# Execution Semantics of Process Models with Data

Maximilian König[1]

[1]Hasso Plattner Institute, Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam

Abstract
Data is one of today's most important currencies. Thus, maintaining an overview of its creation, usage, and manipulation within an organization is of utmost importance. While this fact has been recognized in the Business Process Management (BPM) community in general, its subfield of process modeling has not attributed attention to that for a long time. Extensive research has been conducted on the logical and temporal order of process steps, also called the control flow. While doing so, the impact of data, e.g., that certain tasks require specific information to be executed, has been largely neglected. Even with the extension of some process modeling languages to incorporate data concepts, formal semantics of these concepts, that enable automated analysis and enactment, are often not present or underspecified. Therefore, this paper motivates the definition of a new, more holistic semantics for data concepts in BPMN. This semantics is then to be used as a foundation to adapt existing and define novel verification, compliance, and consistency checking methods regarding the data and data flow of processes.

**Keywords**
Data in Processes, BPMN, Translational Semantics, Petri Nets

## 1. Introduction

For an organization to thrive in a fast-paced, competitive environment, it must constantly monitor its business processes and the data required for and manipulated by their execution. That requires a thorough documentation of these processes and the information involved in them. A means to achieve that are process models describing the required tasks and their data pre- and postconditions. However, due to the size and number of processes in place within an organization, manually managing these process models and ensuring their consistency, correctness, and compliance, especially when undergoing change, is very challenging and error-prone. Hence, automation is desirable, but requires thorough formalization of the involved concepts, i.e., the definition of a concise execution semantics. Looking at the Business Process Modeling and Notation (BPMN) [1], the most widely adopted process modeling language [2], multiple approaches exist defining a concise semantics for its control flow concepts. However, there is currently no formalization that covers all of its data concepts, which prevents automated analyses of process data flow. Therefore, this paper motivates the introduction of a new execution semantics defined through the mapping of BPMN data concepts to Petri net constructs that can serve as a basis for the automated analysis of the data flow of process models.

*ZEUS2023: 15th Central European Workshop on Services and their Composition, February 16–17, 2023, Hannover, Germany*
maximilian.koenig@student.hpi.de (M. König)
0000-0002-2244-1179 (M. König)


S. Böhm and D. Lübke (Eds.): 15th ZEUS Workshop, ZEUS 2023, Hannover, Germany, 16–17 February 2023, published at http://ceur-ws.org

In the remainder of this paper, BPMN data concepts are briefly introduced and related work on formal semantics for BPMN is discussed before the new mapping and its potential application areas are outlined.

## 2. Background and Related Work

Since version 2.0, the BPMN standard [1] includes a number of semantically meaningful concepts regarding the data flow in process models. A minimal example containing a subset of them is shown in Figure 1. *Data objects* (document shapes) are an abstract representation of the data used in a process. They can have *data states* (denoted in squared brackets) assigned to them. Connections to activities indicate that the respective data object is required as input or produced as output. In- and outgoing data objects can be clustered in *input and output sets* (I/O sets, denoted using, e.g., *I1* and *O1*) representing sufficient data enablement or termination conditions. For example, a *Claim* in state *[received]* alone is sufficient to start *Assess Insurance Claim*. Input output specifications (I/O specs, denoted using the BPMN annotation element) define, which data may be produced by an activity given it started with a certain input set. In the example, if an *Insuree Scam History* data object exists (*I2*), the risk assessment always produces a *Risk* in state *[high]* and a *Second Assessment* in state *required* (*O2*). Finally, the BPMN standard employs the rule that only a single data object instance may exist per process instance, unless specified as being multi-instance.

