Analysis and Design of Business Processes Using BPMN

Gustav Aagesen and John Krogstie

Abstract In 2004, the Business Process Modeling Notation (BPMN) was presented as a standard business process modeling language. Its development was considered to be an important step in reducing the fragmentation that was witnessed between the existing process modeling tools and notations. Since then BPMN has been evaluated in different ways by the academic community and has become widely supported by the industry. After completing the first major revisions of BPMN, the Object Management Group (OMG) is working toward a new BPMN standard, BPMN 2.0. This chapter summarizes some of the evaluations of BPMN and presents these together with reported experiences as well as some examples of proposed extensions and future expectations based on these.

1 Introduction

Models of business and work processes have for a long time been utilized to learn about, guide, and support practice in a number of areas. In software process improvement (Derniame 1998), enterprise modeling (Fox and Gruninger 2000), active knowledge modeling (Lillehagen and Krogstie 2008), and quality management, process models describe methods and working procedures. Simulation and quantitative analyses are also performed to improve efficiency (Kuntz et al. 1998) (van der Aalst et al. 2010). In process-centric software engineering environments (Ambriola et al. 1997) and workflow systems (WfMC 2000), model execution is automated. Thus, process modeling is not done for one specific objective only, which partly explains the great diversity of approaches found in literature and

G. Aagesen (✉)
Norwegian University of Science and Technology, Trondheim, Norway
e-mail: gustav.aagesen@idi.ntnu.no

J. vom Brocke and M. Rosemann (eds.), *Handbook on Business Process Management 1*,
International Handbooks on Information Systems,
DOI 10.1007/978-3-642-00416-2_10, © Springer-Verlag Berlin Heidelberg 2010

practice. Five main categories of usage of process modeling can be distinguished (Krogstie et al. 2008):

1. Human sense-making and communication to make sense of aspects of an enterprise and to support communication between different stakeholders. Sense-making models are used within an activity to make sense of something in an ad hoc manner, and will usually not be maintained afterwards.
2. Computer-assisted analysis to gain knowledge about the enterprise through simulation or deduction based on the contents of the model.
3. Quality management, following up the adherence of the work process to standards and regulations. Here the model is meant to act as part of a corporate memory meant to exist as a reference point over time and as input to and basis for process improvement.
4. Model deployment and activation to integrate the model in an information system. Deployment can be manual, automatic (in automated workflow systems), or interactive (Krogstie and Jørgensen 2004).
5. Using the model as a context for a system development project, without being directly implemented (as it is in category 4).

Business Process Management (BPM) is a structured, coherent, and consistent way of understanding, documenting, modeling, analyzing, simulating, executing, and continuously changing end-to-end business process and all involved resources in light of their contribution to business performance (Recker et al. 2006). We see that the potential usage of modeling in BPM covers all the areas of use for process modeling in general as outlined above.

Traditionally, a wide variety of approaches and notations have been used for BPM and workflow. Inspired by a number of previous languages, BPMN has over the last years been promoted and suggested as a standard and has been met with the same kind of diverse needs; i.e., to create models to be understandable both for humans and machines, for sense-making, quality management, simulation, and execution. The main approach for execution is the mapping of BPMN models to BPEL.

This chapter aims to identify and report on the main efforts to evaluate BPMN, both analytical and empirical, and by this providing a current state of the art on this area.

The following section will introduce BPMN and the remaining sections will focus on the evaluation of the language. We will introduce the methods used in evaluating BPMN briefly. The trends of the outcome of the evaluations will be presented. Some of the proposed extensions of BPMN will then be described.

2 Business Process Modeling and BPMN

The wide range of applications of process modeling described in the introduction is reflected in current modeling notations, which emphasize different aspects of work. Ten years ago, Carlsen (1998) identified five categories of process modeling languages: transformational, conversational (speech-act-based), role-oriented, constraint-based, and systemic. The increased interest in modeling processes with UML indicates that object-oriented process modeling can be looked upon as a sixth category. On the other hand, most process modeling languages take a transformational approach (input–process–output). Processes are divided into activities, which may be divided further into subactivities. Each activity takes inputs, which it transforms to outputs. Input and output relations thus define the sequence of work. This perspective is chosen for the standards of the Workflow Management Coalition (WfMC 2000), the Internet Engineering Task Force (IETF) (Bolcer and Kaiser 1999), and the Object Management Group (OMG 2000) as well as most commercial systems for the last 10–15 years (Abbot and Sarin 1994; Fischer 2000). IDEF (1993), Data Flow Diagram (Gane and Sarson 1979), Activity Diagrams (Booch et al. 2005), Event-driven Process Chains (Scheer 2000), BPMN (BPMI.org and OMG 2008) and Petri nets (van der Aalst et al. 2000) are well-known transformational languages. We focus here on this type of process modeling, with the emphasis on BPMN.

2.1 BPMN

In 2004, the Business Process Modeling Notation (BPMN) was presented as the standard business process modeling notation (White 2004). Since then BPMN has been evaluated in different ways by the academic community and has become widely supported in industry.

There are currently 50 current and 4 planned implementation of (BPMN).[1] The tool support in industry has increased with the awareness of the potential benefits of BPM. Analytical evaluations showing weaknesses in BPMN have been available for some time, but the first reports on the experiences and perceived use of BPMN have however been published just recently.

The Business Process Modeling Notation (BPMN version 1.0) was proposed in May 2004 and adopted by OMG for ratification in February 2006. The current version is BPMN 1.1 (OMG 2008) and the following version BPMN 2.0 is in development. BPMN is based on the revision of other notations and methodologies, especially UML Activity Diagram, UML EDOC Business Process, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM, and Event-driven Process Chains.

The primary goal of BPMN was to provide a notation that is readily understandable by all business users, from the business analysts who create the initial draft of the processes, to the technical developers responsible for implementing the technology that will support the performance of those processes, and, finally, to the business people who will manage and monitor those processes (White 2004).

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

Another factor that drove the development of BPMN is that, historically, business process models developed by business people have been technically separated from the process representations required by systems designed to implement and execute those processes. Thus, it was a need to manually translate the original process models to execution models. Such translations are subject to errors and make it difficult for the process owners to understand the evolution and the performance of the processes they have developed. To address this, a key goal in the development of BPMN was to create a bridge from notation to execution languages. As indicated above BPMN models can be activated through the mapping to BPEL.

BPMN allows the creation of end-to-end business processes and is designed to cover many types of modeling tasks constrained to business processes. The structuring elements of BPMN will allow the viewer to be able to differentiate between sections of a BPMN Diagram using groups, pools, or lanes. Basic types of submodels found within a BPMN model can be *private business processes* (internal), *abstract processes* (public), and *collaboration processes* (global).

