Formal Approaches for Modeling and Analysis of Business Process Collaborations

Flavio Corradini[1], Fabrizio Fornari[1], Barbara Re[1(✉)], Lorenzo Rossi[1], Andrea Polini[1], Francesco Tiezzi[2], and Andrea Vandin[3]

[1] School of Science and Technologies, University of Camerino, Camerino, Italy
{flavio.corradini,fabrizio.fornari,barbara.re,lorenzo.rossi}@unicam.it
[2] Dipartimento di Statistica, Informatica, Applicazioni, University of Florence, Florence, Italy
francesco.tiezzi@unifi.it
[3] Scuola Superiore Sant'Anna, Pisa, Italy
Andrea.Vandin@santannapisa.it

Abstract. The paper introduces a general framework for handling BPMN graphical (semi-formal) models used for describing business process collaborations. It offers a systematic and architecturally comprehensive synthesis of the authors' prior work on the direct formalization of business process collaborations, the specification of their properties, and their verification and animation. The results exposed in this paper (and many others, indeed) have a direct "causal link" with what the authors learned collaborating with Rocco De Nicola; methodologies and techniques to the specification and design of complex systems and one of Rocco's scientific career "constants": formality!

Keywords: BPMN · Collaboration · Formalization · Verification · Animation

1 Introduction

BPMN [18] offers the potential to establish a standardized "lingua franca", readily understood by all business stakeholders, including business analysts, technical developers, and business managers, who are involved in the design, implementation, and monitoring of processes within single organizations or collaborations involving multiple interacting participants [3]. The widespread adoption of BPMN can be attributed to its intuitive graphical representation, which enhances the comprehension of business processes among all stakeholders. Furthermore, the availability of supportive tools in the market has significantly contributed to its increased acceptance [4]. Its use extends beyond traditional organizational environments to encompass more unstructured scenarios, particularly those involving IoT-aware business processes [2].

T. Margaria and B. Steffen (Eds.): ISoLA 2024, LNCS 15219, pp. 50–61, 2025.
https://doi.org/10.1007/978-3-031-73709-1_4

Given its semi-formal nature, the BPMN standard lacks formal characterization. However, the Business Process Management community widely recognizes that a robust formal framework for notation can significantly assist designers in thoroughly understanding their BPMN models and articulating as well as verifying model properties. Consequently, significant effort has been directed by the research community in recent years towards establishing a formal semantics for the BPMN notation.

This paper presents a comprehensive synthesis of the authors' previous findings regarding the formalization of business processes. The contributions offered are grounded in direct formalizations of BPMN semantics, expressed in terms of Labeled Transition Systems. This approach mitigates potential misinterpretations stemming from the use of natural language in standard specifications and addresses challenges associated with mapping BPMN to other formal languages, each equipped with its own semantics. Our operational semantics are delineated for a significant subset of BPMN elements, emphasizing the ability to model collaborations among organizations through message exchange. A notable aspect of this formalization is its capacity to handle multi-instance BPMN collaborations, focusing on the interaction among multiple instances, data, and messages. Another distinguishing feature is the capability of the semantics to model business processes with arbitrary topology. This allows designers to specify processes in alignment with real-world scenarios without the constraint of defining strictly structured models. The presented semantics pave the way for properties verification, such as safeness and soundness. In cases where verification becomes infeasible due to state explosion issues, we resort to animation, faithfully implementing the formal semantics to aid designers in identifying erroneous behaviors. In summary, the contributions are delineated in Table 1, providing an overview of the various approaches considering BPMN elements and supported analysis.

The findings presented in this paper and many others are directly influenced by the authors' collaboration with Rocco De Nicola. The methodologies and techniques for the specification and design of complex systems, which are central to Rocco's scientific career, particularly emphasize formality. More specifically, we benefit from the results coming from the Sensoria project [1].

The rest of the paper is organized as follows. Section 2 presents the BPMN standard and discuss a case study. Section 3 refers to the formalization related to collaboration diagram, while Sect. 4 extends the framework with multiple-instance and data interplay. Finally, Sect. 5 summarizes concluding remarks.

2 Background Notions

In this section, we first provide some basic notions on BPMN 2.0 elements that can be included in a collaboration diagram. Then, we present a simple scenario, used throughout the paper as a running example (Fig. 1).

Basic Notions on BPMN. **Pools** represent participants or organizations involved in the collaboration and include details on internal process specifications and related elements. Pools are drawn as rectangles, and they have a name

Table 1. Overview of our modelling and analysis approaches.

