Measuring Method Complexity: The Case of the Business Process Modeling Notation

Marta Indulska[1], Michael zur Muehlen[2], Jan Recker[3]

[1] UQ Business School, The University of Queensland, St Lucia,
QLD 4072, Australia
[2] Howe School of Technology Management, Stevens Institute of Technology,
Hoboken, NJ, 07030, USA
[3] Faculty of Science and Technology, Queensland University of Technology,
Brisbane, QLD 4000, Australia
m.indulska@business.uq.edu.au
Michael.zurMuehlen@stevens.edu
j.recker@qut.edu.au

Abstract. Graphical models are used to depict various aspects of enterprise architectures. Process modeling methods in particular have evolved from simple flowcharts to newer methods that include a large number of diagramming symbols, which increases their perceived complexity. To date, two main approaches for the complexity evaluation of conceptual modeling techniques have been proposed: a set of metrics based on the meta-model of the technique, and an ontology-based approach based on a mapping of modeling constructs to a reference ontology. Existing related work in process modeling has concentrated on the ontological analysis of modeling methods. In this paper we complement the existing ontological analyses by developing a meta-model of the Business Process Modeling Notation (BPMN) and applying established meta-model based complexity metrics. Our research finds that – compared to modeling languages such as UML and the EPC – BPMN has very high levels of complexity. Furthermore, we see evidence that such complexity can be significantly reduced through the use of modeling conventions.

Keywords: BPMN, modeling complexity, complexity metrics, process modeling

1 Introduction

Analysts and designers frequently use graphical models of the business domain they are concerned with to document the requirements for and the design of information systems intended to support this domain. In recent years, the process-based specification of Information System requirements has become increasingly common. Process models are specified using dedicated process modeling methods - such as the Business Process Modeling Notation (BPMN) [13], for example. The need for accurate and current process documentation has emerged as a primary reason to engage in conceptual modeling activities [6] and is now considered a key instrument for the analysis and design of process-aware information systems [7], service-oriented architectures [9], and web services [10] alike. It is also instrumental in Business Process Management (BPM) initiatives and increasingly utilized for process documentation to meet various compliance management requirements.

The increasing popularity of process modeling in practice has led to a wide range of process modeling methods, which range from business modeling approaches (e.g. [19]) and methods initially used for software engineering (e.g. [11]), to methods that provide advanced concepts for simulation, analysis, and even code generation and process execution (e.g. [24]). A very recent addition to the list of available methods is BPMN [13] - a popular industry standard that is based on the synthesis and extension of earlier modeling methods. The evolution of process modeling methods has been accompanied by a notable increase in the expressiveness of the new methods [17], which in turn was accompanied by a notable increase in the complexity of these methods. The complexity of process modeling methods affects the ability of modelers to master the use of these methods, and the inherent complexity of the resulting business process diagrams affects the user's ability to construct domain using a process-centric approach. Mayer [12] suggests that the learning performance of an individual is impacted by their personal characteristics as well as the learning materials and presentation methods provided to them (see Figure 1). His findings show that subjects presented with a conceptual model (i.e., a presentation method) have a significantly increased recall of conceptual information and can solve more problems than subjects provided with just textual information. Method complexity is thus a significant issue because it can affect the learnability, ease of use and overall usage of a method [18], potentially preventing the longevity of the method.

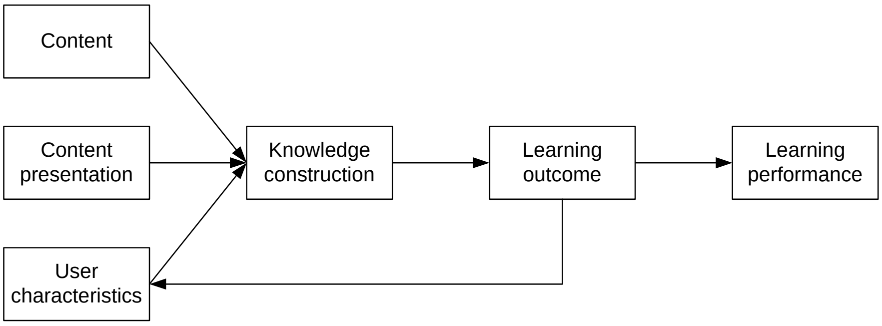

Fig. 1. A model for knowledge construction in process modeling. Adapted from [12].