Private business processes are those internal to a specific organization and are the types of processes that have been generally called workflow or BPM processes.

Abstract processes represent the interactions between a private business process and another process or participant. Abstract processes are contained within a Pool and can be modeled separately or within a larger BPMN Diagram to show the Message Flow between the abstract process activities and other entities.

Collaboration processes depict the interactions between two or more business entities. These interactions are defined as a sequence of activities that represent the message exchange patterns between the entities involved.

2.1.1 Language Constructs and Properties

The Business Process Diagram is the graphical representation of the BPMN. Its language constructs are grouped in four basic categories of elements, viz., Flow Objects, Connecting Objects, Swimlanes, and Artifacts. The notation is further divided into a core element set and an extended element set. The intention of the core element set is to support the requirements of simple notations and most business processes should be modeled adequately with the core set. The extended set provides additional graphical notations for the modeling of more complex processes.

Flow objects (Fig. 1) contain events, activities, and gateways. *Events* are either start events, intermediate events, or end events. *Activities* are divided into process, subprocess, and tasks and denote the work that is done within a company. *Gateways* are used for determining branching, forking, merging, or joining of paths within the process. Markers can be placed within the gateway to indicate behavior of the given construct.

Connecting objects (Fig. 2) are used for connecting the flow objects. *Sequence Flow* defines the execution order of the activities within a process while *Message*

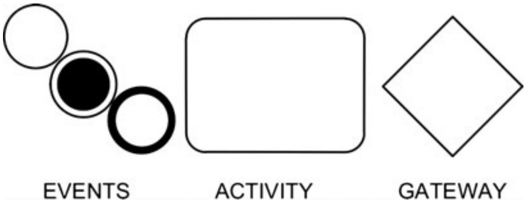

Fig. 1 BPD elements events (start, intermediate, and end), activity, and gateway

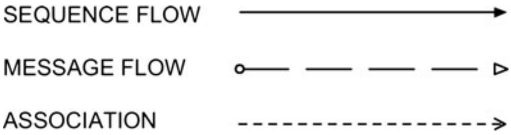

Fig. 2 BPD connection objects: Sequence flow, message flow, and association

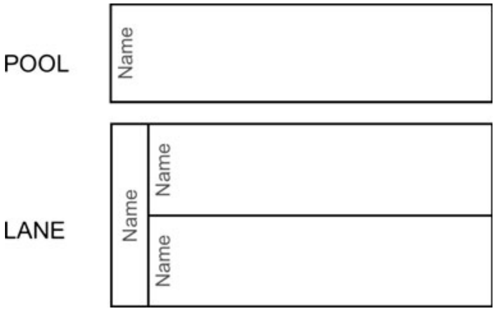

Fig. 3 BPD pool and lanes

Flow indicates a flow of messages between business entities or roles prepared to send and receive them. *Association* is used to associate both text and graphical nonflow objects with flow objects.

Swimlanes (Fig. 3) are used to denote a participant in a process and acts as a graphical container for a set of activities taken on by that participant. By dividing *Pools* into *Lanes* (thus creating subpartitioning), activities can be organized and categorized.

Artifacts (not illustrated) are data objects, groups, and annotations. *Data Objects* are not considered as having any other effect on the process than information on resources required or produced by activities. The *Group* construct is a visual aid used for documentation or analysis purposes while the *Text Annotation* is used to add additional information about certain aspects of the model.

Figure 4 shows an example BPMN process summoning participants for a workshop. The workshop organizer sends out the invitations, which are received by the potential participants. The participants evaluate the relevance of the workshop and decide whether they will participate or not. Those who want to participate, sign up for the workshop by informing the organizer.

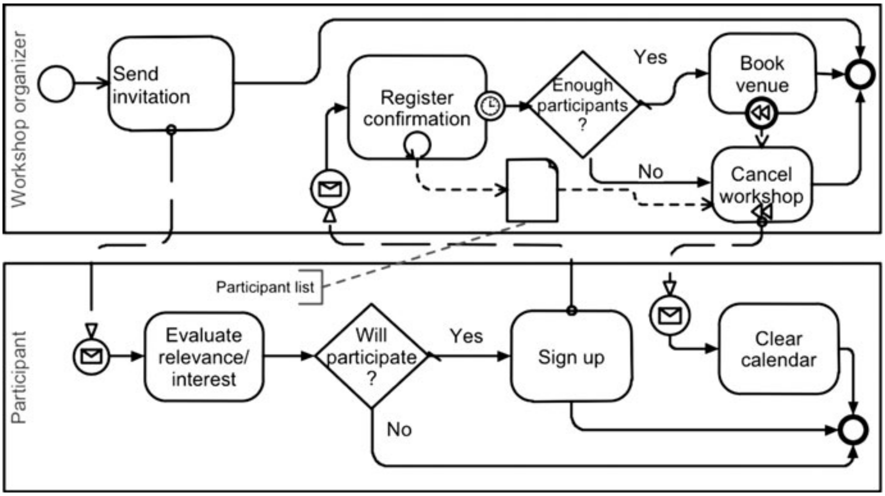

Fig. 4 BPMN model showing the summons for a workshop

The organizer registers the confirmations from the participants until the deadline for registering, making a list of participants. When the deadline is reached (indicated by the timer event on the looping register confirmation activity), the organizer will see if there are enough participants to conduct the workshop. If there are too few participants, the organizer will inform those participants who signed up that the workshop is canceled, and the registered participants will clear their calendar for the day.

If there are sufficient participants registered for the workshop, the organizer will try to book a venue. But if there is no venue available, the workshop will have to be canceled by informing registered participants. This is shown using the compensation and undo activity.

3 Evaluations of BPMN

The importance of evaluating available methods for modeling increases as the amount of available methods grow, since the results will guide the users in selecting the most fit method for the task at hand. Traditionally the research community has focused on creating new modeling languages rather than evaluating those that already exist (Wahl and Sindre 2005).

By evaluating existing methods one will not only be able to compare their suitability for solving the problem at hand, but it will also help determine the skills required of the user and model audience, before taking on the modeling task. By using formalized frameworks in the assessment of newly arrived methods and comparing the data with results from earlier studies it would be possible to determine whether the overall rating of the new method is higher than its predecessors.