	[13]	[5]	[6]	[8]	[12]
BPMN modeling					
Pool	x	x	x	x	x
Multiple instance Pool	–	–	–	–	x
Activities (Task)	x	x	x	x	x
Activities (Receive Task, Send Task)	–	–	–	x	x
Activities (Sub-process)	–	–	–	x	–
Activities (Multiple instance task in parallel, and sequence)	–	–	–		x
Gateway (XOR, AND, OR)	x	x	x	x	x
Gateway (Event-Based)	–	x	x	x	x
Event (Start, End)	x	x	x	x	x
Event (Intermediate, Terminate)	–	x	x	x	x
Event (Start receive, Intermediate send, Intermediate receive, End send)	–	–	–	x	x
Connecting Edge (Sequence Flow, Message)	x	x	x	x	x
Data (Data object, Data collection, Data Store)	–	–	–	–	x
Analysis					
MAUDE's LTL	–	–	x	x	–
Statistical model checking – MultiVeStA	–	–	–	x	–
Animation	–	–	–	–	x

associated. BPMN allows to assign a multi-instance marker (three vertical lines) to a pool, representing multiple instances playing the same role.

Tasks represent a specific work to be performed within a process. They are drawn as rectangles with rounded corners. It is possible to distinguish between the following types of tasks: *Task* represents the performing a unit of work. *Send Task* represents the sending of a message. *Receive Task* represents the receiving of a message. *Multi-instance Parallel Task* represents performing a unit of work more than one time concurrently. *Multi-instance Sequential Task* represents performing a unit of work more than one time, one after the other.

Connecting Edges connect process elements inside or across different pools. *Sequence Edges* are solid connectors used to specify the internal flow of the process, thus ordering elements in the same pool, while *Message Edges* are dashed connectors used to visualize communication flows between pools.

Events are used to represent something that can happen. An event can be a *Start Event* representing the point from which a process starts, an *Intermediate Event* representing something that happens during process execution or an *End Event* representing the process termination. Events are drawn as circles. When an event is the source or target of a message edge, it is called *Message Event*. It is possible to distinguish between the following types of events: *Start Message Event* is a start event with an incoming message edge the event element catches a message and starts a process; *Catch Intermediate Event* is an intermediate

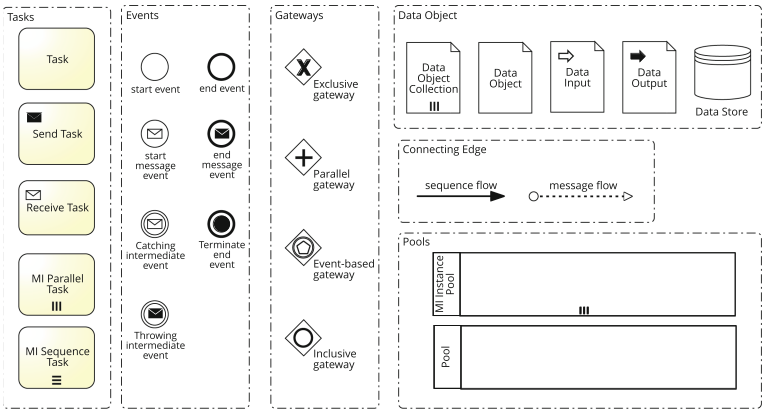

Fig. 1. BPMN elements.

event with an incoming message edge, the event element receives a message; *Throw Intermediate Event* is an intermediate event with an outgoing message edge, the event element sends a message; *End Message Event* is an end event with an outgoing message edge, the event element sends a message and ends a process. There is also a particular type of end event, the *Terminate End Event*, displayed by a thick circle with a darkened circle inside; it stops and aborts the running process - all the ongoing activities are aborted.

Gateways are used to manage the flow of a process both for parallel activities and choices. Gateways are drawn as diamonds and act as either join nodes (merging incoming sequence edges) or split nodes (forking into outgoing sequence edges). Different types of gateways are available: a *Exclusive Gateway* (or *XOR gateway*) gives the possibility to describe choices; a *Parallel Gateway* (or *AND gateway*) enables parallel execution flows; a *Inclusive Gateway* (or *OR gateway*) gives the possibility to select an arbitrary number of (parallel) flows; *Event-Based gateway* is used after a decision to fork the flow into branches according to external choice.

