On Formal Choreographic Modelling: A Case Study in EU Business Processes

Published Version:
Alex Coto, F.B. (2022). On Formal Choreographic Modelling: A Case Study in EU Business Processes. Cham : Springer [10.1007/978-3-031-19849-6_13].

Availability:
This version is available at: https://hdl.handle.net/11585/907472 since: 2022-11-24

Published:
DOI: http://doi.org/10.1007/978-3-031-19849-6_13

Coto, A., Barbanera, F., Lanese, I., Rossi, D., Tuosto, E. (2022). On Formal Choreographic Modelling: A Case Study in EU Business Processes. In: Margaria, T., Steffen, B. (eds) Leveraging Applications of Formal Methods, Verification and Validation. Verification Principles. ISoLA 2022. Lecture Notes in Computer Science, vol 13701. Springer, Cham.

The final published version is available online at: https://doi.org/10.1007/978-3-031-19849-6_13

On Formal Choreographic Modelling: a Case Study in EU Business Processes[⋆]

Alex Coto[1], Franco Barbanera[2], Ivan Lanese[3], Davide Rossi[4], and Emilio Tuosto[1]

[1] Gran Sasso Science Institute (Italy)
[2] Dept. of Mathematics and Computer Science, University of Catania (Italy)
[3] Focus Team, University of Bologna/INRIA (Italy)
[4] University of Bologna (Italy)

Abstract. Formal choreographic modelling advocates a *correctness-by-construction* principle for the development of sound communication protocols. This principle usually hinges on syntactic or semantic restrictions to rule out models that could lead to communication glitches like message losses or deadlocks.

This paper explores how these restrictions impact on the usability of formal modelling. More precisely, we benchmark the use of a formal choreographic modelling language designed to support the correctness-by-construction principle of message-passing systems. To this purpose, we consider the formal choreographic modelling of real business processes taken from the official documentation of European customs business process models. In fact, following a steadily increasing trend, the European Union started to use BPMN to support the legal provisions of the customs business process models.

1 Introduction

Choreographic models [26] are becoming popular in many application domains. For instance, the *business process modelling notation* (BPMN) [25] is used to design and document both software [27] and other kinds of workflows such as those in the health domain (e.g., [22]) or, as for the case study of this paper, in public administrations (e.g., [30]). Such models are described by means of notations presenting a wide range of levels of formalisation. Semi-formal notations like BPMN are rather expressive and flexible since they tend to feature several mechanisms to ease the representation of complex processes. As a matter of fact, semi-formal approaches are customary among practitioners and usually, once requirements are collected, they are the first step taken in the development phase. Formal approaches, instead, have been developed - mostly in academia - with the

---

[⋆] Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska-Curie grant agreement No 778233, by the MIUR project PRIN 2017FTXR7S "IT-MaTTerS" (Methods and Tools for Trustworthy Smart Systems). and by the Progetto di ateneo Unict PIA.CE.RI. 2020-2022 Linea 2. Barbanera, Lanese and Tuosto have also been partially supported by INdAM as members of GNCS (Gruppo Nazionale per il Calcolo Scientifico).

aim of supporting the so-called *correctness-by-construction* principle. Accordingly, a choreography holistically describes the interactions of a system and enables the "automatic" extraction of a skeleton description of its single components. Key to this approach is the guarantee that systems obtained that way do enjoy relevant communication properties such as lock-freedom. Researchers strive for the development of solid theories to support the correctness-by-construction principle. Ideally, it would be desirable to attain theories and tools that are flexible and expressive enough to be largely applicable in practice. The present paper analyses through a practical example the distance between a semi-formal and a formal model. We hope that our small *hands-on* analysis sheds some light on the usefulness of formal models.

The correctness-by-construction principle mentioned above hinges on the interplay between the so-called *global* and a *local* view of a formal choreography. The former view specifies the observable behaviour of the process at hand *as a whole* while the latter view specifies the behaviour of each component *in isolation*. The usual design approach rests on two steps:

- First, the designer defines a global view of the process that enjoys suitable properties, often dubbed *well-formedness* conditions.
- Then, local views (and possibly executable artefacts) are algorithmically derived from the global view by means of a *projection* operation guaranteeing relevant communication properties when applied to well-formed global views.

Often, well-formedness conditions are defined in terms of syntactic or semantic restrictions. Several formal modelling languages have been proposed in the literature to realise this correcteness-by-construction principle (see [14,6,28,19,24,10,29,3,2] to mention but a few, others can be found in the survey in [21]).

In this paper we address the following question:

to what extent well-formedness conditions hinder modelling?

We consider the formal modelling of real business processes taken from the official documentation of European customs business process models available at `https://aris9.itsmtaxud.eu/businesspublisher`. The documentation of these processes is heavily based on BPMN [25] specifying the interactions among participants involved in the bureaucracy of legal provisions. The type of BPMN diagram adopted for this specification is a collaboration diagram, that is a diagram that can represent both the inner structure of the business process of each involved party (in terms of tasks, choices and so forth) and the structure of the interactions among these parties (in terms of message exchanges). As we shall see, this greatly helps in connecting our formalisation to the case study.