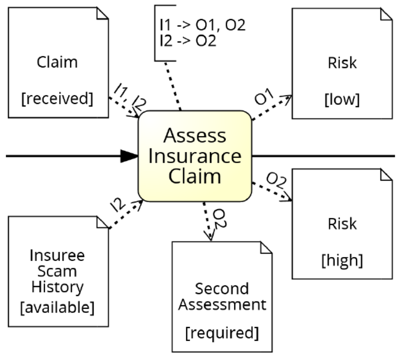


Figure 1: BPMN data concepts

While the BPMN standard provides a textual description of the semantics of these concepts, a thorough formalization is currently missing. In the past, research has been conducted on defining such a formalization through translational semantics, i.e., creating a mapping of BPMN concepts to those of another, well-formalized modeling language [3]. Target languages comprise, inter alia, process algebras [4], WS-BPEL [5], Event-B [6], and, most prominently, Petri nets [7, 8, 9, 10, 11, 12, 13]. However, most approaches solely consider the control flow and disregard the data perspective. Even the ones considering the data perspective only define semantics for subsets of the above-mentioned data concepts. Table 1 shows an overview of a selection of works introducing a translational semantics of BPMN to (colored) Petri nets, and the data-related BPMN concepts they cover. In addition to the concepts introduced in the standard, the table includes *Data Locking*, which represents a mechanism to prevent concurrent write access and inconsistent read access while the data may concurrently be modified as proposed in [9]. The overview shows that, while all aspects are covered by at least one approach, no mapping yet covers all aspects.

Next to the extension of an activity-centric process modeling language with data concepts, related work proposed a variety of different approaches to the representation of data in business processes. Two frameworks have been introduced to evaluate and compare approaches regarding their incorporation of data in business processes [14, 15]. The covered spectrum ranges from extensions to activity-centric approaches such as BPMN [9] through case management, e.g., fragment-based Case Management [16], to artifact-centric, e.g., [17], and object-centric approaches, e.g., [18]. However, most of these approaches require the learning of a new process modeling language, which constitutes a major hurdle to their adoption. In contrast, BPMN already is the de facto industry standard. Hence, providing the precision the BPMN standard currently lacks regarding its data concepts may lower the threshold to incorporate the data perspective in existing BPMN process models and will therefore be the focus of this work.

**Table 1**
Coverage of BPMN data concepts in related works mapping BPMN to Petri nets. Each aspect is either covered (✓), partially covered ([✓]), or not covered (–). SI abbreviates *Single-Instance enforcement*.

| Publication | Data Objects | Data States | I/O Sets | I/O Spec | Data Locking | SI |
|---|---|---|---|---|---|---|
| Meyer [9] | ✓ | [✓] | – | – | ✓ | ✓ |
| Awad et al. [10] | ✓ | ✓ | – | – | – | – |
| Stackelberg et al. [11] | ✓ | – | ✓ | – | – | ✓ |
| Ramadan et al. [13] | ✓ | [✓] | ✓ | ✓ | – | – |
| Dechsupa et al. [12] | ✓ | – | – | – | – | – |

## 3. Contribution

To close the gap of a missing holistic semantics for data concepts in BPMN, as outlined in the preceding section, this paper proposes the introduction of a new translational semantics for BPMN using Petri nets. The mapping should cover data objects, data states, input and output sets, input and output specifications, multi-instance and single-instance behavior, a data locking mechanism, and the combination of these concepts with tasks, subprocesses, and (boundary) events adhering to the restrictions provided in the BPMN standard. To allow a focus on the data flow, the mapping rules for BPMN control flow elements introduced by Dijkman et al. [7] will be extended. The result should be a formal algorithm based on which properties of the derived nets such as the enforcement of a single data object instance per process instance, data locking as defined by Meyer [9], and the correct translation of BPMN concepts can be proven.

The Petri nets resulting from the algorithm's application will then serve as a foundation for a set of analyses. First, a notion of data flow soundness should be defined, ensuring the absence of data deadlocks and proper termination of the processes. Next to that, different categories of data anomalies have been defined in literature [11, 19, 20]. Their application as well as a potential extension of them based on the introduced formalism would allow for a more thorough analysis of processes' data flow. For example, lost data (data objects being written multiple times without being read in between), missing data (required data in a certain process state that is not available), and redundant data (data that is written but never read) can thereby be detected. In addition, a recent literature review revealed a lack of data compliance checking methods for processes [21]. In that context, it would be interesting to see which of their constraint patterns concerning data will be detectable in the derived Petri nets.

## 4. Conclusion

This position paper motivates the need for a new translational semantics thoroughly covering BPMN's data concepts. The lack of such a semantics hinders the capitalization on process models' capability to provide and maintain an overview of the data flow within an organization. The mapping of BPMN concepts to Petri nets should then serve as the foundation for automated data flow analyses regarding correctness, consistency with other processes, and compliance to the process environment's regulations.