Data objects represent information and material flowing in and out of activities. They are depicted as a document with the upper-right corner folded over, and linked to activities with a dotted arrow with an open arrowhead (called *data association* in BPMN). The direction of the data association is used to establish whether a data object is an input or output for a given activity. Sometimes data objects can refer to a state: it can be done by appending the name of the state between square brackets to a data object's label. Different types of data objects are available: *Data Input* expresses (external) input data, and it can be read by an activity. *Data Output* expresses output data, and it can be generated by an activity. *Data Object Collection* expresses to multi-instance data objects; using this a designer can express the involvement of more than one Data Object. *Data Store* expresses data that persists after the process instance finishes.

As stated in the BPMN standard, a key concept related to the BPMN process execution refers to the notion of *token*. The BPMN standard states that *"a token is a theoretical concept that is used as an aid to define the behavior of a process that is being performed"* [18, Sec. 7.1.1]. A token is commonly generated by a start event, traverses the sequence edges , and is eventually consumed by an end event. Process elements retrieve one or more tokens from the incoming sequence flow to be executed. Once finished, they may produce one or more tokens on the outgoing sequence flows, depending on their behavior. In the collaboration, the process execution also triggers message flows that can generate messages. These will be referred to as message flow tokens.

Paper Reviewing Collaboration Model (Running Example 1/3). We introduce here a BPMN collaboration specification, borrowed from [10], used throughout the paper as a running example. The collaboration in Fig. 2 combines the activities of three participants. The Program Committee (PC) Chair organizes the reviewing activities. For the sake of presentation, we assume that the conference has only one chair, whose behavior is represented by the process within the PC Chair pool. A reviewer is someone knowledgeable about some of the conference topics. This role is modeled as a multi-instance pool. Indeed, each process instance describes the tasks one of the three process reviewers must accomplish to complete her/his assignment. Finally, the Corresponding Author is the person who submitted the paper to the conference and acts on behalf of the other authors. The PC Chair starts the reviewing process and assigns a paper to each reviewer. After all paper reviews have been received, the chair starts the evaluation. According to the value of the evaluation, when needed, the paper is discussed, and then the chair prepares the acceptance/rejection letter. After the review letter is prepared, the chair sends feedback to each reviewer and sends the result to the contact author.

3 BPMN Collaboration Diagrams: Formalisation and Verification

Our research journey started by precisely characterizing a subset of BPMN elements largely used in practice [13]. The first formalization was then extended with further elements [5]. Both papers used a pragmatic approach to select BPMN elements to be included in our formalization. Indeed we selected the elements most used in practice [4].

According to the results of the study we conducted in [4], we start with the formalization of BPMN's core elements (i.e. Task; Gateways - XOR, AND, OR, Event-Based; Event - Start, Intermediate, End; Sequence Flow), including also termination end events and communication-related elements (e.g., pool, and message) that are often overlooked by other formalizations (e.g. [15–17]). Our special emphasis on communication aspects within collaboration diagrams is mainly motivated by the need to achieve inter-organizational correctness. The formalization presents a multi-layer structure that ensures the decoupling of process behavior and interaction semantics at the collaboration level. It is suitable

