How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation

Michael zur Muehlen[1] and Jan Recker[2]

[1] Stevens Institute of Technology, Howe School of Technology Management, Castle Point on Hudson, Hoboken, NJ 07030 USA
Michael.zurMuehlen@stevens.edu
[2] Queensland University of Technology, Faculty of Information Technology, 126 Margaret Street, Brisbane QLD 4000, Australia
j.recker@qut.edu.au

Abstract. The Business Process Modeling Notation (BPMN) is an increasingly important industry standard for the graphical representation of business processes. BPMN offers a wide range of modeling constructs, significantly more than other popular languages. However, not all of these constructs are equally important in practice as business analysts frequently use arbitrary subsets of BPMN. In this paper we investigate what these subsets are, and how they differ between academic, consulting, and general use of the language. We analyzed 120 BPMN diagrams using mathematical and statistical techniques. Our findings indicate that BPMN is used in groups of several, well-defined construct clusters, but less than 20% of its vocabulary is regularly used and some constructs did not occur in any of the models we analyzed. While the average model contains just 9 different BPMN constructs, models of this complexity have typically just 4-5 constructs in common, which means that only a small agreed subset of BPMN has emerged. Our findings have implications for the entire ecosystems of analysts and modelers in that they provide guidance on how to reduce language complexity, which should increase the ease and speed of process modeling.

Keywords: BPMN, Language Analysis, Process Modeling.

1 Introduction

The Business Process Modeling Notation (BPMN) [1] is emerging as a standard language for capturing business processes, especially at the level of domain analysis and high-level systems design. A growing number of process design, enterprise architecture, and workflow automation tools provide modeling environments for BPMN. The development of BPMN was influenced by the demand for a graphical notation that complements the BPEL standard for executable business processes. Although this development gives BPMN a technical focus, the intention of the BPMN designers was to develop a modeling language that can equally well be applied to typical business modeling activities. This is clearly visible in the specification document, which

Z. Bellahsène and M. Léonard (Eds.): CAiSE 2008, LNCS 5074, pp. 465–479, 2008.

separates the BPMN constructs into a set of core graphical elements and an extended, more specialized set. BPMN's developers envisaged the core set to be used by business analysts for the essential, intuitive articulation of business processes in very easy terms. The full set of constructs would then enable users to specify even complex process scenarios with a level of detail that facilitates process simulation, evaluation or even execution. This separation mirrors an emerging tendency in industry to separate business-focused process modeling from implementation-oriented workflow implementation.

The evolution of BPMN closely mirrors the emergence of another modeling standard, UML [2]. Both have been ratified by the standardization body OMG. Both contain a larger set of constructs in contrast to competing languages, and offer a multitude of options for conceptual modeling. Both have been found in analytical studies to be not only semantically richer but also theoretically more complex than other modeling languages, [e.g., 3, 4]. And, in UML's case, this complexity motivated users to deliberately reduce the set of constructs for system analysis and design tasks. Related studies found that frequently not even 20% of the constructs are used in practice [5, 6].

The apparent complexity of the BPMN standard seems to be similar to the UML standard, which raises a number of questions: Are BPMN users able – and willing – to cope with the complexity of the language? Does the separation into core and extended constructs provided by the specification hold in modeling practice? And – really – how exactly is BPMN used in practice?

While BPMN has been receiving significant attention not only in practice but also in academia, virtually all contributions have been made on an analytical or conceptual level, [7, 8]. There are only few empirical insights into how BPMN is used in practice – exceptions are reported in [9] and [10].

Accordingly, our research imperative has been to provide empirical evidence on the usage of BPMN in real-life process modeling practice. The *aim of this paper* is to examine, using statistical techniques, which elements of BPMN are used in practice. We collected a large set of BPMN diagrams from three different application areas (i.e., consulting, education, process re-engineering) and analyzed the models regarding their construct usage. This study is a first step to determine the most commonly used set of BPMN constructs and to provide the ecosystem of process modelers with specific advice which elements of BPMN to use when. BPMN training programs could benefit from a structure that introduces students to the most commonly used subset first before moving on to advanced modeling concepts.

We proceed as follows: The next section briefly introduces the background of our research, viz., BPMN and our data sources, and presents our research design. Section 3 presents the analysis results and discusses them. Section 4 concludes this paper with a discussion of contributions, implications and limitations, and provides an outlook to future research.

2 Background

2.1 Introduction to BPMN

The Business Process Modeling Notation [1] is a recently published notation standard for business processes. Its development has been based on the revision of other

notations including UML, IDEF, ebXML, RosettaNet, LOVeM and Event-driven Process Chains.

BPMN was developed by an industry consortium (BPMI.org), whose constituents represented a wide range of BPM tool vendors but no end users. The standardization process took six years and more than 140 meetings, both physical and virtual. The BPMN working group developed a specification document that differentiates the BPMN constructs into a set of core graphical elements and an extended specialized set. The complete BPMN specification defines 50 constructs plus attributes, grouped into four basic categories of elements, viz., Flow Objects, Connecting Objects, Swimlanes and Artefacts. *Flow Objects*, such as events, activities and gateways, are the most basic elements used to create BPMN models. *Connecting Objects* are used to inter-connect Flow Objects through different types of arrows. *Swimlanes* are used to group activities into separate categories for different functional capabilities or responsibilities (e.g., different roles or organizational departments). *Artefacts* may be added to a model where deemed appropriate in order to display further related information such as processed data or other comments. For further information on BPMN refer to [1].