Different approaches to evaluating modeling languages include analytical and empirical methods, and both single-language and comparative evaluations exist. Empirical methods should investigate both the possibility for modelers to use the language, comprehension of models developed in the language, and the ability to learn from and act according to the knowledge provided in the models (Gemino and Wand 2003; Krogstie et al. 2006). While analytical evaluations can be conducted as soon as the specification of the language is made available, empirical evaluations would in most cases require the users of the new method to have some experience with its use, and for that the method would need some time with the user community before evaluations can take place. Empirical studies might involve the investigation of whether the results from the analytical studies are supported and to what extent they have impact in practice. It would also involve performing case studies and surveys to discover if the method is as appropriate as expected and if it is used according to expectation.

BPMN is no longer considered to be new and it has been evaluated both analytically and empirically. The following section introduces the evaluation approaches followed by their outcomes. The evaluation results will be summarized in Sect. 4. For details about the evaluations please refer to their original reporting.

3.1 Ontological Analysis Using the Bunge–Wand–Weber Framework

As computerized information systems are representations of real-world systems, Wand and Weber suggest that a theory of representation based on philosophical *ontology* can be used to help define and build information systems that contain the necessary representations of real-world constructs including their properties and interactions (Rosemann et al. 2006). The Bunge–Wand–Weber framework defines a set of models based on an ontology defined by Bunge in 1977 (Wand and Weber 1993; Recker et al. 2006). The BWW *representation model* is one of these models, and it is suggested that it can be used to analyze a particular modeling technique so as to make predictions on the modeling strengths and weaknesses of the technique, in particular its capabilities to provide *complete* and *clear* description of the domain being modeled. The current key constructs of the BWW model can be grouped into the following clusters: things including properties and types of things; states assumed by things; events and transformations occurring on things; and systems structured around things.

Two main evaluation criteria may be studied according to *Ontological Completeness* and *Ontological Clarity*.

Ontological Completeness is decided by the degree of *construct deficit*, indicating to what level the modeling language maps to the constructs of the BWW representation model.

Ontological Clarity is decided by *construct overload*, where the modeling language constructs represent several BWW constructs, *construct redundancy*, where one BWW construct can be expressed by several language constructs and *construct excess*, having language constructs not represented in the BWW model.

Three reasons for selecting the BWW framework for evaluating BPMN is stated by Recker et al. (2005): It has, unlike other ontologies, been derived with the Information Systems discipline in mind. It is an upper ontology, with a comprehensive scope that allows wide applicability. Further, there is an established track record and demonstrated usefulness of ontological analyses of modeling techniques using BWW.

BWW based evaluations are presented in Recker et al. (2005), Rosemann et al. (2006), and Recker et al. (2007) and their findings include:

Representation of state. The BPMN specification provides a relatively high degree of ontological completeness (Rosemann et al. 2006), but BPMN is not ontologically complete. For example, states assumed by things cannot be modeled with the BPMN notation. This situation can result in a lack of focus in terms of state and transformation law foundations for capturing business rules.

System structure. Systems structured around things are under-represented, and as a result of this problems will arise when information needs to be obtained about the dependencies within a modeled system.

Representational capabilities compared with other approaches. A representational analysis was done in Rosemann et al. (2006) on different approaches that show that BPMN appears to be quite mature in terms of representation capabilities. This can perhaps be partly explained by the fact that the previous approaches like EPC and Petri nets influence the development of BPMN. It is interesting that only BPMN of the process modeling notations is able to cover all aspects of things, including properties and types of things. From this it is possible to note that BPMN appears to denote a considerable improvement compared with other techniques. The combination of ebXML and BPMN would provide maximum ontological completeness (MOC) with minimum ontological overlap (MOO) (Recker et al. 2005).

3.2 The Workflow Patterns Framework

The Workflow Patterns Framework[2] (van der Aalst et al. 2003; Russell et al. 2006) provides a taxonomy of generic, recurring concepts, and constructs relevant in the context of process-aware information systems (Wohed et al. 2005) (see also Ouyang et al. 2010).

The workflow patterns describe a core of foundational structures that one could expect workflow systems to support. Defining these patterns made it possible to

[2] http://www.workflowpatterns.com.

compare the expressive power of available commercial tools for business process modeling. Later, the patterns have been found applicable in a much broader sense and they have been used to examine the capabilities of business process modeling languages such as BPMN, UML Activity Diagrams, and EPCs; web service composition languages such as WCSI; and business process execution languages such as BPML, XPDL, and BPEL (Russell et al. 2006).

The available patterns are divided into the *control-flow perspective*, the *data perspective*, and the *resource perspective*. The original patterns were comprised of 20, 40, and 43 patterns, respectively. A revision of the control-flow patterns conducted in 2006 resulted in additional 23 patterns.

Three reasons for selecting the Workflow Patterns Framework are stated by Recker et al. (2007). It is a well accepted framework that has been widely used both for the selection of workflow management systems as well as vendor's self-evaluations of process modeling products; The framework has proven impact in the industry and it has triggering extensions to process modeling systems and inspired their development.

Workflow pattern-based evaluations are presented in Recker et al. (2007) and Wohed et al. (2005, 2006). The outcomes of the evaluations include:

Representation of state. Due to the lack of representation of state in BPMN there are difficulties in representing certain control-flow patterns (Wohed et al. 2006). There are further inherent difficulties in applying the Workflow Patterns Framework for assessing a language that does not have a commonly agreed-upon formal semantic or an execution environment. The BPEL mapping provided in the BPMN specification is only partial, leaving aside models with unstructured topologies. There are several ambiguities that can be found in the BPMN specification due to the lack of formalization (Wohed et al. 2006).

Multiple representations of the same pattern. The simple workflow patterns have multiple BPMN representations while capturing the most advanced patterns required deep knowledge of the attributes associated to BPMN's modeling constructs that do not have a graphical representation.

Support for instances. Workflow and environment data patterns are not supported due to the lack of support for instance-specific data for a task or subprocess with a "multiple instance" marker cannot be specified.

Resource modeling. Support for the resource perspective in BPMN is minimal, but the modeling of organizational structures and resources is regarded to be outside the scope of BPMN. The authors state that the lane and pool constructs are in contradiction to this.

3.3 SEQUAL

SEQUAL (Semiotic Quality Framework) (Krogstie et al. 2006; Lillehagen and Krogstie 2008) is used for evaluating different quality aspects of models, and for evaluating the potential of the language to build models having high quality based on the appropriateness of the domain in which the language is applied. The framework is based on linguistic and semiotic concepts (Reijers et al. 2010).