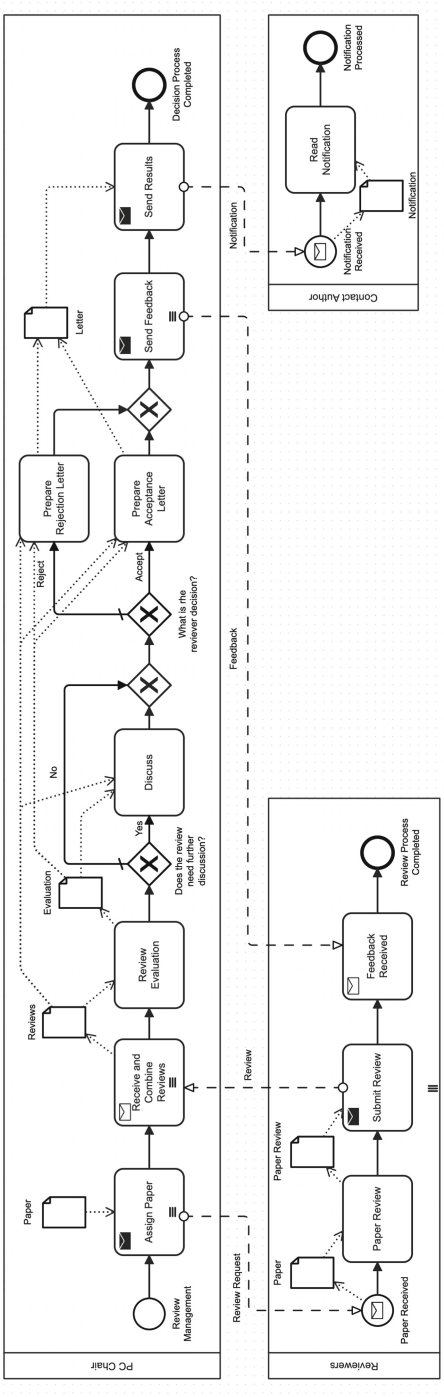

Fig. 2. Paper reviewing collaboration model.

for modeling business processes with arbitrary topologies and well-structured ones. In [5] the provided formalization is also implemented by exploiting the capabilities of Maude. This proof-of-concept implementation takes a collaboration model as an input and explores all the model executions by assessing the benefit of our approach in modeling and verification of BPMN collaborations. By relying on it, automatic verification of properties related to collaborations has been carried out via the Maude LTL model checker [14]. An optimization of this Maude implementation and its validation, as well as discussions on the envisioned users and on how the tool could be introduced in a general modeling methodology, are presented in [6,7].

Considered properties relate both to the internal characteristics of a single process in a collaboration, and to the whole collaboration itself. This holds both for generic properties that are well-established in the business process domain, such as safeness and soundness [9], and for ad-hoc properties specifically defined for given application scenarios. Informally, safeness refers to the occurrence of no more than one token at the same time along the same sequence edge of a process during its execution. Instead, soundness typically includes the following sub-properties: a process can always complete, once started (Option to Complete), and there exists no running or enabled activity for the process when it completes (Proper Completion). Safeness and soundness properties naturally extend to process collaborations, requiring that the processes of all involved organizations satisfy them considering the overall collaboration execution. Concerning ad-hoc properties, instead, in our running scenario for example we can check, at process level, if after completing a given task (say Assign Paper) a related one (say Send Results) can eventually complete. At the collaboration level, we can check, e.g., if in all cases the three processes involved in the airline collaboration successfully communicate with each other. This means that the processes are capable to abide by the protocol prescribed by the collaboration.

In deriving a verification approach for collaborative scenarios that could be used in real contexts, we identified two additional characteristics, which we judged of primary relevance, that was not implemented in the preliminary version of the approach. The first one concerns the fact that a collaborative scenario is by its very nature a parallel scenario, which then could easily lead to a possible explosion of the state space to be managed, making traditional verification strategies not always effective. Therefore, the approach has been augmented to include a statistical model checking strategy. The second characteristic refers instead to the typical difficulties of introducing formal verification techniques in general modeling contexts, where it is often the case that a modeler does not have a strong background in formal methods. So, to make the acceptance of the proposed approach easier, we decided to make available basic verification features via pre-configured and stereotyped properties. We provide a GUI where properties can be derived by selecting and composing the entries made available via a set of pop-up menus. In such a way, a modeler can start to use and experiment with our approach even without a clear understanding of the verified properties. This could help the interested modeler to acquire confidence in the

approach through its usage. It is worth noticing that the modeler is not asked to modify, in any way, his/her modeling habits. The approach includes optional mechanisms for those users with a good understanding of LTL (Linear Temporal Logic) verification, enabling them to define and check customized temporal properties. This advanced feature was presented in [8]. This paper proposes BProVe, a novel verification approach for BPMN collaborations. It combines both standard model checking techniques, through the MAUDE's LTL model checker, and statistical model checking techniques, through the statistical analyzer MultiVeStA [19,20]. The need for the inclusion of two different verification strategies emerged after observing the presence of collaborative models with large state space for which standard verification strategies were not able to provide an answer. The validation we performed aimed at assessing that the inclusion of two different strategies is indeed valuable, as they show complementary characteristics. It also permits the enlarging of the set of models for which the approach can provide an answer. This makes BProVe effective also in those scenarios suffering from the state-space explosion problem usually observed including message exchange. Indeed, when performing an analysis, MultiVeStA run a number of simulations, and provides as a result, the average of the evaluations computed for each simulation. To support the adoption of the BProVe approach, we propose a web-based tool chain that allows for BPMN modeling, verification, and result exploration. The feasibility of BProVe has been validated both on synthetically generated models and on models retrieved from two public repositories. The performed validation highlighted the importance and complementarity of the two supported verification strategies such as LTL and SMC. Figure 3 provides an overview of the BProVe interface.