Recent studies we have conducted indicate that the complexity of process modeling methods can negatively affect the usage of, and perceptions about, these methods (see, e.g. [15; 31; 32]). This situation can potentially impact the overall success of process modeling projects in organisations, which carry a substantial cost. At the same time industrial experience shows that methods with a larger vocabulary (such as BPMN) are used more frequently than methods with a restricted vocabulary (such as Petri Nets). Our previous work has found that users voluntarily restrict their use of a given notation, but give little formal explanation for the specific modeling restrictions they apply (or the modeling style they adhere to). Our anecdotal observations suggest that users deliberately lower the complexity of a given method by introducing restrictive modeling conventions, in order to either make the method easier to apply or the resulting models easier to communicate. In this paper we study this phenomenon on a more formal basis, employing established measurement techniques to determine the complexity of different subsets of popular modeling languages.

To date, two approaches for complexity measurement have been proposed – a set of metrics based on the elements of a method's meta-model, and ontology-based assessments. Ontology-based assessments (e.g. [26]) have successfully been applied to the process modeling domain in general [17] and to BPMN in particular [15]. These studies found that BPMN provides elements that can represent many aspects of real-world. This suggests that BPMN is a very expressive method. However, ontological analyses do not provide an explicit measurement for the complexity of a method, which would indicate how difficult a method is to learn and/or to apply. In our research, we are concerned with examining the complexity of different variants of popular process modeling methods, and thus we need to complement the existing ontological analyses.

Our study investigates to what extent different subsets of BPMN vary in their complexity, and how they compare to other popular process modeling languages. To this end we complement the existing ontological analysis of process modeling languages through the application of the widely used meta-model based complexity metrics of Rossi and Brinkkemper [18]. Using these metrics we can determine the actual complexity of the BPMN specification, and investigate to what extent the two complexity measurement approaches are complementary or substitutive. In particular, we want to investigate whether modeling conventions typically used in organizations mitigate the complexity of BPMN through a selection of constructs with an overall lower complexity. To this end, we apply the Rossi and Brinkkemper [18] metrics to three sub-sets of BPMN obtained through previous case studies, and compare the complexity measurements against the full BPMN specification. We additionally apply the metrics to measure the complexity of the EPC process modeling method, and use the published complexity measures of UML Activity Diagrams, to compare BPMN complexity to those of these two popular process modeling approaches.

We proceed as follows. The next section presents the background on process modeling with BPMN, as well as discussion of the two established approaches for method complexity measurement. We then introduce the BPMN meta-model developed to facilitate our application of the Rossi and Brinkkemper [18] complexity metrics for our analysis. The following section presents the results of complexity measurements applied to the BPMN meta-model as well as to three selected construct subsets of BPMN. We conclude with a discussion of implications and an outlook on future work.

2 Background

2.1 Process Modeling and BPMN

Process modeling is widely used within organizations to communicate of the structure of and responsibility for end-to-end business processes, to deconstruct organizational complexity and to aid the analysis and design of process-aware information systems [1]. Graphical process models typically describe at least the activities of the process, events that may trigger the process or emanate from the process, and control flow logic that describes the dependencies between activities [4]. Additionally, process models may also include information regarding the required input and output of individual activities (the involved data), responsible resources for the performance of activities, and potentially other artifacts such as external stakeholders and performance metrics (e.g. [19]).

A wide range of process modeling methods has been introduced since the advent of flowcharts in the 1920s, and their proliferation has led to a call for standardization efforts [5]. The development of the Business Process Modeling Notation (BPMN) [13] is a response to this call for standardization. BPMN was initially developed by an industry consortium (BPMI.org) and later ratified by the Object Management Group. The members of the BPMN working group represent a wide range of process modeling tool vendors and consultants, but very few users. BPMN was designed to be applicable in many areas, from process documentation and improvement scenarios to technical applications of process modeling such as workflow engineering, simulation or web service composition.