The dimensions in which model quality is determined are as follows. *Physical quality*: The basic quality goal is that the model is available for the audience. This includes aspects related to digital distribution and file formats. *Empirical quality* deals with predictable error frequencies when a model is read or written by different users, coding (e.g., shapes of boxes) and HCI-ergonomics for documentation and modeling-tools. *Syntactic quality* is the correspondence between the model and the modeling language extension. *Semantic quality* is the correspondence between the model and the domain, including validity and completeness. *Perceived semantic quality* is the similar correspondence between the audience interpretation of a mode l and his or hers current knowledge of the domain. *Pragmatic quality* is the correspondence between the model and the audience's interpretation and application of it. SEQUAL differentiates between social pragmatic quality (to what extent people understand and are able to use the models) and technical pragmatic quality (to what extent tools can be made that interpret the models, e.g., for execution purposes). Pragmatic quality also includes in what extent the participants and audience after interpreting the model are able to learn based on the model and are able to act based on that knowledge to interact with or change the domain (preferably in a positive direction relative to the goal of modeling). *Social quality* is determined based on agreement among audience members' interpretations of the model while the *organizational quality* of the model relates to that all statements in the model contribute to fulfilling the goals of modeling (organizational goal validity), and that all the goals of modeling are addressed through the model (organizational goal completeness).

Language quality is a mean to achieve model quality and relates the modeling language used, and its appropriateness for the modeling task based on six quality areas. *Domain appropriateness* relates the language and the domain. Ideally, the language must be powerful enough to express anything in the domain, not having what Wand and Weber (1993) terms construct deficit. On the other hand, you should not be able to express things that are not in the domain, i.e., what Wand and Weber (1993) terms construct excess. Domain appropriateness is primarily a mean to achieve semantic quality. *Participant appropriateness* relates the social actors' explicit knowledge to the language. Participant appropriateness is primarily a mean to achieve pragmatic quality both for comprehension, learning, and action. *Modeler appropriateness* relates the language to the knowledge of the modeler. Modeler appropriateness is primarily a mean to achieve semantic quality. *Comprehensibility appropriateness* relates the language to the social actor interpretation. The goal is that the participants in the modeling effort using the language understand all the possible statements of the language. Comprehensibility appropriateness is primarily a mean to achieve empirical and pragmatic quality. *Tool appropriateness* relates the language to the technical audience interpretations. For tool interpretation, it is especially important that the language lend itself to automatic reasoning. This requires formality (i.e., both formal syntax and semantics being operational and/or logical), but formality is not necessarily enough, since the reasoning must also be efficient to be of practical use. This is covered by what we term analyzability (to exploit any mathematical semantics of the language) and executability (to exploit any operational semantics of the language). Different aspects of tool appropriateness are means to achieve syntactic, semantic, and pragmatic quality (through formal syntax, mathematical semantics, and operational semantics, respectively). *Organizational appropriateness* relates the language to standards and other organizational needs within the organizational context of modeling. These are means to support organizational quality.

For more information on SEQUAL, please refer to Krogstie et al. (2006) and Lillehagen and Krogstie (2008).

3.3.1 Evaluating BPMN Using the Semiotic Framework

Semiotic evaluations of BPMN are performed by Nysetvold and Krogstie (2006), Wahl and Sindre (2005) and discussed in Recker et al. (2007). The approach has also been used for the evaluation and comparison of a number of other modeling notations. In relation to BPMN the following findings can be mentioned:

Support for business-specific terms. Wahl and Sindre (2005) confirm that the constructs of the language do not contain any business-specific terms even though the purpose of the language is the modeling of business processes. Because of this, it would be applicable to model nonbusiness-related processes using BPMN, but only to a certain extent.

Understanding and use of constructs. The language notation is similar to that of other available languages with the same purpose, which would be helpful with users familiar with different approaches. The goal of BPMN is, however, to be understandable not only for users with previous experience and the complexity of the most advanced aspects of BPMN is, according to the authors, unrealistic to grasp without extensive training. This is somewhat confirmed by the case study reported by zur Mühlen and Ho (2008) (see Sect. 3.7).

Diagram layout. The authors also argue that it would be hard to externalize relevant knowledge using only BPDs if the knowledge in question goes beyond the domain of business processes. There are few strict guidelines in the BPMN specification on how to layout diagram constructs in relation to each other, which proposes a potential for creating BPDs with poor empirical quality.

3.3.2 Empirical Evaluation of BPMN, EEML, and UML Activity Diagrams

Nysetvold and Krogstie (2006) conducted an empirical evaluation of BPMN relative to EEML (Krogstie 2008) and UML Activity Diagrams using the SEQUAL framework. The usage area to be supported was process modeling in relation to implementation of Service-Oriented Architecture (SOA) in an insurance company. The evaluation rank BPMN highest in all categories except domain appropriateness (expressiveness), in which EEML came out best. However, EEML lost to BPMN on both tool and modeler appropriateness. The evaluation on domain appropriateness partly overlapped the evaluations above, e.g., by including an evaluation relative to control patterns. Other parts of this evaluation were adapted particularly to the expressed needs in the organization based on existing experience.

Comprehensibility appropriateness is the category that was appointed the second highest importance (based on number of criteria), since the organization regarded it to be very important that it was possible to use the language across the different areas of the organization and to improve communication between the IT-department and the business departments. In this category, BPMN and Activity Diagrams ranked equally high, which is not surprising given that they use the same swimlane-metaphor as a basic structuring mechanism. The reason why EEML came out behind is primarily due to the graphical complexity of some of the concepts, combined with the fact that EEML has a larger number of concepts in total, not surprising given that is a general enterprise modeling notation also useful for data, resource, and goal modeling.

Participant appropriateness and tool appropriateness were given equal importance, and BPMN ranked somewhat surprisingly high on both areas. When looking at the evaluation not taking tool appropriateness into account, the three languages ranked almost equal. Thus, it was in this case the focus toward the relevant implementation platforms (BPEL and web services) that ranked BPMN highest. On the other hand, the focus on tool appropriateness did not appear to get in the way for the language as a communication tool between people, at least not in this case.

In the category organizational appropriateness, BPMN and Activity Diagrams ranked almost equal. The organization had used UML and Activity Diagrams for some time, but it also appeared that tools supporting BPMN were available for the relevant parts of the organization.

3.4 *Combined Semiotic, Ontological, and Workflow Patterns Evaluation*

Recker et al. (2007) propose a generic framework for language evaluation based on the combination of ontological, semiotic, and pattern-based evaluation. They report on the first attempt to classify existing theoretical frameworks for process modeling language evaluation by using this framework. Their work provides an evaluation of existing frameworks as well as an evaluation of BPMN. For more information on the framework, consult Recker et al. (2007).

Some general statements on BPMN can be summarized from the analysis based on the study of Recker et al. (2007), which partly confirms the findings of the studies performed by the standalone approaches:

Representation of state. BPMN lacks the capabilities to model state-related aspects of business processes and is limited, if not incapable of modeling states assumed by things and state-based patterns.