Existing research related to BPMN includes, *inter alia*, analyses and evaluations, [e.g., 9, 11], use in combination with other grammars, especially BPEL [7], or its support for workflow concepts and technologies [8]. This and other research is mostly analytical in nature. Few insights exist into the practical use of BPMN, which has motivated our study.

2.2 Data Sources

In order to arrive at an informed opinion about the use of BPMN in practice we collected BPMN models from three types of sources: A search using Internet search engines for "BPMN model" resulted in 57 BPMN diagrams, obtained from organizations' web sites, from practitioner forums and similar sites. These diagrams were labeled in a variety of languages, but since our study focuses on the modeling constructs and not their content this was no hindrance. We collected an additional 37 BPMN diagrams from consulting projects to which we had access. These diagrams depicted as-is and to-be processes from business improvement projects or software deployment projects. An additional 26 diagrams were collected through BPMN education seminars taught by the authors. These diagrams were created by seminar participants and depicted business processes from the participants' organization. Overall, our data set consists of 126 BPMN models approximating the use of BPMN for a variety of purposes including process (re-) design, education, consulting, and software and workflow engineering. 6 models were excluded from the analysis because they explicitly illustrated nonsensical diagrams or were duplicates.

While by no means do we claim our data set to be statistically representative of the overall use of BPMN in practice, it nevertheless gives us an informed opinion about the *real* use of BPMN beyond the examples typically given by developers or tool vendors.

2.3 Research Design

Having obtained a large set of BPMN models, our next step was to prepare these models for analysis. We created an Excel spread sheet counting the type of BPMN

constructs in use per model. Each occurrence of a BPMN construct was marked as 1, otherwise 0. This coding allowed us to treat the individual models as binary strings for further analysis. In our coding effort, we kept track of the data sources for each model, which, for analysis purposes, we labeled 'web' (those models that we obtained from Internet search engines), 'consulting' (those that we obtained from consulting engagements) and 'seminar' (those obtained from educational seminars).

The resulting tables provided the basis for the application of statistical techniques such as cluster analysis, frequency analysis, covariance analysis and distribution analysis. We employed analysis techniques available in Excel (frequency counts), Mathematica (covariance matrices, Hamming distances) and R (cluster analysis). The following sections provide further details about the exact application of the various techniques used, and discuss the results we obtained.

3 Analysis and Discussion

3.1 Overall Use of BPMN Constructs

BPMN offers 50 modeling constructs, ranging from Task and Sequence Flow to Compensation Associations and Transaction Boundaries. Our first question was: Which of these symbols are used in practice and how frequently?

Fig. 1 shows the frequency distribution of the individual BPMN constructs, separated by the three sample sets and ranked by overall frequency. Generally speaking, the distribution of constructs follows a power-law distribution, with only four constructs being common to more than 50% of the diagrams: Sequence Flow, Task, End Event, and Start Event. Notably, these constructs all belong to the originally specified BPMN core set [1].

Fig. 1 shows that every model contained the Sequence Flow construct, and nearly every model contained the basic Task construct (the diagrams that did not contain the Task construct used the Subprocess construct). The majority of Web and Seminar models contained Start and End Events, while the Consulting models replaced these with more specific event types (e.g., Message or Timer Events for Start Events, Terminate, Message, or Link, for End Events). The other BPMN constructs were unevenly distributed. A visual inspection of Fig. 1 leads to a number of interesting observations:

While the majority of consulting models contained Data-based XOR Gateways (77%), Pools (81%) and Lanes (69%), these constructs were much less frequent in the other two sample sets (57%, 30%, 21% and 23%, 56%, 16% respectively for web and seminar models). This indicates that the consulting models depict organizational structure in more detail than the random web sample. The majority of consulting models contained detailed Gateway constructs, whereas only ¼ of the seminar models did not used them. This implies that beginning modelers tend to create diagrams with few alternative or parallel flows.

The Web diagrams use (non-specific) Gateways frequently (observed in 55% of the models), whereas the consulting and seminar sets make much less use of this symbol (5% and 12%, respectively). Models in the web sample express the control flow logic of the diagrams in plain text (which can be inserted into the basic Gateways), rather than the more formal XOR, AND, and Inclusive OR constructs.