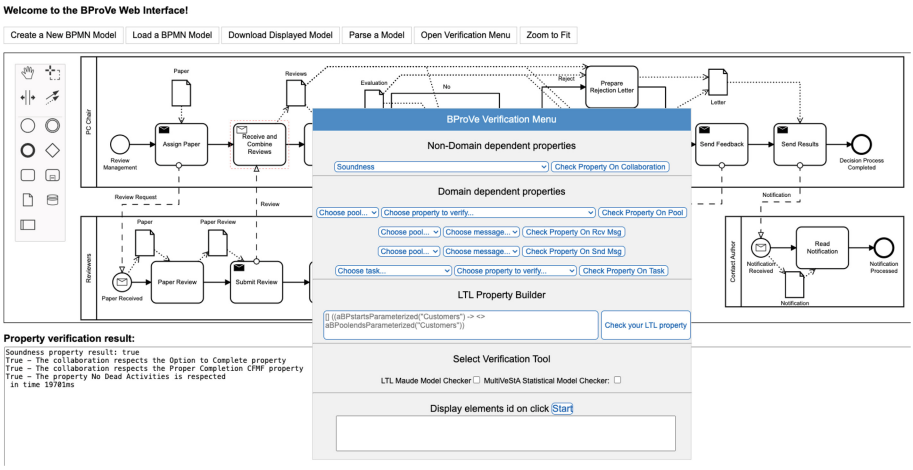

Fig. 3. Paper reviewing collaboration model - BProVe.

Paper Reviewing Collaboration Model (Running Example 2/3). Referring to our running example, by means of BPRoVe we can check if it fits with safety and soundness, and it returns that all the properties were satisfied. Please notice that the data objects, even if they are represented in the canvas, do not influence the verification in any way since they are not part of the formal semantics, and the tool overlooks them.

4 BPMN Collaboration Diagrams: Extended Framework

Our research journey continued by extending the elements included in our semantics. Even though BPMN is a widely accepted notation, only a limited effort has been expended in formalizing the interplay among control features, data handling, and message exchange in scenarios requiring multiple instances of interacting participants. Still maintaining a pragmatic approach, we decide to face the problem of providing formal semantics for BPMN collaborations, including elements dealing with multiple instances [10,12], i.e., multi-instance pools and sequential/parallel multi-instance tasks. Beyond defining a novel formalization, we also provide a BPMN collaboration animator tool, named MIDA, faithfully implementing the formal semantics [11]. The key characteristic of MIDA is the animation of collaboration models, see Fig. 4, which enables model animation by means of token flows. Thus, the user can model a BPMN collaboration with multiple instance tasks/pools and data manipulation. Specifically, each task can be bounded by a conditional expression and can manipulate data, and messages can contain data values. The animation permits to look at the model execution and enables the identification of modeling errors. However, like in software code debugging, the human user is still in charge of identifying and fixing bugs.

Paper Reviewing Collaboration Model (Running Example 3/3). Referring to our running example, using MIDA, we can check the presence of modeling errors by animating the collaboration model. Figure 4 shows the execution step in which the *PC Chair* is waiting for three reviews (the red token in the upper pool), and the three reviewer instances have just finished the review (the three gold tokens with different identifiers). Moreover, on the right side of the interface, there is a *data panel* where to check the current valorization of the data fields.

Assuming that one of the *Reviewers* does not send the *mark* assigned to the paper under review, the *PC Chair* cannot perform the *Review Evaluation*. This can be rendered in MIDA assigning a guard condition to the *Review Evaluation* which states that the marks of the reviews must be non-empty ($mark \neq null$). During the animation, when the *PC Chair* reaches the *Review Evaluation* task, MIDA triggers the error which is rendered with red color and an informative message, see Fig. 5.