An exhaustive analysis on the family of formal choreographic formalisms would be beyond our possibilities. We therefore choose *global choreographies* [29] (g-choreographies for short) because they have a straightforward semantics and feature an intuitive visual presentation akin to BPMN diagrams, so easing the comparison between the two. We refer the reader to [29] for a formal presentation of g-choreographies; here we will overview them in § 2 and exploit their visual notation to informally describe them by examples in § 4.

The correcteness-by-construction principle is rendered in g-choreographies by requiring the so called *well-branchedness* condition [29]. Intuitively, this condition requires that each distributed choice

- has a unique *active* participant, that is only one of the participants of the choice decides which branch has to be followed
- any participant other than the active one is *passive*, namely it either (i) has exactly the same observable behaviour in each branch, or (ii) it is (unambiguously) notified of the choice by a participant already aware of the decision.

Our comparative analysis highlights that well-formedness conditions of formal choreographic models have an heavy impact on the modelling of the processes. As we argue in § 5, this is not only due to the choice of g-choreographies as choreographic formalism but, more generally, to the constraints required to ensure the correcteness-by-construction principle.

Besides the typical fact that assumptions have to be made when semi-formal documentation of processes is ambiguous or incomplete, we have observed that the similarity between BPMN and g-choreographies cannot always be exploited. In particular, we have observed that optional sub-processes do clash with the restrictions imposed by g-choreographies and, as argued in § 5, with most of the formal approaches in the literature. More crucially, exploiting similarities may actually be misleading and leads to models that do not faithfully reflect the intended behaviour of the process. This problem can be resolved using design languages that are not well-structured (yet formal), at the cost of renouncing to exploit the affinity.

Structure of the paper § 2 presents a brief background on BPMN and g-choreographies. § 3 introduces our case study. § 4 shows the g-choreographies of the case study. § 5 reports the analysis of our modelling exercise. Finally, § 6 draws some conclusions and sketches future work.

2 Background

In order to support the modelling of business processes, the Object Management Group (`https://www.omg.org`) introduced the Business Process Model and Notation (BPMN) specification [25]. The main aim of this specification is to provide a notation comprehensible to business users while capturing complex process semantics for technical users. In fact, BPMN is the de facto standard used to describe business processes. In this context a business process is a generic process, in the sense of a coordinated set of activities aiming at achieving a goal, where the goal is the production of goods or the supply of services. BPMN has been designed for a wide array of business users interested in different viewpoints; this translates into the definition of four different diagrams (and their respective underlying metamodels): process, collaboration, conversation and choreography.

Process diagrams focus on detailing the internal structure of a business process. The process is modeled as a network of flow objects (events, activities and gateways) connected by connecting objects (sequence flows). The execution of a process can be thought of as a flow of activities described by the traversing of the process' network by one or more tokens moving through flow objects, carried by connecting objects.

Collaboration diagrams depict the cooperation of different partners' business processes in terms of message exchanges (where message flows are an additional kind of connecting objects). Each partner's internal process can be fully described by a process diagram contained in the partner's pool, or the internal process can be kept opaque by representing the partner as a collapsed pool. Figure 1, detailed later in the paper, depicts a collaboration diagram. Three parties (at the top) are represented by collapsed pools. A fourth participant (at the bottom) is represented by a pool containing the process diagram that details its internal behavior. The interactions between the parties take place in the form of message exchanges represented by message flow (the dashed arrows connecting the various pools).

Choreography diagrams focus on how interactions between participants take place and abstract completely from the internal behavior of each participant. Their network is composed of basic message exchanges (choreography task), a reduced set of events and gateways connected by sequence flows.

Conversation diagrams focus on other interaction-related aspects and are outside the scope of this paper.

In the context of the present work, BPMN choreography diagrams are most definitely relevant, however these diagrams are rarely used, possibly because of their rather convoluted semantics. For example, out of about ∼114000 BPMN models present in various projects in GitHub[5] only 333 of them are choreographies (less than 0.3%).

The case study we present mostly adopts BPMN collaboration diagrams. While our focus stays on the interaction between participants, the modeling of the behavior of some of them represented in the diagrams allows us to better understand their internal logic.

To model the processes described in § 3 we adopt the language of *global choreographies* [29] (g-choreographies for short). The reason to choose g-choreographies is that the definition of their formal syntax and semantics is equipped with an intuitive visual presentation akin to BPMN diagrams. In fact, we will appeal to this intuitive presentation and refer to [29] for the technical definitions.