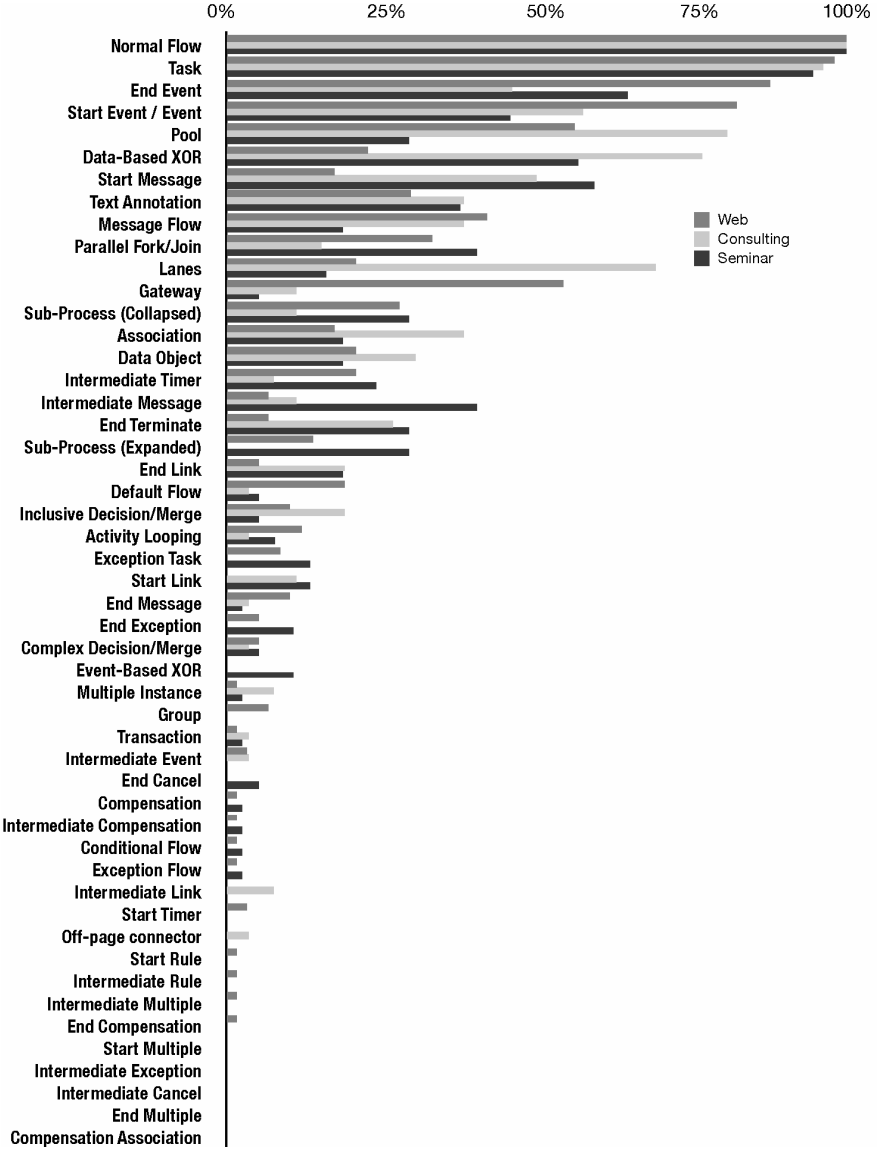

Fig. 1. Occurrence Frequency of BPMN Constructs

A sizable fraction of seminar models contain Intermediate Message constructs (41%) whereas only 7% of web models and 12% of consulting models contain this construct. This indicates that this construct is emphasized in BPMN classes but not very common in practice. A potential explanation may stem from the underlying design paradigm for process choreography in BPMN, which typically requires a lot of time to explain in classrooms. Practitioners in general may not be fully confident in

the use of these choreography concepts, which could be explain the less frequent usage of the related constructs.

3.2 Frequency Distribution of BPMN Constructs

The ranked frequency distribution of BPMN constructs generally follows an exponential (power-law) distribution, similar to long-tailed distributions that have been observed as a result of preferential attachment [12]. This particular shape has been observed previously in studies of natural languages, [e.g., 13, 14]. Fig. 2 shows a plot of the frequency distribution of the BPMN elements in the three sample sets compared with the Zipfian distribution [14].

