Chapter 2
State of the Art

This section provides an overview of related work concerning established approaches for business process modelling, formal representations of the Business Process Model and Notation (BPMN) and the suitability of BPMN for business process modelling regarding gaps and limitations.

2.1 An Overview of Business Process Modelling Techniques

Business process-related topics are an active field of research, one subset being the different languages and notations to describe business processes. There are many languages which are focused on the control flow of the business process, e.g. BPMN [95], Yet Another Workflow Language (YAWL) [1], Event-Driven Process Chains (EPCs) [29, 62, 122] or Petri nets [100]. In addition, there are object-oriented approaches, in particular Unified Modeling Language (UML) activity diagrams, statecharts and use case diagrams [96], where first the necessary objects are identified and then the dynamic behaviour is defined. Data flow-oriented methods, such as the Structured Analysis and Design Technique (SADT) [75] or IDEF [49], are increasingly rarely used in practice [50]. Of particular interest for future demands will be languages which follow a more resource-centric approach, as, for example, in the approach of Subject-Oriented Business Process Management (S-BPM) [45].

According to zur Mühlen et al. [84], modern business process modelling languages like BPMN or EPCs offer more constructs to represent real-world situations than their predecessors, e.g. IDEF or Petri nets. However, the "apparent increase in expressiveness is accompanied by an increase in language complexity" [84]. Besides some guidelines towards model quality to cope with complexity, which exist in practice [74, 126, 127] and research [13, 68, 72], Mendling et al. propose a set of seven process modelling guidelines (7PMG), which address "the mismatch

F. Kossak et al., *A Rigorous Semantics for BPMN 2.0 Process Diagrams*,
DOI 10.1007/978-3-319-09931-6_2

between abstract recommendations for process modeling and technical insights into modeling practice" [76].

Selecting notations for modelling process-oriented applications is a frequently discussed topic. Available languages for conceptual business process modelling (see, e.g. [50, 73, 114, 141]) differ in their extent of modelling elements, as well as in the source domains and application areas targeted.

BPMN [95] targets both business analysts and software architects to collaboratively design, deploy and monitor business processes. It enables analysts to freely design the processes and developers to add necessary technical details afterwards. Due to its maintenance by the Object Management Group (OMG) and its recent adoption as an ISO standard (ISO 19510:2013), BPMN also meets the requirement to use a generally accepted notation, which guarantees certain sustainability. Although BPMN offers a wide range of modelling elements, it also defines a basic set of core elements, which simplifies the modelling and understanding of complex business processes. In order to provide further guidance on how to reduce language complexity, zur Mühlen and Recker [83] investigate subsets of BPMN that are frequently used and agreed in practice (see also Sect. 2.3).

The use of BPMN, however, does not eliminate the need for a system development language, such as UML [46]. The UML standard [96] is also controlled by the OMG and constitutes a relatively open standard, which is strongly aligned with the needs of software architects and developers supporting technical processes in terms of object-oriented programming. For business process modelling, primarily UML activity diagrams are used.

YAWL [1], an open-source workflow language, is based on a rigorous analysis of existing workflow languages. It was developed as an academic standard and provides direct support for all Workflow Patterns [117–119]. Hence, YAWL supports the control flow, data and resource perspective and extends its basic Petri nets with additional features. However, YAWL also puts its main emphasis on modelling the process model and concentrates less on the informational and organisational models (cf. also [7]).

An EPC [29, 62, 122] represents a business process as a chronological, logical sequence of activities, which are triggered by events and whose results are events as well. The concept enables the integration of various views (functional, data, organisational and resource view). Compared to other modelling approaches, e.g. BPMN or UML, EPCs lack standardisation by an appropriate organisation, strongly restricting their use and dissemination.

Petri nets [100], in particular high-level Petri nets and further enhancements, are suited for modelling business processes as well. They graphically depict the structure of a business process as a bipartite graph which is composed of place nodes, transition nodes and directed arcs connecting places with transitions. Additionally, they describe not only structure but also execution of the business process by their operational semantics. The use of Petri nets in Business Process Management (BPM) tools, however, often takes place unbeknown to the user due to their rapidly increasing complexity. For instance, a business process can be modelled as a UML activity diagram but is internally mapped on a Petri net, which is then used by the tool for analysis and execution [144].

2.2 Formal Representations of BPMN

An important aspect before utilising BPM in general, and BPMN in particular, is to define strict syntax and semantics of the business process models and to formally verify them [82].

As of 2008, according to Recker [104], BPMN was already used in over thirty countries, and it seems that the influence of BPMN will further increase. However, BPMN is repeatedly criticised for its lack of formal semantics, e.g. concerning the OR-join [21]. Thus, several research projects have investigated approaches to formally define BPMN, e.g. based on Petri nets, Communicating Sequential Processes (CSP) or Abstract State Machines (ASMs).