The basic elements of g-choreographies are interactions of the form $\mathsf{A} \rightarrow \mathsf{B} \colon \mathsf{m}$ where A and B are distinct *role names* and m is a *message type* drawn from a given set $\mathcal{M}$ (disjoint from the set $\mathcal{P}$ of role names). Intuitively, $\mathsf{A} \rightarrow \mathsf{B} \colon \mathsf{m}$ represents the fact that participant A sends to B the message m (and B receives it). A g-choreography is basically interpreted as a regular language over an

---

[5] These models have been identified by (i) searching through the GitHub API the files with the extension `.bpmn` and then (ii) using pattern matching to identify the ones describing choreographies.

alphabet of interactions. For simplicity, we do not consider here another key feature of g-choreographies, namely parallel composition. In fact, the scenarios we consider in § 3 do not require parallel composition. However, in practice parallel composition is sometimes needed. For instance, the documentation for the amendment process discussed in § 3 specifies a procedure which is independent of the ones we consider here and can be formalised using parallel composition of g-choreographies[6]. Instead of giving a formal definition of g-choreographies, we will describe them using examples drawn from our scenarios. The following remarks are worthwhile. In selecting our scenarios we strove to identify parts of the BPMN models that have a straightforward counterpart in g-choreographies as well as parts where the connection is not that clear cut. This, and our consequent modelling choices, are instrumental to make the points of § 5.

3 Legal provisions

In 2010 the European Union introduced the *customs business process models* [13] on request of member states' customs authorities. These processes are expressed for the most part as a mix of text and BPMN diagrams [25] collected in the repository at `https://aris9.itsmtaxud.eu/businesspublisher` (and publicly available by following the 'Anonymous access' link there). The purpose of these models is to document and facilitate the reading of the legal provisions.

We will focus on the workflow of the procedures for the entry of goods. A key element within these procedures is the *entry summary declaration* (ENS), a document submitted by *declarants* stating relevant information for the import (nature of the goods, carrier, etc.). These procedures activate several tasks which depend on multiple factors such as where ENS documents were lodged and the transport infrastructures through which the goods enter (i.e. sea, air, or road&rail). Also, after its submission, an ENS goes through a validation process. The validation culminates either in the registration of the ENS or in the request of amendments or additional information. In the latter case, the EU regulations require to notify these requests to the involved parties.

Amendment We illustrate part of this procedures starting by briefly describing how an amendment[7] gets processed on the *customs office of first entry* (COFE). The process is documented by the `L3-ENT-01-05-Process ENS At Office Of First Entry - Amendment` diagram in the repository, reported in Fig. 1.

Upon being received by the COFE, the amendment goes through an initial verification process to check, for example, whether the amendment concerns previously submitted information. If the amendment of the ENS is deemed valid, it is registered and multiple participants are notified; according to the documentation,

---

[6] A precise formalisation would involve also other considerations that do not emerge very clearly from the current documentation.

[7] Amendments may be necessary due to changes of e.g., carriers, or dates of travel of the goods, etc.

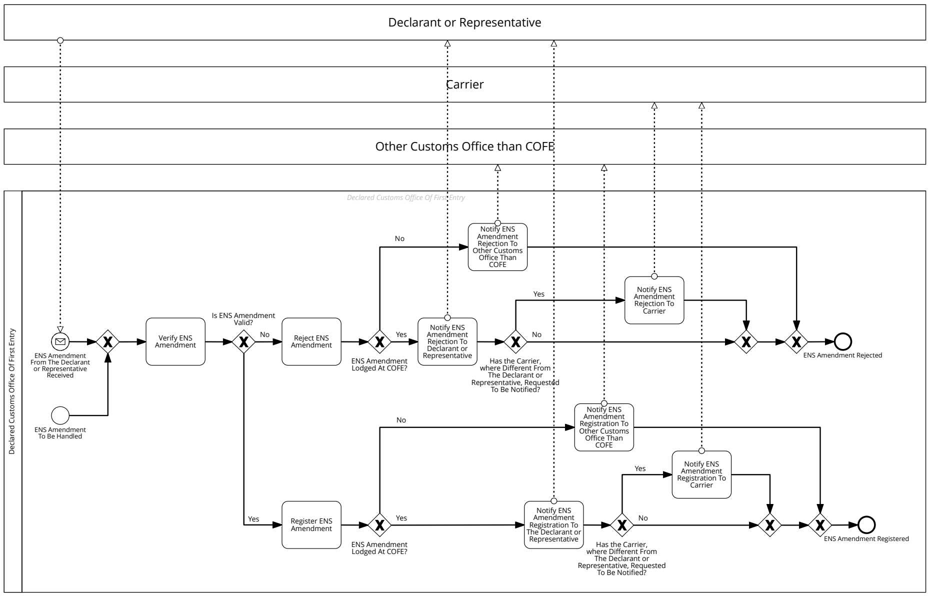

Fig. 1. BPMN specification of the amendment process

- if the amendment was lodged at the COFE, a registration notification is sent to the *declarant*;
- a *carrier* is notified when different from the declarant and asks to be notified;
- if, instead than at COFE, the amendment was lodged in another customs office (OCOFE) then the OCOFE has to be notified;

If the amendment is invalid instead, the participants are notified of the rejection following a communication pattern for notifications similar to the one above.