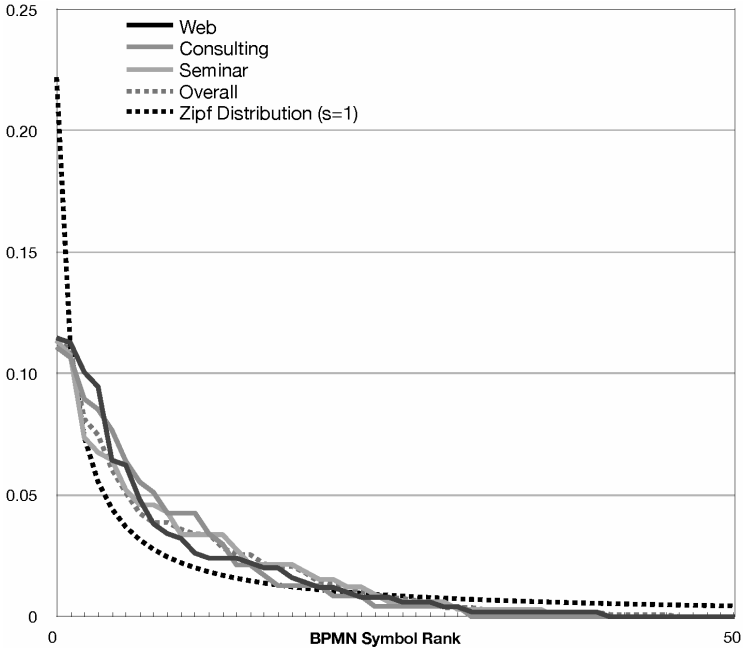

Fig. 2. Frequency Plot of BPMN Constructs by Rank

Zipf's Law states that the frequency of words in natural languages is inverse to their rank (in other words, the second most frequent word is used 1/2 the time of the first, the third most frequent word 1/3 of the time, and so on) and has been observed in numerous contexts [see, for instance, 13]. While not a perfect fit, the BPMN subsets exhibit a distribution that is very close to the distribution of word usage in natural languages. This suggests that the use of BPMN constructs to design (graphical) statements about organizational or system processes mirrors the use of natural languages.

This finding is of importance for future research on the way users learn, retain, and use BPMN constructs, and – really – any other graphical modeling language. For instance, linguistics research could be used to formulate conjectures about appropriate modeling training programs – a still under-researched aspect of modeling research in

IS. In general terms, the distribution of BPMN constructs shows that BPMN – as many natural languages – has a few essential constructs, a wide range of constructs commonly used, and an abundance of constructs virtually unused. Based on this observation, training and usage guidelines can be designed to reduce the complexity of the language to inexperienced analysts and to deliberately build such models that can safely be assumed to depict the core essence of a process without adding too much complexity.

3.3 BPMN Construct Correlations

Having determined the most frequent set of BPMN constructs in use, we turn to some related questions: Which of the BPMN constructs are typically used in combination? Which are used in alternation? In order to answer these questions, we used Mathematica to generate covariance matrices, which allowed us to examine pairs of BPMN constructs with regard to their combined or alternative use. Those pairs of constructs with negative covariance ($p < -0.05$) indicate alternatively used constructs while those with positive covariance ($p > 0.05$) indicate constructs used in combination. Table 1 summarizes the results.

Table 1. Combined and alternative use of BPMN constructs

Constructs with p > 0.05	Constructs with p < -0.05
Data Object → Association	Start Event → Start Message
Pool → Message Flow	Gateway → Data-based XOR
Start Event → End Event	Text Annotation → Message Flow
Start Message → Data-based XOR	Start Message→ End Event
Start Message → Intermediate Message	Start Message → Gateway
Start Message → End Terminate	Start Event → Data-based XOR
Pool → Lane	End Event → Data-based XOR
Lane → Message Flow	

Our findings present some interesting implications regarding BPMN modeling practice. Looking at the combined use of BPMN constructs (left column in Table 1), most correlations confirm that BPMN modeling practice obeys the grammatical rules of BPMN. For instance, Data Objects need to be linked to flow objects via the Association constructs, Pools can only communicate with other Pools via message flow, Lanes require Pools, and BPMN models require both Start and End Event. However, at least two interesting observations emerge. First, the positive correlation of Start Message events with End Terminate events indicates a more sophisticated level of BPMN modeling, suggesting that when users start using the differentiated event constructs, they tend to use a variety of these. Similarly, the combined use of Start Message events with the Data-based XOR constructs indicates an advanced use of the language for models in which different types of messages lead to different variants of a process, depending on the actual content of the arriving message.

Looking at the alternative use of BPMN constructs (right column in Table 1), we can identify additional interesting patterns of BPMN use. For instance, the negative correlation between Gateway and Data-based XOR suggests that when modelers refine the semantics of their models they choose the data-based XOR over the unspecific

Gateway in order to clarify the control flow semantics of their models. The negative correlation between Text Annotation and Message Flow suggests that at initial stages, modelers avoid choreography concepts and instead use free-form text to indicate message exchange. More advanced modeling relies on the provided semantic constructs instead of simple textual additions. Similarly, the negative correlations between Start Message event and the Gateway construct, and the Start/End Event and the Data-based XOR imply that modelers who refine the event constructs have achieved a level of sophistication of language use at which they avoid the use of the non-descriptive gateways altogether and instead rely on the more differentiated gateway and event subtypes.

3.4 BPMN Construct Clusters

In addition to identifying pairs of constructs that are used alternatively or in combination, we were also interested in uncovering whether clusters of BPMN constructs can be found in practice. To that end, we performed a hierarchical cluster analysis using the Euclidian distance measure in order to classify the set of BPMN constructs into distinct subsets. Fig. 3 shows the resulting dendrogram.