Specialization of constructs. BPMN lacks attributes in the specification of the language constructs.

Weak support for resource modeling. There is lacking support for representing resource patterns and the evaluation comment the same as Wohed et al. (2006) when regarding the lane and pool constructs that are additionally criticized for being overloaded.

Redundant constructs. There is a relatively high degree of construct redundancy, which might explain why there are as many as three different BPMN representations for the same basic workflow patterns (Wohed et al. 2006).

3.5 Formal Analysis Using Petri Nets

Dijkman et al. map BPMN models to Petri Nets to be able to use efficient analysis techniques already available for Petri Net models. In doing this, they are able to evaluate the semantic correctness of BPMN models as well as disambiguating the core constructs of BPMN. The approach is used for empirical analysis with BPMN models found online. For more information on their work, consult Dijkman et al. (2007).

In converting BPMN diagrams to Petri Nets, Dijkman et al. (2007) discovered some issues in the BPMN specification and discuss possible solutions for these.

Process models with multiple start events. This is a situation where the BPMN specification indicates that each start event should generate a process instance. In situations where there are multiple start events without wait, there has to be some correlation mechanism to link the occurrence of a start event to an appropriate process instance.

Process instance completion. This is a situation where there are multiple end events and no clear indication in the specification when a process model is considered to be "completed". When the first end is reached, or when all tasks have met their end.

Exception handling for concurrent subprocess instances. There are unaddressed issues in the specification regarding the interrupt caused by subprocesses experiencing exceptions in a parallel multi-instance activity. The unclarity is related to whether the exception caused would only affect the subprocess in question or all subprocess instances spawned by the invocation activity.

OR-join gateway. The semantics of OR-join gateways is argued to be unclear regarding the relative definition of "upstream". It is advised that the BPMN specification adopt existing semantics with a formal foundation rather than attempting to define a new one.

3.6 Semistructured Interviews of BPMN Users

One effort to seek empirical evidence of theoretical propositions is done by following up a BWW representational analysis (see Sect. 3.1) with semistructured interviews with BPMN users. The research questions for this study were initially to discover the representational shortcomings of BPMN in light of the BWW-framework and to discover which of these were perceived as actual shortcomings by the BPMN users. This study involved 19 participants from six organizations distributed over four Australian states. The results are reported in Recker et al. (2005, 2006).

A follow-up of this study is the latest reported empirical evaluation of BPMN. A web-based survey performed between May and August 2007 including 590 BPMN users from different parts of the world. A presentation of the results is available in Recker (2008).

Interviews based on weaknesses discovered by representational analysis uncover how this affects the users (Recker et al. 2006).

Workarounds to fit local needs. The general impression regarding construct deficit is that even though the participants claim that they do not need to model state changes, business rules, or system structure they in fact find workarounds and represent this information outside the BPD itself. In modeling events, as many as 74% did not experience any limitation in using BPMN for this, and the problem declined for users using the expanded set compared with interviewees using the core set of elements. This is in contradiction to the theoretical proposition claiming that there would be confusion connected to using the expanded set.

Construct overload. The analytical evaluation proposed that there would be ambiguities regarding the lane and pool constructs. This was supported by the interviews and is mainly based on the fact that these constructs are used to represent a whole range of different real-world constructs as discussed in Recker et al. (2007).

In reporting the web-based quantitative survey (Recker 2008), the following issues were identified:

Support for business rule specification. Rule specification is an essential task in understanding business processes, and it would be good to see that process modeling solutions acknowledge this a bit better and provide support for this. This is suggested by one of the participants to be as simple as an additional graphical symbol implying that there is a business rule at work.

Weak support for resource modeling. The ambiguity that comes with the flexible semantics of lanes and pools is contradictory to their ease of use in modeling. One advice here is to provide better support for differentiating the multiple purposes for which lanes and pools can be used.

Understanding and use of constructs. The survey show that there is some doubt related to the use of gateways, off-page connectors, and groups. Basically, there is confusion on when to use these concepts and why. This might stem from the fact that they are constructs of the model and not the process modeled. When it comes to events, it is a question of frustration related to selecting the right kind of event.

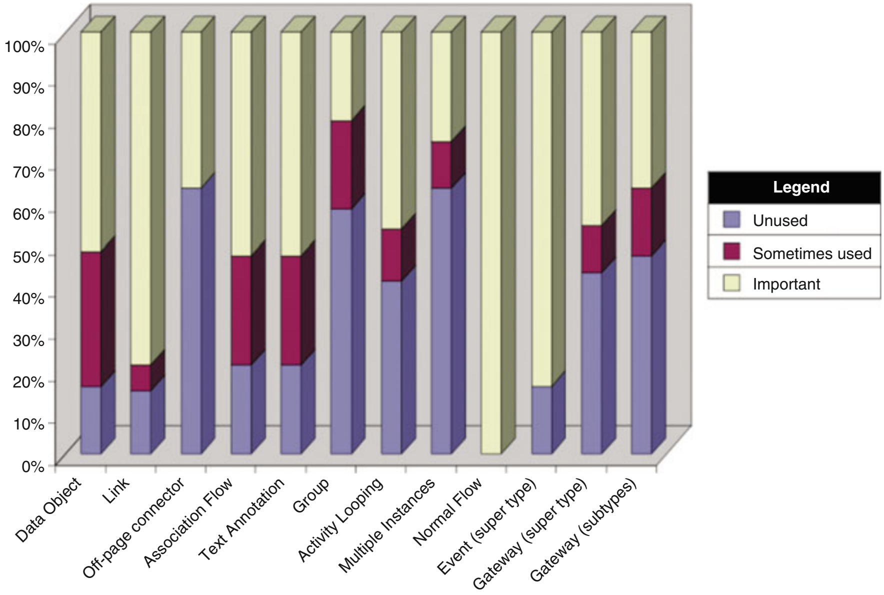

Fig. 5 Reported need for BPMN constructs (Adapted from Recker 2008)

Figure 5 shows results from the survey for the expressed need for the different BPMN constructs.

3.7 *Case Study of BPMN in Practice*

zur Mühlen and Ho (2008) followed the redesign of a service management process in a truck dealership in USA using action research. The study included reports on experiences from using BPMN with participatory modeling of the AS_IS and TO_BE process and the activation of the models for simulation purposes, providing the following results:

Understanding and use of constructs. Experience from the case study shows that the core set is used and understood. In cases where the entire set of BPMN constructs is used, the audience tends to disregard the richer meaning provided by the extended set (zur Mühlen and Ho 2008). The applied notation is primarily limited to the core constructs.

