developed for the human and social sciences to the field of computer sciences. She mainly works in the domains of information systems engineering, method engineering and technology enhanced learning (TEL). She is in charge of the quality mission in the Laboratory of Informatics of Grenoble (LIG).