The standardization process took six years, and resulted in a specification that differentiates between a set of core BPMN elements and an extended specialized set. The complete BPMN specification in its version 1.2 defines 54 graphical modeling constructs plus attributes, grouped into four basic categories of elements, *viz*., Flow Objects, Connecting Objects, Swimlanes and Artifacts. Flow Objects, such as events, activities and gateways, are the most basic elements used to create BPMN models. Connecting Objects are used to link 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). Artifacts may be added to a model where deemed appropriate in order to display further related information such as processed data or other comments.

2.2 Complexity Measurement in Process Modeling

Two main approaches to measuring method complexity have been proposed in the literature, *viz*. ontological analysis and meta-model based metrics. The approaches differ fundamentally in their focus [18]. Ontological analyses of modeling methods are based on the observation that, in their essence, models of information systems are essentially models of real-world systems and hence can be evaluated against ontologies of real-world domains. Ontology-based evaluations use as a basis a model of real-world concepts (a representational ontology) and involve a mapping between the model and the constructs of the modeling method [26]. The representational ontologies utilised in ontological analysis, such as the Bunge-Wand-Weber (BWW) representation model [25-27], specify the types of representational constructs that a modeling method should provide to completely represent a particular domain. An ontological analysis (e.g. [14; 29]) then contrasts the elements of a modeling method against the representational ontology constructs to identify the modeling method's deficiencies. The basic assumption of this approach is that evaluating a method's coverage of ontology elements provides an indication of the quality of a technique. In other words, this measure serves as a proxy of how well a method can represent the real world. The extent of coverage should be maximized, while a 1:1 mapping between the constructs of the method and the elements of the ontology should be maintained, i.e., the mapping (and thus the constructs) should be as unambiguous as possible. The indication of quality thus comes through two main measures – ontological completeness and ontological clarity [28], which are defined as follows:

- *Ontological Completeness* is indicated by the degree of construct deficit, i.e., the extent to which a modeling method covers completely the constructs proposed in the ontological model.
- *Ontological Clarity* is indicated by the degrees of construct overload, where one method construct covers several ontological constructs, construct redundancy, where one ontological construct maps to several method constructs, and construct excess, where method constructs exist that do not map to any ontological construct.

Ontological clarity then is a measurement of method complexity as it describes how much effort method users have to apply in creating models that are unambiguous and clear in their interpretation. However, this metric does not indicate how complex the vocabulary of a method is. For instance, in a UML class diagram the 'class' construct can be used to model static information, objects that can be changed, and active components of a system, all with just one type of modeling element. Ontological completeness, on the other hand, is a measurement of the ability of the modelling method to completely represent a domain in question.

The meta-model based metrics, on the other hand, focus squarely on the structural properties of a modeling method. They utilise conceptual models of a modeling method, which capture information about the concepts, representation forms and uses of the method [20]. To enable the use of such meta-models for method analysis and comparison, Rossi and Brinkkemper [18] developed a comprehensive set of complexity metrics that quantitatively establish a method's complexity and ease of use based on the number of concepts, and relationships between them, defined in a method meta-model. The premise in this approach is that the presence of more modeling elements increases the cognitive load for the modeler [3], thus making a method harder to learn, with more difficult rules to follow. The metrics are counts of objects, relationships, and properties, as well as the average connectivity of method elements and two overall metrics that position a method in a three-dimensional measurement cube. Overall, Rossi and

Brinkkemper [18] present seventeen metrics clustered in three tiers, *viz*., individual, aggregate and method-level metrics. This set of metrics has widely been used to evaluate object-oriented methods (e.g. [21]), and we apply this set to process modeling methods, beginning with BPMN.

From a first glimpse, the two complexity measurement approaches appear complementary. Surprisingly, previous work on the assessment of process modeling methods has mostly used the ontological analysis approach. Indeed, researchers have shown that BPMN suffers from a number of shortcomings that impact the clarity of the method. For more information on the results and the details of ontological analysis in general we refer the reader to the works of [15; 17]. In this paper, we distinguish our work by exploring the application of the Rossi and Brinkkemper [18] metrics for the process modeling domain, an area to which the metrics have not yet been adopted, and exploring the complexity of BPMN and some of its commonly used subsets of modelling constructs.