Workarounds to fit local needs. Use of constructs different from what suggested in the specification has been observed. Modelers purposely create syntactically wrong models to improve readability and to simplify the modeling task. One example of this is placing activity constructs across lanes to indicate that there are several organizational units participating in completing a task.

Tool dialects. The tool used had its own BPMN dialect that was not fully compliant with the official BPMN specification.

3.8 Statistical Analysis of BPMN Models

Similar to the work of Dijkman et al. (2007) mapping models to Petri Nets for analysis, zur Mühlen and Recker (2008) have coded BPMN models to Excel spreadsheets and used the representation with different mathematical tools for statistical analysis and comparison. The models investigated were collected from three different groups: models used in consulting project, models created as part of BPMN education seminars, and models found online. Investigated phenomena include the general use of constructs, their frequency of use, and the correlation of use of different constructs.

Modeling constructs use similar to that of natural language. By arranging constructs by frequency, the study revealed a distribution similar to the distribution previously observed for natural languages. This suggests that the use of BPMN constructs for expressing business processes mirrors the use of natural language. This would further suggest that expressiveness is based on the modelers existing vocabulary and that one will use whatever constructs one has knowingly available. The study found further support for this through observing that precise semantics is used by the consultant group and for models created in seminars, thus suggesting that this is based on formal training increasing construct vocabulary. Like many natural languages, BPMN has a few essential constructs, a wide range of constructs commonly used, and an abundance of constructs virtually unused (zur Mühlen and Recker 2008).

Precise constructs replace the need for text annotations. Another issue discovered by mapping the correlation of constructs is based on the negative correlation between the extended set gateways and text annotations. Text annotations seem to act as a substitute for formal event and gateway types by describing their behavior informally.

Practical language complexity does not equal theoretical complexity. Based on the result, the study also made an attempt to measure the practical complexity of BPMN based on the number of semantically different constructs used in each model. On average this resulted in the number of different constructs used as 9 (consulting), 8.87 (web), and 8.7 (seminars). There is, however, variation in what constructs are used, but nevertheless this has provided an image of a far less practical complex language compared with its theoretical complexity opening for as many as 50 different constructs in one model. Altogether, there was found six pairs of models out of 120 models examined that shared the same constructs, but there were several models sharing the same construct combinations or subsets.

Models focus on choreography or orchestration, not both. By organizing the model subsets using Venn diagrams showing what subsets were used in combination, the study revealed that modelers either focus on process orchestration by refining models by means of extended gateways or they focus on process choreography by adding organizational constructs, such as pools and lanes (zur Mühlen and Recker 2008).

4 Reported Results of the Evaluations

Even if there were criticism of a modeling approach based on analytical evidence, the potential weaknesses would have to be backed up or confirmed empirically to determine its real impact. A weakness based on analytical proof found in some remote part of a specification might not even be apparent to the user not aware of its existence, or in the opposite case the user might end up designing erroneous or ambiguous models due to poor formalism or tool support.

In this section, we will look at both the analytical and empirical evaluations together to identify similarities and difference. We will see that the consequences of the findings to a large extent depend on the goal of the modeling task, and that the goal of the language itself also must be taken into consideration when assigning the final score. BPMN seeks to serve both a broad audience in the business segment on the one hand, and on the other hand it reaches out to the technical community. In doing so, it is of potential use within all five categories of process modeling, as suggested by Krogstie et al. (2008), and further it has several groups of users whose requirements for use and modeling goals are quite different.

We will use the six language quality areas of SEQUAL (Krogstie et al. 2006) to classify the findings in the different evaluations. This is both out of convenience and based on the fact that it is a readily available framework for classifying quality, and thus it should be able to cover the findings.

4.1 Domain Appropriateness

Weak support for resource modeling is discovered using the Workflow Patterns Framework and the generic framework. This is confirmed also by the semistructured interviews and web-based surveys. In addition the BWW framework finds BPMN to have weak support for modeling system structure. The statistical analysis shows that BPMN models focus on choreography or orchestration, not both.

The BWW and Workflow Patterns Framework also find the representation of state to be weak. The generic framework confirms this, which does not come as a surprise since it is based on the first two.

4.2 Modeler Appropriateness

Missing support for business rule specification is one weakness mentioned in the web-based survey, whereas the semiotic and generic evaluation framework is missing the support for business-specific terms or specialized constructs. One workaround for these issues is observed in the semistructured interviews where there are cases where own constructs are used to fit the modeling needs. There is also an observed difference in the use of text annotations, particularly they tend to be used less for models designed by using more precise constructs from the extended set and in the opposite case act as a surrogate for the expressiveness of rich constructs in less precise models.

4.3 Participant Appropriateness

Several evaluations discuss the understanding and use of constructs and the key findings include the fact that some form of training is needed to use BPMN properly. Constructs like the off-page connectors support modeling and not the process which can be confusing for some users.

4.4 Comprehensibility Appropriateness

There are redundant constructs in BPMN and there are cases of multiple representations of the same patterns. In addition the lane and pool constructs are considered to be overloaded. The practical language complexity does not, however, equal the theoretical complexity and in understanding models, there is a tendency to disregard the richer meaning of the extended set. This is probably the only area in which the empirical evaluations do not directly support the analytical.

4.5 Tool Appropriateness

Workflow patterns report the lack of support for representation of multiple instances.

The Petri net analysis reveals some issues regarding the use of BPMN for simulation in cases with multiple start or end events and concurrency of subprocesses. There are also indications of a need for a more formal definition of the semantics of the language.

4.6 Organizational Appropriateness

The case study of BPMN in practice discovered an issue related to the fact that there are several different tool dialects and these are not fully compliant with the BPMN specification.

5 BPMN Extensions

Results from the evaluations show that users are able to find workarounds for some of the weaknesses found in BPMN. In most of these cases, there is a gap between what is possible to achieve using BPMN and the desired goal of the user. One way to approach this problem is by building extension to close this gap, and by doing this, prototype different kinds of functionality possible to include in the BPMN specification. The following section presents four reported efforts to extend BPMN and by this show identified weaknesses discovered by means of practical use and proposed solutions for these weaknesses. The first three proposals address issues related to choreography, semantic correctness, and modeling of resources while the fourth discusses a topic not discussed in the evaluations but which is still important: Combining user-interface modeling with process modeling which is relevant in scenarios involving the reengineering of existing processes supported by information systems for the end user.