Arrival of goods A more elaborate process describes the arrival of goods. For the purpose of illustration, let us focus on the case where the goods arrive through road and rail as depicted in the set of diagrams collected in the repository (cf. `L3-ENT-01-04-Process ENS At Office Of First Entry-Road and Rail`), one of which is reproduced[8] in Fig. 2. The diagram in Fig. 2 is more complex than the one in Fig. 1; in fact, it contains nested choices with a loop involving several participants. We now comment on the procedure designed for the arrival of goods as depicted in Fig. 2.

When a full or partial ENS is received at the declared COFE, a first check occurs for the validity and consistency of the information provided. If the ENS is deemed invalid, it is rejected, and this is notified to the declarant so that they

---

[8] The diagram is hardly readable; it is reported only to give readers a hint on the complexity of these models. Some details of the model can be inferred from the g-choreography Fig. 4 (cf. § 4) corresponding to the diagram in Fig. 2.

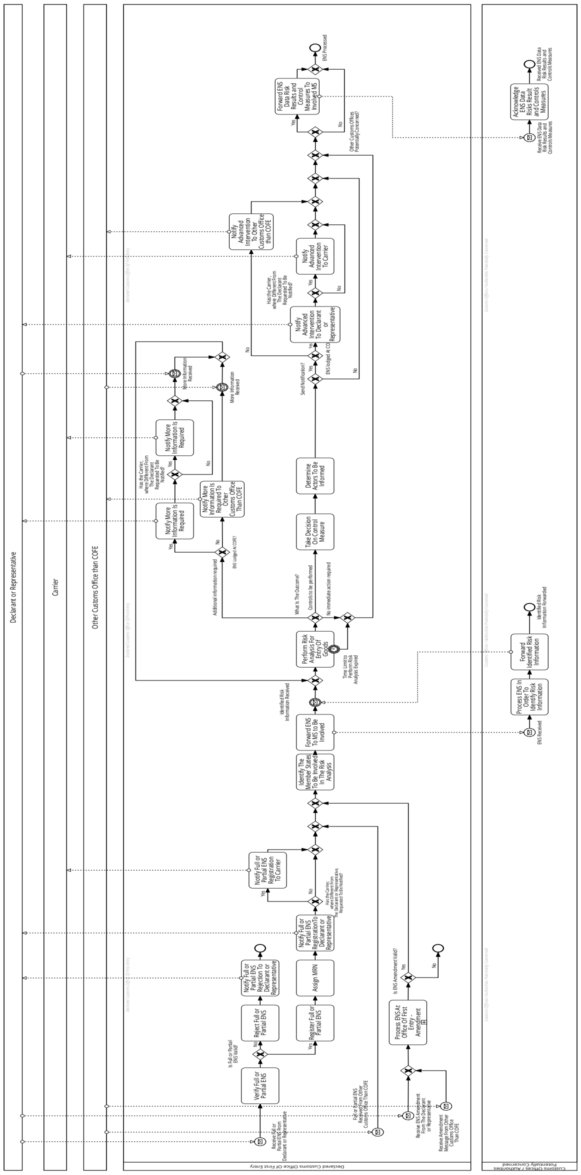

Fig. 2. BPMN specification of the process for the arrival of goods

can optionally send amendments and iterate the above procedure. Otherwise, the ENS is registered and automatically assigned a *master reference number* (MRN), which is stored in the system. This registration is notified to the declarant and optionally to the carrier (under the same conditions as for the amendment procedure).

At this point, a risk analysis involving some member states needs to occur. Therefore, the ENS is forwarded to the states and the custom offices deemed to be involved in the risk analysis. These offices should send back information about any identified risks. If any additional information is required by the COFE or the OCOFE (depending on where the ENS was lodged to), the declarant and (optionally) the carrier are requested to send such information to the relevant office. Also, there might be the need for additional controls to be performed, in which case some of the involved participants (i.e., Declarant, Carrier, and OCOFE) might need to be notified, in a similar fashion as when additional information is required. Once the additional controls (if any) are performed, the ENS Data Risk results and control measures should be forwarded to the involved member states customs offices.

Simplifying assumption The processes described above are part of a larger workflow, the formalisation of which falls out of the scope of this paper. In fact, in the next sections we will assume that the declarant decided to lodge the ENS to the COFE. The other option would yield basically the same g-choreographies; hence we do not consider it, so reducing the amount of branching and avoiding repetitions. Another assumption is that carrier and declarant do not coincide; the other case can simply be obtained by removing from our formalisation all the branches involving the carrier.

4 Formal models

We now describe the g-choreographies corresponding to the BPMN diagrams of our case studies. These models have been manually produced from the documentation available at `https://aris9.itsmtaxud.eu/businesspublisher`. We believe that this process could be partly automated by "compiling" the BPMN models into g-choreographies; however, this is not in the scope of this paper.

Amendment A g-choreography capturing the amendment process is given by the diagram in Fig. 3. The intuitive graphical notation uses *gates* to represent the combinators of the language; in particular branching and merging points of a non-deterministic choice (and loops) are represented by ◈-gates. Noteworthy, the branching structure of the process in Fig. 3 reflects the one of the BPMN diagram in Fig. 1. Actually, any g-choreography is *well-structured* in the sense that gates behave as well-balanced parenthesis: for each *opening* gate there is a corresponding *closing* one. (In passing we note that well-structuredness is also considered in BPMN modelling and it is studied in e.g., [9] and references therein.)