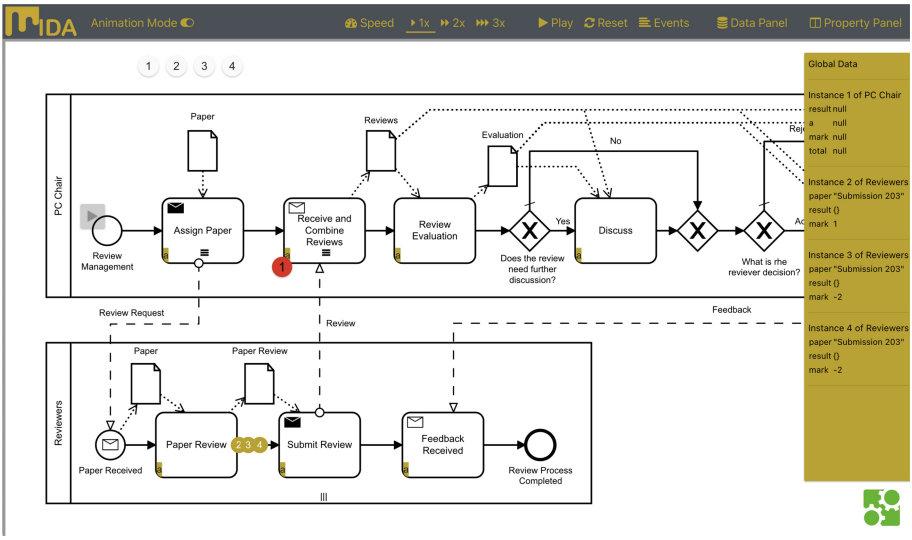

Fig. 4. Paper reviewing collaboration model - MIDA.

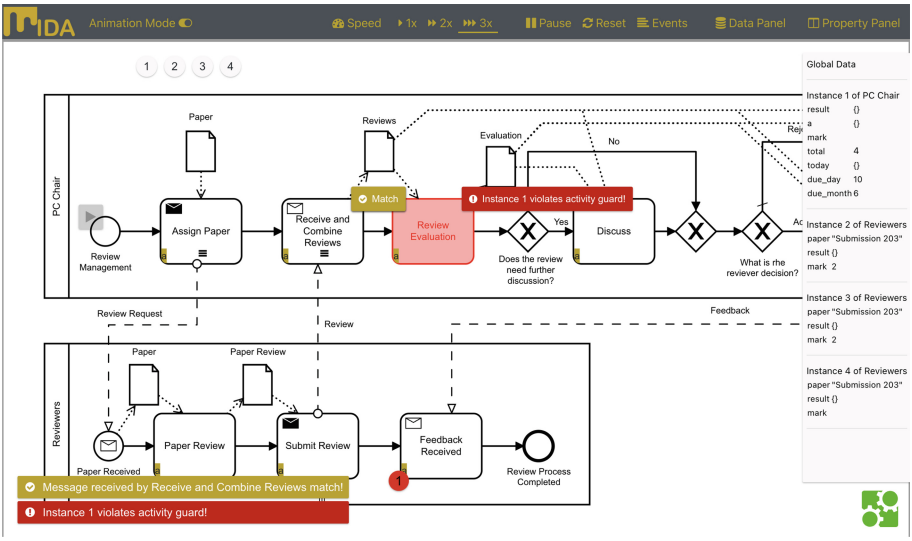

Fig. 5. Paper reviewing collaboration model - MIDA.

5 Concluding Remarks

This paper presents a comprehensive overview of our previous research on the formalization of business processes. Our contributions are based on direct formalizations of BPMN semantics, expressed through Labeled Transition Systems.

Operational semantics are provided for a significant subset of BPMN elements, highlighting the ability to model collaborations among organizations through message exchange involving multiple instances, data, and message. The outlined semantics establish a foundation for property verification, including safety and soundness. In cases where verification becomes challenging due to state explosion issues, designers are aided by the animation, which faithfully implements the formal semantics, in identifying and rectifying erroneous behaviors.

We express our gratitude to Rocco De Nicola for his guidance. The findings outlined in this paper, along with many others, are directly influenced by his contributions and works.