3 Research Approach

The application of the Rossi and Brinkkemper [18] metrics requires the use of a BPMN meta-model to facilitate the calculation of BPMN complexity. Since such a model is not part of the BPMN 1.2 specification, we have developed such a meta-model to support our calculations. We base the meta-model development on BPMN version 1.2 [13]. The development follows a full review and analysis of the BPMN specification, supported by our experience in modeling real-world business processes with BPMN. The meta-model is designed as an UML class diagram. The overview aspect of the meta-model is presented in Fig. 2. It differentiates abstract BPMN concepts (such as message recipient and flow objects) from their concrete, graphical instantiations (such as activity, message, event).

The overall BPMN meta-model consists of nearly 100 distinct elements, which are mainly connected through inheritance relationships. For instance, the method element Sub-Process inherits properties from the elements BPMN Element, Graphical Element, Flow Object, and Activity. BPMN Element, Graphical Element, and Flow Object are abstract concepts, as they do not have a corresponding graphical representation, whereas Sub-Process has a graphical icon associated with it.

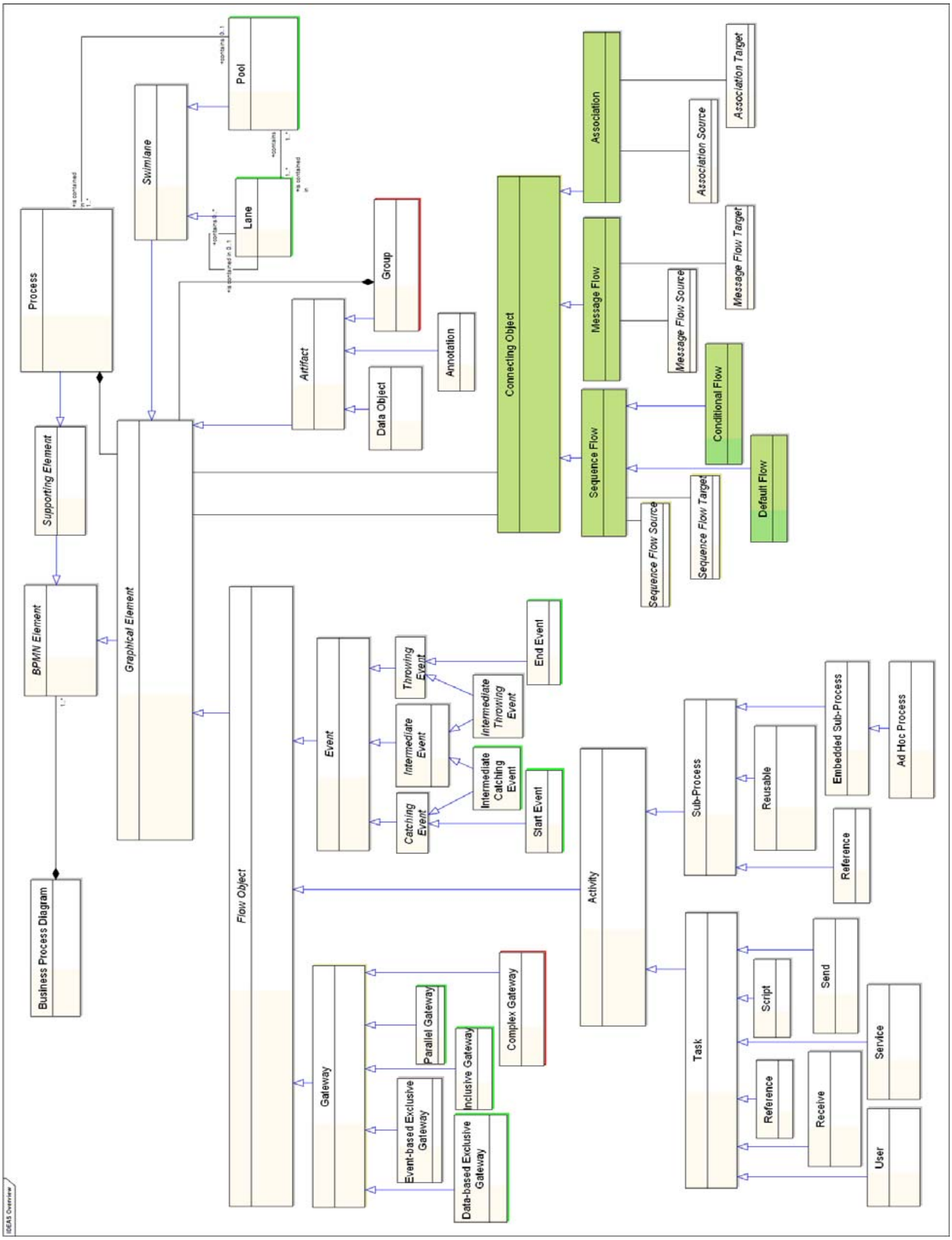

Fig. 2. BPMN 1.2 Meta-Model

3.1 Complexity Metrics

The Rossi and Brinkkemper [18] method complexity metrics facilitate the analysis of the complexity of a method based on the number of constructs that a potential user of the method would need to learn. The metrics in their essence are based on calculations of the counts of object types, relationship types and property types of a method. These fundamental counts allow the derivation of the average number of properties per object type, average number of properties per relationship type, and the average number of relationship types that can be linked with a particular object type in a given method. All of which indicate the complexity of of describing relationship types or object types in a given method. These metrics, in turn, form the basis for the calculation of the total conceptual complexity of a method, which can then be used as a benchmark for comparison of conceptual complexity of different methods [18].

For brevity, we omit from this paper the details of the mathematical formulas and reasoning behind the calculations. Instead, we refer the reader to the original Rossi and Brinkkemper [18] work for full details of the approach.

3.2 Application of Complexity Metrics