This subsection presents several approaches that support a formal or textual representation of BPMN. Although there are several publications describing the semantics of BPMN (or at least of parts thereof), they either rely on previous versions of the BPMN standard or they do not sufficiently go into detail concerning the whole range of BPMN elements in order to provide answers to our questions.

First of all, the BPMN standard [95] itself provides a metamodel of all BPMN elements as a UML class diagram and in the form of an XML schema. BPMN 2.0 is the first release to provide such a formal definition. Consequently, BPMN diagrams can be textually represented in XML, which is used by several modelling tools, e.g. the Eclipse BPMN Modeler.

Another textual notation for BPMN (and also for the UML activity diagram) is called TN4PM and is proposed by Mogos and Urzica in [80]. The notation is based on entity blocks. Furthermore, Urzica and Tanase [138] provide a textual notation for BPMN based on the Backus–Naur Form (BNF).

The syntax of BPMN can also formally be defined within an ontology based on description logic. The S-BPMN [3] and BPMNO [52] ontologies are both based on former releases of BPMN, and their classes are mainly defined for concrete BPMN elements. We have defined an ontology based on BPMN 2.0 in [86], which comprises the syntactical definitions from the BPMN metamodel and the natural text of the specification. The ontology can be used as a knowledge base to investigate the BPMN elements in a very effective way and for syntax checking to validate concrete BPMN models.

According to Dijkman et al. [32], the static analysis of BPMN models is complicated by the complexity of the language, since BPMN integrates constructs from graph-oriented process definition languages with features for the concurrent execution of multiple instances. The lack of formal semantics of BPMN hinders the development of tool support for checking the correctness of BPMN models from a semantic perspective. Therefore, the approach by Dijkman et al. introduces the formal semantics of BPMN defined in terms of a mapping to Petri nets in order to be able to statically analyse the business process models and check their semantic correctness [32]. The approach deals with a comprehensive subset of BPMN 1.0.

The proposed mapping from BPMN to Petri nets, however, lacks features which coincide with the limitation of Petri nets that in turn motivated the design of YAWL [1]. YAWL extends Petri nets with a number of high-level features to facilitate the modelling of complex workflows. However, modelling with Petri nets soon becomes very complex, as can be seen, for example, in [133], where a relatively simple transaction in a travel agency's business process leads to a very complex Petri net representation.

Cervantes [27] also proposes an approach to formalise the semantics of BPMN by using a mapping to Petri nets, however, also not including mappings for several elements, e.g. for instantiating event-based gateways.

Further mappings exist for (subsets of) previous versions of the BPMN standard, e.g. a mapping to YAWL [154] or a mapping to the Business Process Execution Language (BPEL) [97], whereby a graph structure is transformed into a block structure.

Weidlich et al. [140] present the other perspective of the latter alignment, i.e. a BPEL-to-BPMN mapping and its pitfalls. For example, they show that the *pick* activity, with the attribute *createInstance* set to "no", can be directly mapped to its counterpart in BPMN, the event-based gateway. However, they indicate compatibility issues concerning the process instantiation mechanisms of BPEL and BPMN, i.e. BPEL scenarios involving multiple start activities are only partially mappable to instantiating event-based-gateways.

Nicolae et al. [93] use high-level modelling of languages based on UML to provide a common understanding in terms of an abstract syntax of the involved concepts of *Service Interaction Patterns* that are directly supported by BPMN.

A semantic foundation for BPMN which is based on the Calculus for Orchestration of Web Services (COWS) is presented in [102]. This approach enables to derive a COWS specification from XML representations provided by modelling applications.

A formal process semantics for a subset of the BPMN standard is also provided in terms of the process algebra CSP by Wong and Gibbons in [150]. The approach presents an abstract syntax for BPMN based on the Z notation [131] and the behavioural semantics in CSP, whose expressiveness is strictly more than that of BPMN. Such semantics allow domain experts and developers to formally analyse and compare BPMN diagrams. Moreover, the process semantics in CSP can also be applied to reasoning as well as to the refinement of BPMN diagrams. Wong [149] also presents an untimed process semantics as well as a relative timed semantics for BPMN 1.0 in CSP.

Additionally, an analysis of the former BPMN 1.0 standard is presented by Wohed et al. in [147] to identify inconsistencies and limitations, whereby they also consider the Workflow Patterns [117–119]. A formalisation in terms of graph rewrite rules for a subset of the execution semantics of BPMN 2.0 is given by Dijkman and Gorp in [33]. Yet another approach by Zahoor [156] uses the event calculus to formalise BPMN.