5.1 Using BPMN for Modeling Choreography

An assessment of BPMN using the Service Interaction Patterns (Barros et al. 2005) presented by Decker and Puhlmann (2007) shows weak support for modeling complex choreographies in BPMN. This weakness is connected to distinguishing between several instances of participants and using references to single participants for messaging. By adding participant sets, references, reference sets, and reference passing to BPMN this paper demonstrates that it would be possible to support most of the service integration patterns. The authors also point out an unclarity in the semantics of the BPMN data objects regarding their ability to buffer data similar to what is possible in UML Activity Diagrams. Based on this, a required distinction between data object and data object sets is introduced to their extension of BPMN. Aspects raised by the need of choreography modeling are discussed by Barros et al. (2009) in this Handbook.

5.2 Checking Semantic Correctness Using Petri Nets

By using the XML serialization created by a BPMN tool, Dijkman et al. (2007) have implemented a tool to translate BPMN models to Petri Nets via the Petri Net Markup Language (PNML). Once converted to a Petri Net, the BPMN model can be semantically analyzed using Petri net analysis toolset. This work is limited to the control-flow perspective of BPMN and the order in which activities and events are allowed to occur. Weaknesses found in this paper are discussed in Sect. 4, but the suggested extension allowing semantic validation of BPMN models is considered to be a potentially helpful tool for assisting the building of formal models.

5.3 Modeling of Task-Based Authorization Constraints in BPMN

An extension of BPMN is suggested by Wolter and Schaad (2007) to support resource allocation patterns. These patterns allow specifying authorization constraints, for instance role-task assignments, separation of duty, and binding of duty constraints. This is done by adding security relevant semantics to the group and lane elements of BPMN and deriving a new textual artifact from the textual annotation element. Extending BPMN with the support for describing security aspects of workflow can widen its scope and application and can be relevant also for modeling business scenarios.

5.4 Combined User-Interface and Process Modeling

The main approach for execution support of BPMN is mapping to BPEL. On the other hand, the focus of BPEL engines is on process executions and not on the user-interface of the applications, which in practice can result in good process support systems that is hampered by an inappropriate user-interface, thus meeting unnecessary implementation problems. Trætteberg (2008) presents an approach for combining model-based user-interface design (MBUID)-approaches with BPMN as a task modeling language to make it easier to develop appropriate user-interfaces and user-interfaces applicable for user tailoring for BPM-solutions.

6 Discussion and Conclusions

This chapter has identified and reported on the main efforts to evaluate BPMN, both analytical and empirical. From the findings it is possible to suggest that the analytical evaluations performed are at this point sufficient and self-confirming. Even though there is little evidence from the empirical investigations so far, it seems like most of the weaknesses uncovered by analytical evaluations are by the users treated lightly and through workarounds.

Local model interpretation and tool dialects might be problematic, as models will not be directly available for externalization and interoperability issues might arise when moving models between organizations or groups within organizations.

Two issues related to tool appropriateness not mentioned by the reported evaluations covered already, but which are apparent problems in BPMN, are that there is no explicit meta-model for BPMN and there is not specified any means for interchanging BPDs between the different modeling tools (Frankel 2008).

By limiting the evaluation of practical use of BPMN within one organization or group, some of the analytically identified weaknesses might not be problematic since the model has limited use and fit local (but not organizational) goals. When evolving the same model through different phases, from sense-making to analysis through simulation, and when integrating the model to the process by involving different tools for modeling, simulation, and execution, which also requires different levels of formalism and detail and user skill, this suggests that BPMN in fact does not scale up for the use across organizations unless there is formal training based on precise semantics and that the BPMN tools are built on a precise meta-model.

There is a level of freedom requested by the modelers not needing to express formal models and by restricting the creation of ad hoc models and process sketches one might discriminate against one of the key user groups. The question is whether formality and freedom are in conflict and if there are conflicts within the goal of the language of being readily available for both technical and nontechnical users.

The focus in most evaluations so far has been on BPMN in isolation and, except for two cases, little comparison between BPMN and other approaches has been done. The evaluations on which this report is based are primarily based on BPMN 1.0 and not the maintenance version (BPMN 1.1). As for the empirical studies these are partly reliant on the local implementation of BPMN and the dialect of the BPMN tool in question, rather than the specification.

On the account of BPMN 2.0 it might be that there are issues within BPMN that are more important to solve than others in order for the continued use and growth of BPMN. The overall goal for BPMN 2.0 (OMG 2007) is to integrate both notations, meta-model and interchange format within one language. Requested features include the following: Aligning BPMN with the Business Process Definition Meta-model (BPDM). Based on current proposals (Frankel 2008), it is not sure whether BPMN will be used as meta-model or if there will be a dedicated BPMN 2.0 meta-model mapping to BPDM; Enabling the exchange of business process models and their diagram layouts among process modeling tools to preserve semantic integrity; Expand BPMN to allow model orchestrations and choreographies as stand-alone or integrated models; Support the display and interchange of different perspectives on a model that allow a user to focus on specific concerns; Serialize BPMN and provide XML schemas for model transformation and to extend BPMN toward business modeling and executive decision support (Recker 2008). The RFP also rate consistency checks and model validation as important features.

From the empirical studies one can further see that there is a difference in the perceived use of BPMN regarding the use of the core or the expanded set. Few of the studies indicate whether they are based on the one or the other, which might impose a problem on the user-side. One might select BPMN for a task based on expressiveness, but planning to use the core set which at one point would go wrong.

There is room for more empirical work on the actual use of BPMN. It would be wise to perform replication studies on future BPMN work on the revision of the standard when it becomes available to determine eventual improvement.

Some other questions for future work are: How fast the tool support for a revised version of the standard will be available and what are the consequences of having two significantly different versions available? How will the different versions of BPMN map to each other? If the proposed weaknesses found impose actual problems or if the workarounds found among the users (extending BPMN with local support utilities of their choice) provide a better approach all together than trying to build an all-in-one language.

References

Abbot KR, Sarin SK (1994) Experiences with workflow management: issues for the next generation, In: ACM CSCW conference

Ambriola V, Conradi R, Fuggetta A (1997) Assessing process-centered software engineering environments. ACM Trans Softw Eng Meth 6(3):283–328

Barros A, Dumas M, ter Hofstede AHM (2005) Service interaction patterns. In: Proceedings of the 3rd international conference on business process management, Nancy, France, September 2005, Springer Verlag, pp 302–318

Barros A, Hettel T, Flender C (2009) Bridging the gap from process analysis to design through choreography modeling. In: Rosemann M, vom Brocke J (eds) Handbook on business process management. Berlin, Springer

Bolcer G, Kaiser G (1999) SWAP: leveraging the web to manage workflow. IEEE Internet Comput 3(1):85–88

Booch G, Rumbaugh J, Jacobson I (2005) The unified modeling language: user guide, 2nd edn. Addison-Wesley, Reading, MA

Carlsen S (1998) Action port model: a mixed paradigm conceptual workflow modeling language. In: Third IFCIS conference on cooperative information systems (CoopIS'98), New York

Decker G, Puhlmann F (2007) Extending BPMN for modeling complex choreographies. In: Proceedings of the 15th international conference on cooperative information systems (Coop-IS), LNCS 4803

Derniame JC (ed) (1998) Software process: principles, methodology and technology. Lecture notes in computer science, vol 1500. Springer, Berlin, Heidelberg, New York

Dijkman RM, Dumas M, Ouyang C (2007) Formal semantics and analysis of BPMN process models. Queensland University of Technology, Brisbane

Fischer L (2000) Excellence in practice IV – innovation and excellence in workflow and knowledge management. Workflow Management Coalition, Future Strategies Inc., Florida, USA

Fox MS, Gruninger M (2000) Enterprise modeling, AI Magazine

Frankel DS (2008) Requirements for BPMN 2.0, MDA Journal. http://www.bptrends.com/. Accessed June 2008.