We proceed by applying the complexity metrics to the developed BPMN meta-model. In applying the metrics, we calculate the number of objects, relationships and properties in the developed BPMN meta-model. We then calculate the average number of properties per object and the average number of properties per relationship type, to finally arrive at the aggregated measure of complexity. These calculations are represented in column headings in Table 1. To understand the effect of different method subsets on the complexity metrics we apply the metrics to five different sets of the BPMN method overall. First, we consider the complete BPMN meta-model as per the BPMN specification [13]. This approach allows us to gauge the theoretical complexity [22] of BPMN. Second, we consider only the concrete BPMN set (as per the meta-model), i.e., the set of graphical instantiations only. This approach allows us to compare the complexity of the full BPMN specification, including all of its constructs and attributes, with the complexity that can be encountered in purely graphical BPMN models. Third, we consider a set of BPMN elements used in a reference project in industry. To this end, we refer to the subset of BPMN constructs used in a case study of service process innovation at a North-American automotive enterprise [30; 32]. Within this subset we again differentiate between the full set of meta-model concepts and the set of concrete graphical concepts only. Third, we consider the set of the 12 most frequently used BPMN elements found in practice. We base this analysis on a published empirical study of 120 real-world BPMN models reported in [31]. Again, we divide the concepts into a full and a concrete set and perform the complexity calculations accordingly. Finally, we consider a deliberately restricted version of BPMN selected for use by the U.S. Department of Defense [23], again comparing complexity of the full and concrete set of concepts.

This approach allows us to explore BPMN complexity on three levels. First, we are able to contrast the complexity of the full BPMN specification, against the complexity of the set of just the BPMN graphical constructs. Second, we are able to contrast the complexity of the full BPMN specification against its practical complexity, i.e., the complexity of BPMN as actually used by practitioners. Last, we are able to compare the complexity of BPMN to that of other popular process modeling methods. To facilitate the latter, we consider two popular process modeling methods, *viz.* EPC and UML Activity Diagrams. We apply the Rossi and Brinkkemper [18] metrics to a published EPC meta-model [2] and base the comparison of complexity of UML on published UML complexity calculations [22].

4 Results

Table 1 shows a summary of the results of the application of the Rossi and Brinkkemper [18] metrics to the sets of BPMN constructs outlined above. For benchmark purposes, we also include our calculations of EPC complexity and the published complexity metrics of UML Activity Diagrams [21]. These two methods are chosen because they are used in similar areas as BPMN.