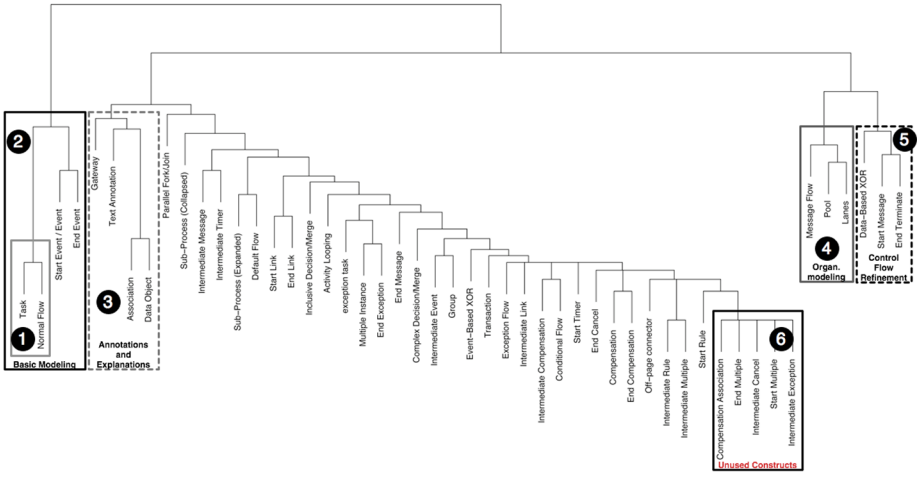

Fig. 3. Cluster Dendrogram of BPMN Constructs

In Fig. 3 six construct clusters are highlighted. First, the Task and Normal Flow cluster depicts the core of process modeling – the orchestration of activities that constitute a business process. Together with start and event conditions (through the use of events), these clusters indicate the simplest form of depicting the essence of a process in a graphical model. A third cluster is comprised of elements that are used to embellish and explain such process models through the use of text annotations, gateways (that specify control flow conditions of sequences of tasks) and data processing information. Clusters four and five essentially denote additions to these core modeling concepts by adding information about the organizational task allocation schemes,

required roles and responsibilities as well as choreography information in collaborative scenarios, or refinements to the orchestration of the flow of the process through different types of event and gateway constructs. The sixth cluster we found denotes the set of constructs that are very simply not used at all (e.g., compensation association, end message, etc).

The clustering of BPMN constructs provides a promising starting point for a complete ecosystem of BPMN users – vendors, consultants, coaches and end users alike. These users can be guided in their efforts to learn and apply BPMN in an effective and efficient manner. Training programs, for instance, could focus on the 'basic modeling' clusters first before teaching advanced concepts such as organizational modeling and control flow orchestration. Coaches and consultants in charge of modeling conventions are guided by delineating the most common – and most frequently avoided – BPMN constructs.

3.5 Core or Extended Set?

According to the BPMN specification, BPMN modelers are envisaged to choose either the core set of ten BPMN constructs, or an extended set in which these core constructs are modified (i.e., revised and extended). Our questions are: Do modelers use core or extended constructs? Do they comply with the differentiation?
In order to answer these questions we split the modeling constructs into 10 sets:

- Tasks are split into Basic Tasks and an extended task set which contains the constructs for Subprocesses (collapsed and expanded) as well as Tasks with additional semantics, such as Multiple Instance Tasks, Compensations, or Transactions.
- Sequence flow constructs are split into a basic set (the Normal Flow) and an extended set (consisting of Default Flow, Conditional, and Exception flow).
- Gateways are split into the Basic (blank) XOR Gateway, and an extended Gateway set, which comprises Data- (X-labeled) and Event-based XOR, Inclusive-OR, and Parallel Gateways. We contrast these two sets with the representation of routing information through the Conditional Sequence Flow construct.
- Events are split into the Basic Events, and an extended Event set including constructs such as Messages, Rule Events, Links, etc.
- In addition we distinguished from these constructs Layout elements such as off-page connectors and the Grouping construct.

For these sets, we performed three separate frequency counts, for each of the three data sets. The results are shown in Fig. 4.

The usage patterns exhibited in Fig. 4 shed some light on when users turn to elements from the extended set of BPMN constructs. First, while users tend to employ basic task and sequence flow constructs, they mostly employ an extended set of gateway constructs. Especially the sequence flow extensions are rarely used in practice. In terms of event constructs, basic and extended sets appear to be equally utilized. The following additional observations can be made from the frequency analysis:

- Consultants especially avoid extended task constructs and use mainly basic tasks. On the other hand, they largely utilize the set of specialized gateway constructs.