References

1. Caires, L., De Nicola, R., Pugliese, R., Vasconcelos, V.T., Zavattaro, G.: Core calculi for service-oriented computing. In: Wirsing, M., Hölzl, M. (eds.) Rigorous Software Engineering for Service-Oriented Systems. LNCS, vol. 6582, pp. 153–188. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20401-2_8
2. Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: A systematic literature review on iot-aware business process modeling views, requirements and notations. Softw. Syst. Model. **22**(3), 969–1004 (2023). https://doi.org/10.1007/S10270-022-01049-2
3. Compagnucci, I., Corradini, F., Fornari, F., Re, B.: Trends on the usage of BPMN 2.0 from publicly available repositories. In: Buchmann, R.A., Polini, A., Johansson, B., Karagiannis, D. (eds.) BIR 2021. LNBIP, vol. 430, pp. 84–99. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87205-2_6
4. Compagnucci, I., Corradini, F., Fornari, F., Re, B.: A study on the usage of the BPMN notation for designing process collaboration, choreography, and conversation models. Bus. Inf. Syst. Eng. **66**(1), 43–66 (2024). https://doi.org/10.1007/S12599-023-00818-7
5. Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: A formal approach to modeling and verification of business process collaborations. Sci. Comput. Program. **166**, 35–70 (2018). https://doi.org/10.1016/J.SCICO.2018.05.008
6. Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F., Vandin, A.: Bprove: a formal verification framework for business process models. In: Rosu, G., Penta, M.D., Nguyen, T.N. (eds.) Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2017, Urbana, IL, USA, 30 October–03 November 2017, pp. 217–228. IEEE Computer Society (2017). https://doi.org/10.1109/ASE.2017.8115635
7. Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F., Vandin, A.: Bprove: tool support for business process verification. In: Rosu, G., Penta, M.D., Nguyen, T.N. (eds.) Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, ASE 2017, Urbana, IL, USA, 30 October–03 November 2017, pp. 937–942. IEEE Computer Society (2017). https://doi.org/10.1109/ASE.2017.8115708
8. Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F., Vandin, A.: A formal approach for the analysis of BPMN collaboration models. J. Syst. Softw. **180**, 111007 (2021). https://doi.org/10.1016/J.JSS.2021.111007

9. Corradini, F., Morichetta, A., Muzi, C., Re, B., Tiezzi, F.: Well-structuredness, safeness and soundness: a formal classification of BPMN collaborations. J. Log. Algebraic Methods Program. **119**, 100630 (2021). https://doi.org/10.1016/J.JLAMP.2020.100630
10. Corradini, F., Muzi, C., Re, B., Rossi, L., Tiezzi, F.: Animating multiple instances in BPMN collaborations: from formal semantics to tool support. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 83–101. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_6
11. Corradini, F., Muzi, C., Re, B., Rossi, L., Tiezzi, F.: MIDA: multiple instances and data animator. In: van der Aalst, W.M.P., et al. (eds.) Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, 9–14 September 2018. CEUR Workshop Proceedings, vol. 2196, pp. 86–90. CEUR-WS.org (2018). https://ceur-ws.org/Vol-2196/BPM_2018_paper_18.pdf
12. Corradini, F., Muzi, C., Re, B., Rossi, L., Tiezzi, F.: BPMN 2.0 or-join semantics: global and local characterisation. Inf. Syst. **105**, 101934 (2022). https://doi.org/10.1016/J.IS.2021.101934
13. Corradini, F., Polini, A., Re, B., Tiezzi, F.: An operational semantics of BPMN collaboration. In: Braga, C., Ölveczky, P.C. (eds.) FACS 2015. LNCS, vol. 9539, pp. 161–180. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28934-2_9
14. Eker, S., Meseguer, J., Sridharanarayanan, A.: The maude ltl model checker. Electron. Notes Theor. Comput. Sci. **71**, 162–187 (2004)
15. El-Saber, N.A.S., Boronat, A.: BPMN formalization and verification using maude. In: Proceedings of the 2014 Workshop on Behaviour Modelling - Foundations and Applications, BM-FA 2014, York, United Kingdom, 22 July 2014, p. 1. ACM (2014). https://doi.org/10.1145/2630768.2630769
16. Gorp, P.V., Dijkman, R.M.: A visual token-based formalization of BPMN 2.0 based on in-place transformations. Inf. Softw. Technol. **55**(2), 365–394 (2013). https://doi.org/10.1016/J.INFSOF.2012.08.014
17. Kheldoun, A., Barkaoui, K., Ioualalen, M.: Specification and verification of complex business processes - a high-level petri net-based approach. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 55–71. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_4
18. OMG: Business Process Model and Notation (BPMN V 2.0) (2011)
19. Sebastio, S., Vandin, A.: MultiVeStA: statistical model checking for discrete event simulators. In: Proceedings of ValueTools 2013, pp. 310–315. ICST/ACM (2013)
20. Vandin, A., Giachini, D., Lamperti, F., Chiaromonte, F.: Automated and distributed statistical analysis of economic agent-based models. J. Econ. Dyn. Control **143**, 104458 (2022)