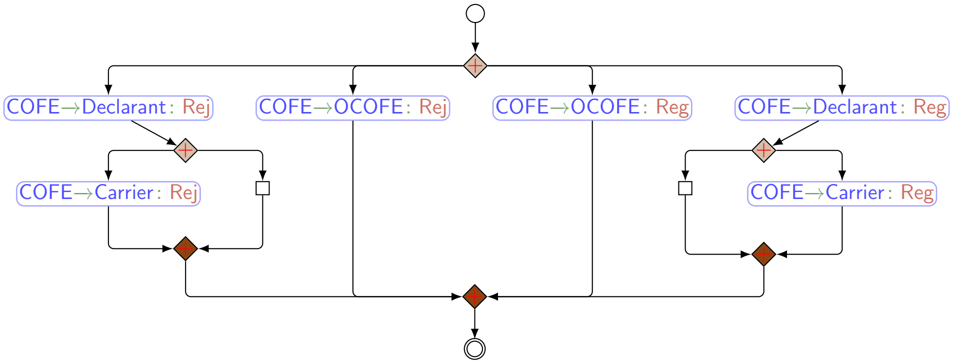

Fig. 3. The amendment process as g-choreography

Well-structuredness is pictorially represented in Fig. 3 by making closing gates darker than opening ones. For instance, the topmost (opening) ⊕-gate is matched by the bottommost (closing) one; note[9] that this is not the case for the BPMN diagram of Fig. 1 where the second diamond from the left is not matched by any closing gateway. The ○- and the ◎-gates mark the start and the end of the g-choreography, respectively.

The semantics of a g-choreography can be described in intuitive terms as the sequences of interactions[10] from the ○-gate to the ◎-gate. For instance, in the diagram of Fig. 3 the topmost ⊕-gate allows four possible branches that capture the options that COFE has to consider when validating an ENS. The following remarks are worthwhile for our considerations in § 5. Firstly, the diagram in Fig. 3 should be considered embedded in a larger process where Declarant had previously sent the ENS either to COFE or OCOFE. Secondly, as per the informal description of the scenario in § 3, the diagram abstracts away from the details about how COFE decides the next course of action. Finally, the innermost choices do not respect the conditions that g-choreographies require to guarantee correctness (which will be described in § 5).

Arrival of goods The g-choreography for the arrival of goods process, shown in Fig. 4, is rather more complicated than the one for the amendment. Since the former process has more optional interactions than the latter one, Fig. 4 has several ⊕-gates with a branch with an empty box directly connected to the corresponding closing gate; this visually represents the fact that the choice is possibly resolved without explicit interactions. As said, the diagram is obtained from the `L3-ENT-01-04-Process ENS At Office Of First Entry - Road and Rail` BPMN present in the repository as depicted in Fig. 2.

---

[9] This remark will be reconsidered in § 5.

[10] Formally, these are the words generated by the g-choreography interpreted as a regular expression on an alphabet of interactions.