Finally, a further interesting approach defines the semantics of BPMN in terms of ASMs (compare Börger and Sörensen [20] for a formalisation of BPMN 1.1 and Börger and Thalheim [23, 24] for a formalisation of BPMN 2.0 (beta)), where the formalisation is kept relatively abstract. We refine their approaches in order to cope with adaptions and extensions to the previously published versions of BPMN, e.g. regarding sub-processes and events, and specify communication and messaging details by enabling multiple process instances to run in parallel. ASMs will be briefly explained in Chap. 3, where we will also introduce the ASM notation used in this book.

2.3 Suitability of BPMN for Business Process Modelling

Besides the formal specification of syntax and semantics of BPMN, several other issues regarding the suitability of BPMN for business process modelling have been identified by related work.

For example, Recker et al. [105] present an evaluation of BPMN based on the Bunge–Wand–Weber (BWW) ontology and on interviews. The ontological evaluation reveals construct deficits (e.g. state, history and system structure), construct redundancies (e.g. pool and lane concept, transformations and events), construct excess (e.g. off-page connectors and groups) and construct overload (e.g. pool and lane concept). Furthermore, considering the BPMN elements activity, task, collapsed sub-process, expanded sub-process, nested sub-process and transaction, Recker et al. do not see significant semantic differentiation in terms of their use. Beneath other issues, the problem of unnecessary elements as well as the pool and lane concept were voted in the highest problem category during the conducted interview [105].

In further publications, Recker et al. studied the drawbacks of business process modelling in general and BPMN in particular [104, 106]. In [106], they identified several issues concerning decomposition and reusability, such as limited support for process decomposition and lack of integration of business rule specification. In [104], Recker presents a global survey, involving 590 BPMN users. The survey is based on BPMN 1.0 but nevertheless provides interesting information about the usage of BPMN elements. For example, 36% of the respondents only use a core set of BPMN symbols, 37% use an extended set and 27% use all the functionality BPMN 1.0 has to offer. In addition, five major drawbacks of BPMN were identified in the survey, including limited support for business rule specification, process decomposition, organisational modelling, the extent of unnecessary elements and the large number of event types.

Subsequently, Recker [108] also describes a number of problems related to the practical usage of BPMN and stresses the need for more insights into the practice of process modelling for future development.

Concerning the unnecessary elements of the BPMN standard, a more recent analysis is provided by Kunze et al. in [70]. The authors studied the use of elements in 1210 BPMN models. The most popular elements, which are used in more than 50% of the BPMN models, are sequence flows, tasks, start/end events and the pool and lane concept. However, more than 20 elements are used in less than 10% of the BPMN models, e.g. the intermediate error event, the inclusive gateway or the event sub-process (not used at all).

One of the major challenges of modelling process-oriented applications is to capture all different aspects and their interrelationships in the context of business processes. For example, in the area of enterprise architecture frameworks, a number of well-established methodologies like ARIS or the Zachman Framework comprise several views, primarily including organisational and data models in addition to a central process model.

ARIS, for example, consists of five views that are symbolically presented in the form of a house, the so-called ARIS house, with the Organisation View as the roof; the Data View, Control View and Function View as the three pillars; and the Output View as the basis of the house [121, 122]. According to [121], the designations "function", "process" and "activity" are used synonymously; thus, the Function View is similar to the BPMN process diagrams, which primarily express the flow of activities.

In addition, the Zachman Framework [79, 155] consists of six rows for the viewpoints (Scope, Enterprise (or Business) Model, System Model, Technology Model, Detailed Representations and Functioning Enterprise) and six columns for the aspects (Data, Function, Network, People, Time and Motivation). The aspect "Function" contains the business process model and is similar to the control flow-oriented BPMN process diagrams.

The workflow modelling language YAWL supports three different perspectives, i.e. control flow, data and resources [4], to facilitate the modelling of complex workflows (see also Sect. 2.1).

Whereas the functional and behavioural aspects (activities, gateways and events) are well represented in BPMN [73], the organisational and informational aspects are only partly supported. Wohed et al. [145] also study the suitability of BPMN for business process modelling and thereby use the Workflow Patterns as an evaluation framework. According to this publication, BPMN provides good support for the control flow perspective, medium support for the data perspective but only poor support for the resource perspective.