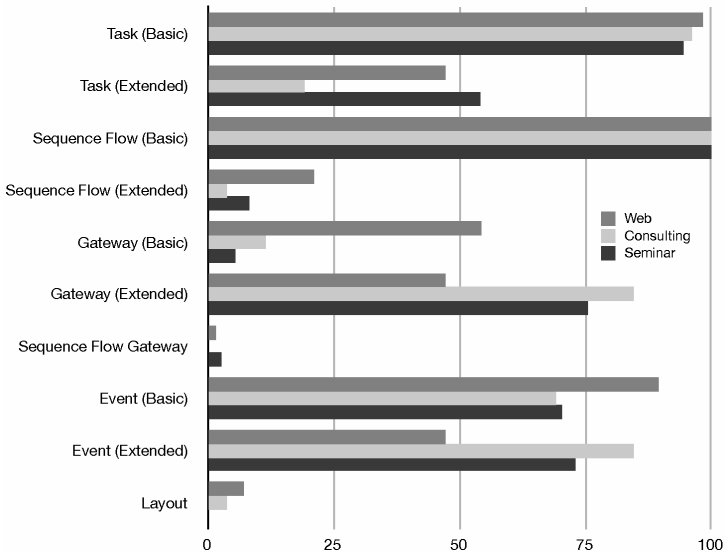

Fig. 4. Use of Core and Extended BPMN Constructs

- Decision Sequence Flow constructs are very rarely used. This would suggest that BPMN users prefer the explicit decision routing representation capacity of Gateways over the alternative, rather implicit way of annotating sequence flows.
- Basic Gateways are dominant on the web. However, neither consulting nor seminar models use them in large numbers. This suggests that formal training (as exercised through seminar courses or trained consultants) leads to the use of precise semantics for articulating process orchestration.
- Layout constructs are very rarely used. This suggests two things. First, language users often use tool functionality to annotate diagrams (e.g., meta-tags, free form tags, navigation capacity). Second, it may be worthwhile externalizing such constructs from a modeling language in order to reduce their complexity.

3.6 Complexity of BPMN Models

Previous studies on the usage of UML [5, 6] uncovered that the theoretical complexity of a language (as measured by the number of constructs originally specified) often considerably differs from the practical complexity (the number of constructs actually used in a model). We are interested in whether a similar situation exists in the case of BPMN. In other words, while the theoretical complexity of BPMN is standardized by its specification [1], we wanted to measure the practical complexity of BPMN (i.e., the vocabulary used in practice). To that end, we contrasted the semantic complexity of the BPMN models we obtained (i.e., the size of the models) with their syntactic complexity (i.e., the number of semantically different BPMN constructs used in these models). Fig. 5 illustrates the results of this analysis.

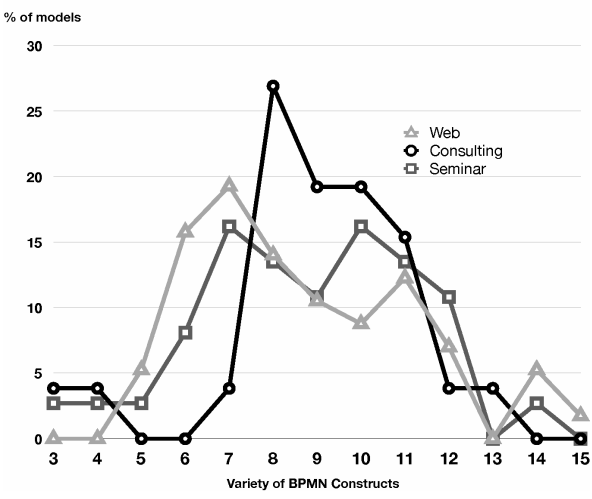

Fig. 5. Syntactic Complexity of BPMN Models

While the 50 BPMN constructs theoretically allow for 2^{50} permutations, the actual number of usable subsets is much smaller. All BPMN models obviously require the use of Tasks and Sequence Flow. Since the majority of models we observed used a BPMN vocabulary of between 6 and 12 constructs, the number of possible BPMN vocabulary subsets in practice is between $\binom{48}{4}$ =194,580 and $\binom{48}{10}$ = 6,540,715,896. Given that 9 constructs in our sample were used by fewer than two models we can exclude these from the search space and arrive at a theoretical range from $\binom{39}{10}$ = 82,251 to $\binom{39}{4}$ = 635,745,396. On average, we found the average number of semantically different BPMN constructs to be 9 (consulting), 8.78 (web), and 8.7 (seminar), respectively. However, while this finding indicates the size of the average BPMN vocabulary used in practice, it does not mean that every model with 9 BPMN constructs uses the exact same BPMN subset. In fact, a pair wise comparison of the 120 models revealed only 6 pairs of models that shared the same BPMN subset between each pair (i.e., there were 6 identical pairs of construct sets).

3.7 Variety of BPMN Subsets

In order to determine the variety of BPMN subsets, we computed the Hamming Distance [15] for each model vocabulary. Originally, the Hamming distance between two strings of equal length is the number of positions for which the corresponding symbols are different. In other words, it measures the minimum number of substitutions required to change one into the other. In the case of BPMN, we treated each model vocabulary as a 50-bit binary string, where a positive bit at position *i* signals the usage of BPMN construct [*i*]. The Hamming Distance between two model vocabularies then indicates the number of bits that differ between the two vocabularies, in other words the discrepancy between the BPMN constructs used in the creation of two models. The results are visualized in Fig. 6.