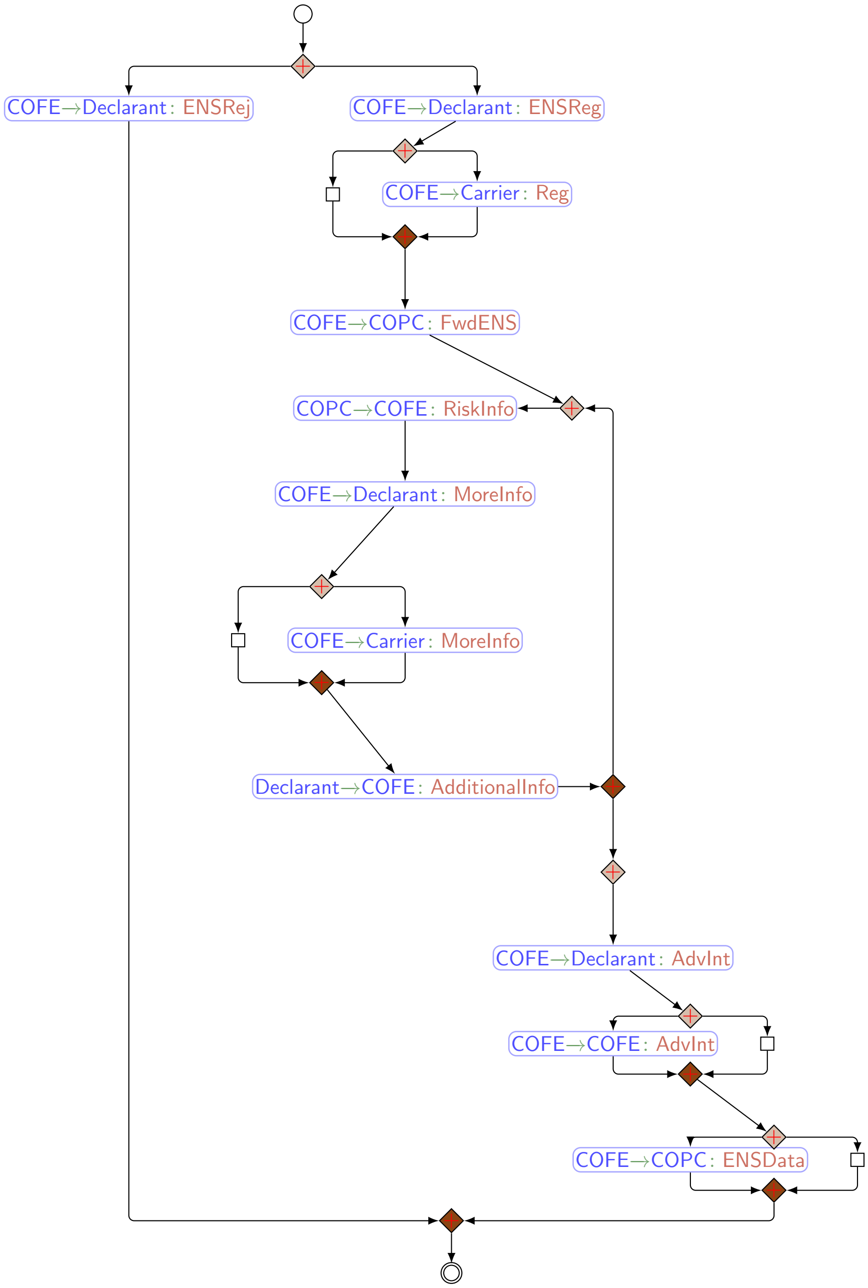

Fig. 4. The arrival of goods process as g-choreography

A source of complexity in this diagram is the presence of the loop for the request of additional information whose entry point is the rightmost ⟐-gate. This loop is structurally isomorphic to the corresponding one in the BPMN diagram. We will see in § 5 that this is not however a correct formalisation of the process due to the different semantics.

5 From BPMN to Global Choreographies

This section discusses the nuances of the formalisation of the scenarios with g-choreographies. A definite advantage has been the use of the BPMN diagrams specifying procedures as starting point; indeed these diagrams greatly help in understanding the processes, without having to dive in the actual legislation. In fact, this allowed us to exploit the similarities between g-choreographies and BPMN. Nonetheless, several assumptions and modelling decisions had to be taken due to factors which are worth discussing here. In fact, we believe that they would occur regardless of the formal choreographic language used.

The BPMN diagrams of our scenarios heavily rely on value passing. In fact, several decisions cannot be taken without assuming that some information is properly propagated to participants. For instance, the decision of notifying the carrier is taken according to some information somehow available at COFE through an interaction between carrier and COFE, presumably after carrier and declarant agreed on the ENS. In a modelling language which abstract away from data, such choices either become non deterministic choices (like in our g-choreographies) or explicit interactions. We opted for the first alternative to maintain an higher degree of similarity between BPMN diagrams and g-choreographies as well as because it is not always straightforward to extract the conditions determining such choices from the documentation. For instance, it is not clear from the documentation if an ENS where declarant and carrier coincide can be amended to one where they are different; also, the interpretation of the BPMN diagram would be drastically different if this amendment were possible. This decision has however drawbacks which we now spell out.

A first problem is the one announced at the end of § 4. The loop in Fig. 4 seemingly corresponds to the one in the BPMN diagram but, actually, it has a semantics different than the intend one. In the BPMN diagram, the notification to the carrier is either sent in all iterations or in none of them; this is possible because the decision of COFE is taken according to previously received information which does not change during the iteration. Instead, this is not the case in the g-choreography of Fig. 4: at each iteration COFE can non-deterministically decide to perform the notification or not. To faithfully model the intended behaviour the loop should be preceded by a choice where COFE decides either to iterate and always send the notification or to iterate just the interactions with the declarant.

More in general, choreographic formalisms ensuring correctness-by-construction require that choices are taken by a single participant and communicated to other involved participants via message passing, that is by sending them different messages. This rules out the possibility, quite common in practice and explicitly covered by BPMN's data-based exclusive gateway, that decisions are taken based on data. For instance, in the example above the choice of notifying the carrier or not depends on shared data.

The other issue is also related to the fact that formal choreographic languages are often conceived to support the correctness-by-construction principle. Correctness typically refers to communication properties such as deadlock freedom or guarantees about messages (e.g., no message loss). This requires to limit language expressivity. For instance, many variants of global types (see [21] and references therein) do not allow choices with empty branches. As seen, the BPMN models of the scenarios heavily rely on *optional* communications to some participants (e.g., the notification of rejection to the carrier). Unlike global types, g-choreographies can faithfully reflect this; in fact, optional communications have been rendered as g-choreographies with ◈-gates representing a non-deterministic choice where a branch is empty. However, this bluntly violates the *well-branchedness* condition required for the correctness of g-choreographies [29]. In fact, according to the semantics of g-choreographies, Carrier cannot be certain whether the ENS is rejected or there is a delay in the communication from COFE. An analysis of the g-choreography in Fig. 3 would flag this as a problem whose solution would force us to introduce some interaction on the empty branch. On the other hand, optional processes (like the notification to the carrier) occur rather often in practice. Optional behaviour is supported in *choreography automata* [2], according to the refined theory recently proposed in [15,16]. Optional behaviour can also be specified in [18,20], but there if a participant joins only a part of a choreography, explicit connection and disconnection actions are required. Indeed, choreography automata allow *selective participation*, that is there can be participants occurring only on some of the branches of a non-deterministic choice.