Gane C, Sarson T (1979) Structured systems analysis: tools and techniques. Prentice Hall, Englewood Cliffs, NJ

Gemino A, Wand Y (2003) Evaluating modeling techniques based on models of learning. Commun ACM 46:79–84

IDEF-3x (1993) Process modeling language specification, standard NA-94-1422B, Rockwell International

Krogstie J (2008) Integrated goal, data and process modeling: from TEMPORA to model-generated work-places. In: Johannesson P, Søderstrøm E (eds) Information systems engineering from data analysis to process networks. IGI Publishing, Hershey, PA, pp 43–65

Krogstie J, Jørgensen H (2004) Interactive models for supporting networked organizations. In: 16th conference on advanced information systems engineering. Springer, Riga, Latvia, Berlin, Heidelberg, New York

Krogstie J, Sindre G, Jørgensen HD (2006) Process models representing knowledge for action: a revised quality framework. Eur J Inform Syst 15(1):91–102

Krogstie J, Dalberg V, Jensen SM (2008) Process modeling value framework, Enterprise Information Systems In: Manolopoulos Y, Filipe J, Constantopoulos P, Cordeiro J (eds) Selected papers from 8th international conference, ICEIS 2006, Paphos, Cyprus, May 23–27, 2006. Lecture notes in business information processing, vol 3. Springer, Berlin

Kuntz JC, Christiansen TR, Cohen GP, Jin Y, Levitt RE (1998) The virtual design team: a computational simulation model of project organizations. Commun ACM 41(11):84–92

Lillehagen F, Krogstie J (2008) Active knowledge models of enterprises. Springer, Heidelberg

Nysetvold AG, Krogstie J (2006) Assessing business process modeling languages using a generic quality framework. In: Siau K (ed) Advanced topics in database research, vol 5. Idea Group. Hershey, Pennsylvania, pp 79–93

OMG (2000) Workflow management facility v. 1.2. Object Management Group

OMG (2007) Business process model and notation (BPMN) 2.0 request for proposal. Technical report, OMG. http://www.omg.org/. Accessed June 2007

OMG (2008) BPMN v1.1 specification. Technical report, OMG. http://www.omg.org/, http://www.omg.org/spec/BPMN/1.1/. Accessed January 2008.

Ouyang C et al. (2010) Workflow management. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1. Springer, Heidelberg

Recker J (2008) BPMN modeling – who, where, how and why. BPTrends

Recker J, Indulska M, Rosemann M, Green P (2005) Do process modelling techniques get better? A comparative ontological analysis of BPMN. 16th Australasian conference on information systems, Sydney, Australia

Recker J, Indulska M, Rosemann M, Green P (2006) How good is BPMN really? Insights from theory and practice. 14th European conference on information system (ECIS 2006). Gothenburg, Sweden, 1–12

Recker J, Rosemann M, Krogstie J (2007) Ontology-versus pattern-based evaluation of process modeling languages: a comparison. Comm Assoc Inform Syst 20:774–799

Reijers HA et al. (2010) Business process quality management. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1. Springer, Heidelberg

Rosemann M, Recker J, Indulska M, Green P (2006) A study of the evolution of the representational capabilities of process modeling grammars. Advanced Information Systems Engineering-CAiSE, vol 4001, 447–461

Russell N, ter Hofstede AHM, van der Aalst WMP, Mulyar N (2006) Workflow control-flow patterns: a revised view. BPM Center Report BPM-06-22, BPMcenter. org, 6–22

Scheer AW (2000) ARIS – business process modeling, 3rd edn. Springer, Berlin

Trætteberg H (2008) UI design without a task modeling language – using BPMN and Diamodl for task modeling and dialog design. To be published at Tamodia 2008

van der Aalst, WMP, Desel J, Oberweis A (eds) (2000) Business process management. Lecture notes in computer science, vol 1806. Springer, Berlin, Heidelberg, New York

van der Aalst WMP, ter Hofstede AHM, Kiepuszewski B, Barros AP (2003) Workflow patterns. Distrib Parallel Databases 14:5–51

van der Aalst WMP et al. (2010) Business process simulation. In: vom Brocke J, Rosemann M (eds) Handbook on business process management, vol 1. Springer, Heidelberg

Wahl T, Sindre G (2005) An analytical evaluation of BPMN using a semiotic quality framework. In: Proceedings of EMMSAD'2005, 533–544

Wand Y, Weber R (1993) On the ontological expressiveness of information systems analysis and design grammars. J Inform Syst 3(4):217–237

WfMC (2000) Workflow Handbook 2001. Workflow Management Coalition, Future Strategies Inc., Lighthouse Point, Florida, USA

White SA (2004) Introduction to BPMN. IBM Cooperation

Wohed P, van der Aalst WMP, Dumas M, ter Hofstede AHM (2005) Pattern-based analysis of BPMN – an extensive evaluation of the control-flow, the data and the resource perspectives (revised version), BPM Center Report BPM-06-17, BPMcenter.org

Wohed P, van der Aalst WMP, Dumas M, ter Hofstede AHM (2006) On the suitability of BPMN for business process modelling. Business Process Management – BPM 2006, Vienna, Austria. Lecture notes in computer science, vol 4102. pp 161–176

Wolter C, Schaad A (2007) Modeling of task-based authorization constraints in BPMN. BPM Conference. Springer, 64–79

zur Mühlen M, Ho DT (2008) Service process innovation: a case study of BPMN in practice. Hawaii international conference on system sciences, proceedings of the 41st annual, 2008

zur Mühlen M, Recker JC (2008) How much language is enough? Theoretical and practical use of the business process modeling notation, Proceedings CAiSE'08. Springer Verlag, Montpellier, France