Table 1: Calculated Complexity Measures

Method	*Objects (Obj)*	*Relationships (Rel)*	*Properties (Prop)*	*Prop/Obj*	*Prop/Rel*	*Total Complexity*
BPMN $^{\text{FULL}}$	90	6	143	1.52	1.33	169.07
BPMN $^{\text{CONCRETE}}$	57	6	74	1.19	1.33	93.60
Case Study $^{\text{FULL}}$	36	5	81	2.11	1.4	88.78
Case Study $^{\text{CONCRETE}}$	23	5	43	1.593	1.4	49.02
DoD $^{\text{FULL}}$	59	4	112	1.83	1.5	126.65
DoD $^{\text{CONCRETE}}$	30	4	45	1.36	1.5	54.23
Frequent Use $^{\text{FULL}}$	21	4	59	2.65	1.5	62.75
Frequent Use $^{\text{CONCRETE}}$	8	4	25	2.44	1	26.55
EPC $^{\text{FULL}}$	15	5	11	.79	0	19.26
EPC $^{\text{CONCRETE}}$	1	5	8	.73	0	9.49
UML Activity diagrams	8	5	6	.75	.2	11.18

The results of the calculations of the complexity metrics provide interesting insights. Notably, the complexity of the full BPMN meta-model is significantly higher than the previously calculated complexities of various other modeling methods (such as Activity Diagrams) as presented in [21] and also significantly higher than our calculated complexity of EPC. Yet, despite the high complexity BPMN has enjoyed significant uptake in industry, indicating that complexity is not necessarily an inhibitor to method uptake or usage. This result is in line with the findings of Eriksson and Siau [8] who found that users continue to use UML Class Diagrams, despite their relatively high complexity of 26.40 (which is still much lower than that of BPMN). This finding would suggest that in practice there may be a requirement for the expressiveness of BPMN, even at the expense of increased complexity.

Our consideration of three additional subsets of BPMN – one used by the U.S. Department of Defense [23], one from a published case study [30], and one from a study of the most frequently used BPMN constructs (based on an analysis of 120 models) [31], and – shows that the careful selection of subsets of a modeling method can significantly reduce its complexity. The three different sets are referred to in Table 1 as 'Case Study', 'DoD' and 'Frequent Use' respectively. When considering the set of BPMN constructs from the process innovation case study, the complexity level drops by almost 50% (from 169.07 to 88.78) as compared to the complexity of the full BPMN meta-model. Similarly, consideration of the DoD set of constructs also shows that the BPMN complexity level drops by approximately 25% (from 169.07 to 126.65) of that of the full BPMN specification. If we consider the set of the most frequently used BPMN constructs, based on the analysis of 120 real-world models, the complexity is further reduced to 62.75 (by over 60% as compared to the full BPMN specification). The finding indicates that BPMN users, consciously or not, take active steps to reduce modeling complexity. Anecdotal evidence suggests that a common way of such reduction is through development and enforcement of modeling conventions within organizations.

Furthermore, our findings show that BPMN exhibits a large amount of method complexity due to method constructs and constraints that are not graphically rendered. This situation is indicated by the differences in complexity measures of the full BPMN meta-model and its concrete subset (i.e., 169.07 vs. 93.60 in the full BPMN meta-model), which also persist in calculations of complexity in the three considered subsets of BPMN (see Table 1). This finding indicates that the underlying rules and constraints of the method are a significant source of complexity. In turn, this finding leads us to a number of propositions.

First, users might ignore the underlying rules, thus reducing the complexity of the method in practical use. If this is the case, we should see BPMN models that violate its underlying grammatical rules. We have seen anecdotal evidence that users apply certain constructs outside of their designated scope, but this aspect warrants further study.

Second, not every BPMN construct is equally popular. It may be that the bulk of the hidden complexity is contained in elements that are rarely used in practice (see the published classification in [31]). In this case, the hidden complexity is invisible to the users since they do not have to pay attention to these rules.

Third, the complexity of the method may be a result of the consensus-based approach to method design that was used during the development of BPMN, and which is in sharp contrast to the largely autonomous development efforts of early object-oriented modeling methods that have comparatively low complexity levels [21].

Our analysis makes clear that the design of BPMN is far from elegant. In practice, however, BPMN has found widespread acceptance and wide-ranging tool support. In order to determine how to make BPMN easier to use, an interesting pathway would be to assess the ontological coverage (as per ontological analysis) of each added modeling element under consideration of its added complexity. As part of the work presented in this paper we are currently undertaking this tradeoff study.