As discussed in § 4, g-choreographies are well-structured. This is often a desirable property of design languages and basically all syntax-based choreographic modelling languages are well-structured. In practice however, well-structured languages may be forced —by their own nature— to handle some typical situations in a cumbersome way. For instance, the BPMN diagram `L3-ENT-01-04-Process ENS At Office Of First Entry - Road and Rail` allows amendments to an ENS to arrive while the ENS itself is being processed. This would basically correspond to "jumping" from one branch to another of a choice, which is not allowed by well-structuredness. To circumvent this problem one has to replicate the common behaviour on each branch. Notice that this might not be always straightforward (e.g., when the replication is inside an iteration). In modelling languages such as choreography automata, which are not well-structured, this problem is less acute.

6 Conclusions and Future Work

This work just scratched the surface of the comparison between BPMN modelling, widely used in practice, and formal choreographic approaches, which have been deeply investigated theoretically (see the survey in [21], and in particular all the thread on multiparty session types [19]). We are not aware of other similar attempts while many works study formal semantics and analysis techniques for BPMN (see e.g., [9] and references therein). Formal choreographic models are abstractions built to support the correctness-by-construction principle. Typically, these formalisms abstract away from many aspects and focus on specific aspects "in vitro". For instance, some formal models of choreographies feature value passing [4,5,23,15], some others have considered adaptive choreographies [8,10,7,18]. Nonetheless, a formalism comprehensively combining these features is missing and, perhaps, not even worthwhile since it would probably be so complex to be not applicable in practice.

As we have discussed, easy modelling calls for generalised conditions on choreographic models, e.g., participants may be involved only in some branches of a choice. This is in particular the case for optional actions. Such a generalization have been recently considered in [18,15], but they are still missing from many approaches. On the other side, we believe that it is best practice to avoid diagrams which are obscure to the reader, and even less suitable for automatic analysis. An example here is the case of choices which depend on data. We believe that what triggers such choices should instead be made apparent by the messages used in the communications.

Another note is that we found out that most of the available diagrams, including the ones we selected as case studies, represent the view of a system from a single participant (e.g., COFE in our case study). It would be good to develop techniques for composing such local views into a global view along the lines in [1] and in [12].

References

1. F. Barbanera, M. Dezani-Ciancaglini, I. Lanese, and E. Tuosto. Composition and decomposition of multiparty sessions. *J. Log. Algebraic Methods Program.*, 119:100620, 2021.
2. F. Barbanera, I. Lanese, and E. Tuosto. Choreography automata. In S. Bliudze and L. Bocchi, editors, *Coordination Models and Languages - 22nd IFIP WG 6.1 International Conference, COORDINATION 2020, Held as Part of the 15th International Federated Conference on Distributed Computing Techniques, DisCoTec 2020, Valletta, Malta, June 15-19, 2020, Proceedings*, volume 12134 of *Lecture Notes in Computer Science*, pages 86–106. Springer, 2020.
3. D. Basile, M. H. ter Beek, and R. Pugliese. Synthesis of orchestrations and choreographies: Bridging the gap between supervisory control and coordination of services. *Logical Methods in Computer Science*, 16(2), 2020.
4. L. Bocchi, K. Honda, E. Tuosto, and N. Yoshida. A theory of design-by-contract for distributed multiparty interactions. In P. Gastin and F. Laroussinie, editors, *Concur 2010*, volume 6269 of *LNCS*, pages 162–176. Springer, 2010.
5. L. Bocchi, H. C. Melgratti, and E. Tuosto. On resolving non-determinism in choreographies. *LMCS*, abs/1904.08337, 2019. To appear.
6. N. Busi, R. Gorrieri, C. Guidi, R. Lucchi, and G. Zavattaro. Choreography and orchestration conformance for system design. In *Coordination Models and Languages, 8th International Conference, COORDINATION 2006, Proceedings*, pages 63–81, 2006.