Focusing on business processes, BPMN naturally includes a role concept, but it does not distinguish between the different types of process participants. Swimlanes are used to display external and internal participants, which can be humans, organisational roles or units as well as software components like services or applications. Throughout the intuitive modelling of activities by pools and lanes, respectively, basic patterns such as *direct distribution* or *role-based distribution* can be realised straightforward, whereas not all advanced patterns are supported by BPMN. In detail, considering the resource perspective, BPMN supports only 8 out of the 43 Workflow Resource Patterns [117], since swimlanes are specified in a restrictive manner [145]. Thus, it is important that the resource perspective is more widely acknowledged as an integral part of business process modelling. In addition, although the BPMN standard states that organisational modelling and resources are out of the scope of BPMN, the pool and lane concepts reveal the need for these concepts.

According to the Workflow Data Patterns [119], the visibility of data is realised through properties of a task, a sub-process or a process. Interaction issues are supported through the notion of data objects associated to sequence flows or parameter passing between sub-processes. Data transfer is supported via message flows, and data routing is handled by data objects and miscellaneous event types. However, the data used in business processes, especially at a higher level of abstraction, often is only roughly specified; there is no means of specifying details of data in BPMN, such as the concrete description of attributes and data types, or the relations between data entities, as we are accustomed to do in, e.g. entity-relationship diagrams.

Considering these characteristics, BPMN (in part) offers semantics for structuring process elements, to specify when and in which order they are performed, who is responsible for them and which informational entities are created or manipulated during the process. What is missing, however, is a clear interface for integrated modelling of user interaction. Like many current business process modelling approaches, BPMN is not integrated with the user interaction—the dialogues of an enterprise application—and does not offer appropriate integration of the data model [9].

To address the gaps and limitations of BPMN regarding business process modelling, several extensions to BPMN have already been suggested. For example, Awad et al. [12] propose a task-based human resource allocation and extended the BPMN metamodel with the Object Constraint Language (OCL) to express resource allocation constraints. Task-based authorisation constraints for BPMN supporting different patterns like *separation of duties*, *role-based allocation* or *case handling* are further presented by Wolter and Schaad in [148]. In addition, a BPMN extension considering security requirements like access control, non-repudiation, privacy or integrity is proposed by Rodríguez et al. [113]. Furthermore, Korherr and List [65] extend BPMN with goals and performance measures, and Milanovic et al. [78] provide a rule-based extension of the BPMN metamodel based on the REWERSE Rule Markup Language (R2ML).

Auer et al. [9, 10] present an extension to BPMN for describing user interaction following the submit/response-style interaction paradigm, which is characteristic of form-based applications ranging from small Web applications to large Enterprise Resource Planning (ERP) systems. They use UML for the data model defining both the information and the message model.

An alternative approach is proposed by Trætteberg and Krogstie in [136, 137] based on BPMN and Diamodl, a data flow-oriented visual modelling language for the logic and behaviour of a user interface. The basic idea of this work is to augment BPMN to cover tasks by adding information concerning the object lifecycle. To overcome the weakness of BPMN in domain and data modelling, the authors integrate XML schemas and extend BPMN with annotations in order to define pre- and postconditions within the task model.

As in [136, 137], Dividino et al. [34] use BPMN and Diamodl to propose a model-driven approach for integrating business process and user interface models. The authors use the ISE methodology [26], which is based on the Zachman Framework and combines the business approach with the concepts of Model-Driven Engineering (MDE) as integration platform. They identify both vertical and horizontal dependencies for the synchronisation of models on different layers of abstraction in one dimension and on the same layer, respectively. To maintain the consistency and integrity of the models, they extend the modelling languages with new event-based components and implement QVT-based model transformation rules to achieve an event-based synchronisation between the business process and user interface models.

So, summing up, we identified several business process modelling techniques; however, all of them focus on different aspects of business process modelling and, therefore, have their respective shortcomings. From our perspective, BPMN has (and will have) the greatest impact in industrial practice, and thus, it is of great value to discuss it based on a rigorous semantics. There is a diversity of formal representations of BPMN, but they either are based on previous versions of BPMN, only address subsets of the BPMN specification or are not detailed enough. In addition, most attempts on formalisation easily become difficult to understand even for experts. For those reasons, we decided to apply ASMs to rigorously define the semantics of BPMN process diagrams. Besides the lack of formal rigour, we also found several open issues, which are largely due to semantic ambiguities or even gaps in the specification, such as unnecessary elements as well as insufficient support for organisational, data and interaction modelling.

http://www.springer.com/978-3-319-09930-9

A Rigorous Semantics for BPMN 2.0 Process Diagrams

Kossak, F.; Illibauer, C.; Geist, V.; Kubovy, J.;
Natschläger, C.; Ziebermayr, Th.; Kopetzky, T.;
Freudenthaler, B.; Schewe, K.-D.

2014, X, 235 p. 80 illus., 1 illus. in color., Hardcover
ISBN: 978-3-319-09930-9