Initial investigation into the complementarity of ontological analyses and structural complexity metrics provides fruitful insights. Ontological analysis of BPMN and EPC in particular indicate that BPMN has a significantly higher representation ability than EPC [16]. EPC are assessed as having a 62% degree of construct deficiency with respect to the BWW representation ontology, compared to a 35% degree of deficit in BPMN [16]. This result, combined with our complexity metric calculations, indicates that while BPMN has a higher representation power, it is significantly more complex than EPC. Indications of ontological clarity of BPMN and EPC also indicate potential complementarity between the two methods of complexity measurement. The 'lack of clarity to coverage' of BPMN and EPC is 58% and 62% respectively [16]. This result indicates that despite BPMN's good representation ability, its complexity (lack of ontological clarity) reduces its average 'lack of clarity to coverage' ratio to one that is close to EPC. The relationships between the two types of complexity measurement require further investigation. As part of the work presented in this paper we are currently exploring this area, as well as undertaking a tradeoff study that explores the reduction of structural complexity guided by ontologically-informed removal of constructs.

5 Conclusions

This paper is a first contribution towards the assessment of process modeling methods with a set of conceptual modeling method complexity metrics that have previously only been used in the object-oriented domain. We specifically focus our work on the practically predominant Business Process Modeling Notation and show that it has very high levels of complexity. In particular, we show that the theoretical complexity of BPMN is significantly higher than the complexity of its graphical set of constructs. Overall, we find that the complexity of BPMN is significantly higher than that of other popular process modeling methods – namely EPC and UML Activity Diagrams. Moreover, it is clear from the additional BPMN construct sets considered that users, purposefully or not, decrease the complexity of BPMN through the use of subsets of the method.

We argue that such analysis of complexity is complementary to the established method of ontological analysis of method representation capability and show preliminary insights that indicate the relationship between the two complexity measurement methods should be further explored. Ontological analysis of modeling methods concentrates purely on the methods' ability to represent real world content (i.e., its usefulness), whereas the complexity metrics focus on the usability of the method (i.e., its ease of use). By looking at the complexity metrics, we can observe how reduction of elements can reduce complexity. In turn, the ontological analysis can guide whether the removal of such elements would impact the method's ability to represent the real world (i.e., its usefulness).

Our future work in this area will take a number of directions. First, we will consider the published ontological analysis of BPMN and further investigate the relationship between ontological analysis and meta-model based complexity measurement. Second, we will analyze the complexity of the less frequently used BPMN constructs to determine whether much of the BPMN complexity stems from constructs that are largely unused in practice. Third, we will investigate existing BPMN models in order to determine the source of frequent modeling errors.

References

[1] Bandara, W., Gable, G. G., & Rosemann, M. (2005). Factors and Measures of Business Process Modelling: Model Building Through a Multiple Case Study. *European Journal of Information Systems*, *14*(4), 347-360.

[2] Becker, J., Delfmann, P., Falk, T., & Knackstedt, R. (2003). Multiperspektivische ereignisgesteuerte Prozessketten. In M. Nüttgens, & F. J. Rump (Eds.), *Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des 2. GI-Workshop EPK 2003* (pp. 45-60). Bamberg:.

[3] Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction*, *8*(4), 293-332.

[4] Curtis, B., Kellner, M. I., & Over, J. (1992). Process Modeling. *Communications of the ACM*, *35*(9), 75-90.

[5] Davenport, T. H. (2005). The Coming Commoditization of Processes. *Harvard Business Review*, *83*(6), 100-108.

[6] Davies, I., Green, P., Rosemann, M., Indulska, M., & Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, *58*(3), 358-380.

[7] Dumas, M., van der Aalst, W. M. P., & ter Hofstede, A. H. M. (Eds.). (2005). *Process Aware Information Systems: Bridging People and Software Through Process Technology*. Hoboken, New Jersey: John Wiley & Sons.

[8] Erickson, J., & Siau, K. (2007). Theoretical and Practical Complexity of Modeling Methods. *Communications of the ACM*, *50*(8), 46-51.

[9] Erl, T. (2005). Service-Oriented Architecture: Concepts, Technology, and Design.