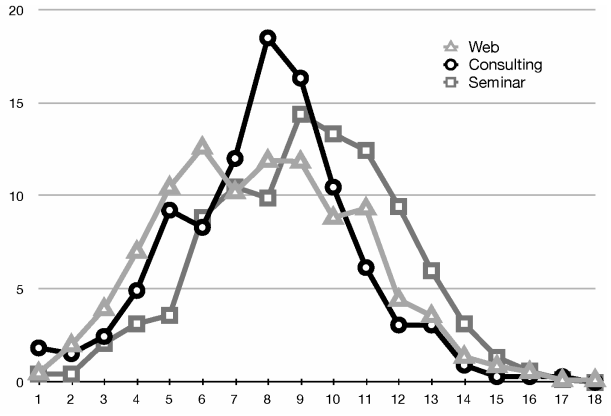

Fig. 6. Hamming Distance of BPMN Vocabularies

The average Hamming distance for the three subsets was 7.6 (web), 7.5 (consulting), and 8.8 (seminar), indicating a slightly more diverse use of BPMN constructs by novice modelers, whereas the web and consulting sets were slightly more homogeneous (but not by much). These metrics indicate that the average dissimilarity between two BPMN subsets is 7-8 constructs. A common scenario would be that one model uses 4 BPMN constructs that the other model does not exhibit and vice versa. As BPMN becomes more prevalent we plan on observing this metric over time, to see whether the commonly used vocabularies become more homogeneous over time. Annotating these BPMN subsets with context information (e.g., the process modeling purpose), in turn, could provide a starting point for deriving the most suitable BPMN subsets for a variety of application areas.

3.8 The Common Core of BPMN

Our evaluation thus far has focused on the individual elements and their grouping into core and extended constructs. However, one of our questions relates to the subset of BPMN constructs that are shared by different models. While we found six pairs of models that each share a complete set of constructs, there are subsets that are shared by more than two models. Figure Fig. 7 shows a Venn diagram of different BPMN construct combinations. The number in the corner of each grouping indicates the number of models that contained this specific subset of the language. We included combinations of constructs that were shared by more than 10 models.

The most apparent subset is the combination of Tasks and Sequence Flow – 97% of the models we analyzed shared this subset, and those that did not used a representation for tasks from the extended BPMN set (e.g., Subprocess). The addition of Start and End Events is the next most common subset – used by more than half of the models we analyzed. The following subsets show an interesting pattern: Either modelers focus on *process orchestration* through by adding gateways and their refinement to their models, or they focus on *process choreography* and add related organizational constructs, such as Pools and Lanes. While the addition of Pools leads to a subset that is common in nearly 30% of all models, the addition of Lanes halves this fraction.

Adding Basic Gateways or Parallel Gateways to the core set leads to a subset that is shared by 20% of all models. The popularity of the Data-based XOR Gateway and the Parallel Gateway construct indicate that they are a core element in many modeler vocabularies, even though the BPMN specification places them in the extended set of the language. The same situation holds for Message and Timer Events (both Start Events and Intermediate Events). While other event types were used very infrequently, these two event types were the most popular addition to the core modeling set in lieu of unspecified events.

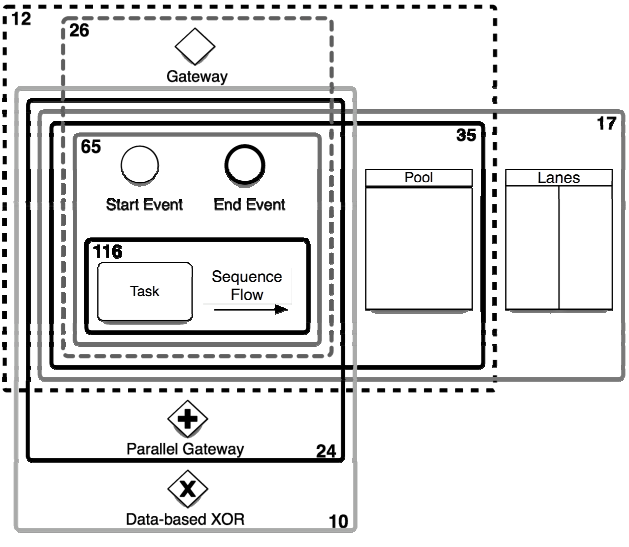

Fig. 7. Most popular BPMN Vocabulary Subsets

Overall, BPMN models appear to fall into two main sets (indicated in Fig. 7 by horizontal versus vertical grouping). The horizontal groups contain tasks, basic events plus constructs for separating organizational duties and responsibilities (Pools and Lanes). Consultants will use these types of models will most likely for organizational (re-)engineering and process improvement. The vertical groups add to this set of constructs refined constructs for specifying the exact control flow of processes (through various gateway types) as well as the exact event conditions pertaining to a process (i.e., various event construct types). This is not shown in Fig. 7 in the interest of clarity. Overall, this set of BPMN construct combinations can be expected to be favored by designers and analysts seeking to articulate the precise flow conditions, for instance, in the context of workflow engineering or process simulation rather than the organizational responsibilities (depicted by Lanes or Pools).

An interesting property of the BPMN subsets is their frequency distribution. The ranked frequency distribution again follows an exponential distribution, mirroring the behavior of individual BPMN constructs. This suggests that modelers use blocks or subsets of BPMN constructs in a similar fashion as they use individual constructs.

Combinations of BPMN constructs can thus be treated as metawords and be analyzed as such.