7. I. Castellani, M. Dezani-Ciancaglini, and J. A. Pérez. Self-adaptation and secure information flow in multiparty communications. *Formal Asp. Comput.*, 28(4):669–696, 2016.
8. M. Coppo, M. Dezani-Ciancaglini, and B. Venneri. Self-adaptive multiparty sessions. *Service Oriented Computing and Applications*, 9(3-4):249–268, 2015.
9. F. Corradini, A. Morichetta, C. Muzi, B. Re, and F. Tiezzi. Well-structuredness, safeness and soundness: A formal classification of BPMN collaborations. *J. Log. Algebraic Methods Program.*, 119:100630, 2021.
10. M. Dalla Preda, M. Gabbrielli, S. Giallorenzo, I. Lanese, and J. Mauro. Dynamic choreographies: Theory and implementation. *Log. Methods Comput. Sci.*, 13(2), 2017.
11. U. de'Liguoro, H. C. Melgratti, and E. Tuosto. Towards refinable choreographies. In J. Lange, A. Mavridou, L. Safina, and A. Scalas, editors, *Proceedings 13th Interaction and Concurrency Experience, ICE 2020, Online, 19 June 2020*, volume 324 of *EPTCS*, pages 61–77, 2020.
12. U. de'Liguoro, H. C. Melgratti, and E. Tuosto. Towards refinable choreographies. *Journal of Logical and Algebraic Methods in Programming*, 127:100776, 2022. Extended and revised version of [11].
13. EU customs BPMN. `https://ec.europa.eu/taxation_customs/customs-4/union-customs-code/ucc-bpm_en`.
14. X. Fu, T. Bultan, and J. Su. Conversation protocols: a formalism for specification and verification of reactive electronic services. *TCS*, 328(1-2):19–37, 2004.
15. L. Gheri, I. Lanese, N. Sayers, E. Tuosto, and N. Yoshida. Design-By-Contract for Flexible Multiparty Session Protocols. In K. Ali and J. Vitek, editors, *36th European Conference on Object-Oriented Programming, ECOOP 2022, June 6-10, 2022, Berlin, Germany*, volume 222 of *LIPIcs*, pages 8:1–8:28. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
16. L. Gheri, I. Lanese, N. Sayers, E. Tuosto, and N. Yoshida. Design-by-Contract for Flexible Multiparty Session Protocols (Artifact). *Dagstuhl Artifacts Ser.*, 8(2):21:1–21:5, 2022.
17. R. Guanciale and E. Tuosto. An abstract semantics of the global view of choreographies. In *Proceedings 9th Interaction and Concurrency Experience, ICE 2016, Heraklion, Greece, 8-9 June 2016.*, pages 67–82, 2016.
18. P. Harvey, S. Fowler, O. Dardha, and S. J. Gay. Multiparty session types for safe runtime adaptation in an actor language. In A. Møller and M. Sridharan, editors, *35th European Conference on Object-Oriented Programming, ECOOP 2021, July 11-17, 2021, Aarhus, Denmark (Virtual Conference)*, volume 194 of *LIPIcs*, pages 10:1–10:30. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
19. K. Honda, N. Yoshida, and M. Carbone. Multiparty asynchronous session types. *J. ACM*, 63(1):9:1–9:67, 2016. Extended version of a paper presented at POPL08.
20. R. Hu and N. Yoshida. Explicit connection actions in multiparty session types. In M. Huisman and J. Rubin, editors, *Fundamental Approaches to Software Engineering - 20th International Conference, FASE 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10202 of *Lecture Notes in Computer Science*, pages 116–133. Springer, 2017.
21. H. Hüttel, I. Lanese, V. T. Vasconcelos, L. Caires, M. Carbone, P. Deniélou, D. Mostrous, L. Padovani, A. Ravara, E. Tuosto, H. T. Vieira, and G. Zavattaro. Foundations of session types and behavioural contracts. *ACM Comput. Surv.*, 49(1):3:1–3:36, 2016.

22. Percorsi diagnostici terapeutici e assistenziali adenocarcinoma della prostata. `https://www.ispro.toscana.it/sites/default/files/ReteOncologica/PROSTATA.ALLEGATO.D.pdf`.
23. S.-S. Jongmans and P. van den Bos. A predicate transformer for choreographies. In I. Sergey, editor, *Programming Languages and Systems*, pages 520–547, Cham, 2022. Springer International Publishing.
24. I. Lanese, C. Guidi, F. Montesi, and G. Zavattaro. Bridging the gap between interaction- and process-oriented choreographies. In A. Cerone and S. Gruner, editors, *Sixth IEEE International Conference on Software Engineering and Formal Methods, SEFM 2008, Cape Town, South Africa, 10-14 November 2008*, pages 323–332. IEEE Computer Society, 2008.
25. Object Management Group. Business Process Model and Notation (BPMN). `https://www.omg.org/spec/BPMN/2.0.2/PDF`.
26. C. Peltz. Web services orchestration and choreography. *IEEE Computer*, 36(10):46–52, 2003.
27. R. Pillat, T. Oliveira, P. Alencar, and D. Cowan. BPMNt: A BPMN extension for specifying software process tailoring. *Information and Software Technology*, 57:95–115, 2015.
28. Z. Qiu, X. Zhao, C. Cai, and H. Yang. Towards the theoretical foundation of choreography. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007*, pages 973–982, 2007.
29. E. Tuosto and R. Guanciale. Semantics of global view of choreographies. *JLAMP*, 95:17–40, 2018. Revised and extended version of [17].
30. The r/ukpersonalfinance flowchart v3.0.7. `https://flowchart.ukpersonal.finance`.