[10] Ferris, C., & Farrell, J. (2003). What are Web services? *Communications of the ACM*, *46*(6), 31.

[11] Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* Addison-Wesley Professional.

[12] Mayer, R. E. (1989). Models for understanding. *Review of Educational Research*, *59*(1), 43-64.

[13] OMG. (2009). Business Process Modeling Notation. *V1.2 OMG Available Specification*, *formal/2009-01-03*, 316.

[14] Opdahl, A. L., & Henderson-Sellers, B. (2002). Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model. *Software Systems Modeling*, *1*, 43-67.

[15] Recker, J., Indulska, M., Rosemann, M., & Green, P. (2006). *How Good is BPMN Really? Insights from Theory and Practice*. Paper presented at the 14th European Conference on Information Systems, Goeteborg, Sweden.

[16] Recker, J., Rosemann, M., Indulska, M., & Green, P. (2009). Business Process Modeling - A Comparative Analysis. *Journal of the Association for Information Systems*, *In Press*.

[17] Rosemann, M., Recker, J., indulska, M., & Green, P. (2006). *A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars*. Paper presented at the Proceedings of the Conference on Advanced Information Systems Engineering (CAiSE 2006), Grand-Duchy of Luxembourg.

[18] Rossi, M., & Brinkkemper, S. (1996). Complexity Metrics for Systems Development Methods and Techniques. *Information Systems*, *21*(2), 209-227.

[19] Scheer, A.-W. (2000). *ARIS-Business Process Modeling.* Springer.

[20] Siau, K., & Rossi, M. (2008). Evaluation Techniques for Systems Analysis and Design Modelling Methods - A Review And Comparative Analysis. *Information Systems Journal*, *In Press*.

[21] Siau, K., & Cao, Q. (2001). Unified Modeling Language: A Complexity Analysis. *Journal of Database Management*, *12*(1), 26-34.

[22] Siau, K., Erickson, J., & Lee, L. Y. (2005). Theoretical vs. Practical Complexity: The Case of UML. *Journal of Database Management*, *16*(3), 40-57.

[23] U.S. Department of Defense. (2009). Enterprise Architecture based on Design Primitives and Patterns. Guidelines for the Design of Business Process Models (DoDAF OV-6c) using BPMN. *Technical Report OSD/DCMO CTO/CA BMA, February 10, 2009,* 47pp.

[24] van der Aalst, W. M. P., & ter Hofstede, A. H. M. (2005). YAWL: yet another workflow language. *Information Systems*, *30*(4), 245-275.

[25] Wand, Y., & Weber, R. (1990). An Ontological Model of an Information System. *IEEE Transactions on Software Engineering*, *16*(11), 1282-1292.

[26] Wand, Y., & Weber, R. (1993). On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. *Journal of Information Systems*, *3*(4), 217-237.

[27] Wand, Y., & Weber, R. (1995). On the Deep Structure of Information Systems. *Information Systems Journal*, *5*(3), 203-223.

[28] Weber, R. (1997). *Ontological Foundations of Information Systems.* Melbourne, Australia: Coopers & Lybrand and the Accounting Association of Australia and New Zealand.

[29] Zhang, H., Kishore, R., & Ramesh, R. (2007). Semantics of the MibML Conceptual Modeling Grammar: An Ontological Analysis Using the Bunge-Wand-Weber Framework. *Journal of Database Management*, *18*(1), 1-19.

[30] zur Muehlen, M., & Ho, D. T.-Y. H. (2008). *Service Process Innovation: A Case Study of BPMN in Practice*. Paper presented at the *41st Annual Hawaii International Conference on System Sciences*, Waikoloa, Hawaii.

[31] zur Muehlen, M., & Recker, J. (2008). How much language is enough? Theoretical and Practical Use of the Business Process Management Notation. *Conference on Advanced Information Systems Engineering (CAiSE '08).*

[32] zur Muehlen, M., Recker, J., & Indulska, M. (2007). *Sometimes Less is More: Are Process Modeling Languages Overly Complex?* Paper presented at the 3rd International Workshop on Vocabularies, Ontologies, and Rules for the Enterprise (VORTE 2007), Annapolis, MD.