4 Contributions, Limitations, and Outlook

In this paper we studied the use of BPMN in actual process modeling practice. We obtained 126 (120 considered) BPMN models and used a wide range of statistical techniques to shed light onto the practical complexity afforded by the use of BPMN. Our paper makes a key contribution to the growing area of process modeling by reflecting on empirical data about the use of a rising industry standard. The most important finding is that the complexity of BPMN in practice differs considerably from its theoretical complexity. This, in turn, suggests that future research should take this distinction into account when considering BPMN's expressive power, complexity or other features or characteristics. Our study shows that the frequency of BPMN constructs follows an exponential distribution, both at the elementary level and the subset level. This means that the practical use of a formal modeling language shows similarities to the use of natural language, and suggests that linguistic techniques can be applied to better understand the formation and use of languages in conceptual modeling overall. We see an opportunity for replicating our study with other standardized modeling approaches (e.g., UML) to obtain further evidence for this conjecture.

Our findings have major implications, both for language developers and the organizational ecosystems in which modeling languages are used. Our findings point to some areas of concern in current language standardization practices, which appear to prefer language extensions (more expressive languages) to language revision (more lean languages). Our findings indicate that this may be to some extent contradictory to practical usage. Also, our findings motivate organizations to invest resources into *conventions management* in order to be able to manage and limit the complexity brought to bear by the languages employed for process modeling.

The presented research findings have to be contextualized in light of some *limitations*. First, the source of empirical evidence is limited to three sets of data sources and 126 BPMN models overall. We also did not consider any longitudinal data (e.g., the evolution of BPMN models through various iterations). However, we made an effort to collect data from multiple application areas and to consider these in our analysis. While we grouped the models by origin, we did not have sufficient information about the model content to analyze the models based on their intended use. We performed a hierarchical cluster analysis on the models themselves, but did not identify significant clusters. While this supports the random nature of our sample, it contradicts one of our expectations – that there is a clear differentiation between BPMN models depending on their intended use.

In future research, we will continue our data collection and extend it with more context-related information, e.g., for what purpose were the models created, what types of modelers created the models etc. This will allow us to triangulate our findings with contextual variables so as to arrive at informed opinions about BPMN usage across a wide range of application areas. In a related stream of research, we will apply a number of complexity metrics [e.g., 16] to the identified BPMN clusters to make a statement about how complex the frequently used BPMN constructs subsets are.

References

1. BPMI.org, OMG: Business Process Modeling Notation Specification. Final Adopted Specification. Object Management Group (2006), http://www.bpmn.org
2. Fowler, M.: UML Distilled: A Brief Guide To The Standard Object Modelling Language, 3rd edn. Addison-Wesley Longman, Boston, Massachusetts (2004)
3. Siau, K., Cao, Q.: Unified Modeling Language: A Complexity Analysis. Journal of Database Management 12, 26–34 (2001)
4. Rosemann, M., Recker, J., Indulska, M., Green, P.: A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 447–461. Springer, Heidelberg (2006)
5. Siau, K., Erickson, J., Lee, L.Y.: Theoretical vs. Practical Complexity: The Case of UML. Journal of Database Management 16, 40–57 (2005)
6. Kobryn, C.: UML 2001: A Standardization Odyssey. Communications of the ACM 42, 29–37 (1999)
7. Ouyang, C., Dumas, M., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Pattern-based Translation of BPMN Process Models to BPEL Web Services. International Journal of Web Services Research 5, 42–61 (2008)
8. Recker, J., Rosemann, M., Krogstie, J.: Ontology- versus Pattern-based Evaluation of Process Modeling Languages: A Comparison. Communications of the Association for Information Systems 20, 774–799 (2007)
9. Recker, J., Indulska, M., Rosemann, M., Green, P.: How Good is BPMN Really? Insights from Theory and Practice. In: Ljungberg, J., Andersson, M. (eds.) Proceedings of the 14th European Conference on Information Systems. Association for Information Systems, Goeteborg, Sweden, pp. 1582–1593 (2006)
10. zur Muehlen, M., Ho, D.T.-Y.: Service Process Innovation: A Case Study of BPMN in Practice. In: Sprague Jr., R.H. (ed.) Proceedings of the 41th Annual Hawaii International Conference on System Sciences, Waikoloa, Hawaii (2008)
11. Wahl, T., Sindre, G.: An Analytical Evaluation of BPMN Using a Semiotic Quality Framework. In: Siau, K. (ed.) Advanced Topics in Database Research, vol. 5, pp. 102–113. Idea Group, Hershey, Pennsylvania (2006)
12. Barabási, A.-L., Bonabeau, E.: Scale-Free Networks. Scientific American 288, 50–59 (2003)
13. Li, W.: Random Texts Exhibit Zipf's-Law-Like Word Frequency Distribution. IEEE Transactions on Information Theory 38, 1842–1845 (1992)
14. Zipf, G.K.: On the Dynamic Structure of Concert Programs. Journal of Abnormal and Social Psychology 41, 25–36 (1946)
15. Hamming, R.W.: Error Detecting and Error Correcting Codes. Bell System Technical Journal 26, 147–160 (1950)
16. Rossi, M., Brinkkemper, S.: Complexity Metrics for Systems Development Methods and Techniques. Information Systems 21, 209–227 (1996)