## References

[1] OMG, Business Process Model and Notation (BPMN), Version 2.0.2, Technical Report, Object Management Group, 2014. https://www.omg.org/spec/BPMN/2.0.2/PDF.

[2] M. Dumas, D. Pfahl, Modeling software processes using BPMN: When and when not?, in: Managing Software Process Evolution, Springer, 2016, pp. 165–183.

[3] A. H. M. ter Hofstede, H. A. Proper, How to Formalize it?: Formalization Principles for Information System Development Methods, Inf. Softw. Technol. 40 (1998) 519–540. doi:10.1016/S0950-5849(98)00078-0.

[4] P. Y. H. Wong, J. Gibbons, A Process Semantics for BPMN, in: S. Liu, T. S. E. Maibaum, K. Araki (Eds.), Formal Methods and Software Engineering, 10th International Conference on Formal Engineering Methods, ICFEM 2008, Kitakyushu-City, Japan, October 27-31, 2008. Proceedings, volume 5256 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 355–374. doi:10.1007/978-3-540-88194-0_22.

[5] C. Ouyang, M. Dumas, A. H. M. ter Hofstede, W. M. P. van der Aalst, From BPMN Process Models to BPEL Web Services, in: 2006 IEEE International Conference on Web Services (ICWS 2006), 18-22 September 2006, Chicago, Illinois, USA, IEEE Computer Society, 2006, pp. 285–292. doi:10.1109/ICWS.2006.67.

[6] J. W. Bryans, W. Wei, Formal analysis of BPMN models using event-b, in: S. Kowalewski, M. Roveri (Eds.), Formal Methods for Industrial Critical Systems - 15th International Workshop, FMICS 2010, Antwerp, Belgium, September 20-21, 2010. Proceedings, volume 6371 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 33–49. doi:10.1007/978-3-642-15898-8_3.

[7] R. M. Dijkman, M. Dumas, C. Ouyang, Semantics and Analysis of Business Process Models in BPMN, Inf. Softw. Technol. 50 (2008) 1281–1294. doi:10.1016/j.infsof.2008.02.006.

[8] I. Raedts, M. Petkovic, Y. S. Usenko, J. M. E. M. van der Werf, J. F. Groote, L. J. Somers, Transformation of BPMN Models for Behaviour Analysis, in: J. C. Augusto, J. Barjis, U. Ultes-Nitsche (Eds.), Modelling, Simulation, Verification and Validation of Enterprise Information Systems, Proceedings of the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems, MSVVEIS-2007, In conjunction with ICEIS 2007, Funchal, Madeira, Portugal, June 2007, INSTICC PRESS, 2007, pp. 126–137.

[9] A. Meyer, Data Perspective in Business Process Management, PhD Thesis, Universität Potsdam, 2015.

[10] A. Awad, G. Decker, N. Lohmann, Diagnosing and Repairing Data Anomalies in Process Models, in: S. Rinderle-Ma, S. W. Sadiq, F. Leymann (Eds.), Business Process Management Workshops, BPM 2009 International Workshops, Ulm, Germany, September 7, 2009. Revised Papers, volume 43 of *Lecture Notes in Business Information Processing*, Springer, 2009, pp. 5–16. doi:10.1007/978-3-642-12186-9_2.

[11] S. von Stackelberg, S. Putze, J. Mülle, K. Böhm, Detecting Data-Flow Errors in BPMN 2.0, Open Journal of Information Systems (OJIS) 1 (2014) 1–19. URL: http://nbn-resolving.de/urn:nbn:de:101:1-2017052611934.

[12] C. Dechsupa, W. Vatanawood, A. Thongtak, Hierarchical Verification for the BPMN Design Model Using State Space Analysis, IEEE Access 7 (2019) 16795–16815. doi:10.1109/ACCESS.2019.2892958.

[13] M. Ramadan, H. G. Elmongui, R. Hassan, BPMN formalisation using coloured petri nets, in: Proceedings of the 2nd GSTF annual international conference on software engineering & applications (SEA 2011), 2011, pp. 83–90.

[14] S. Steinau, A. Marrella, K. Andrews, F. Leotta, M. Mecella, M. Reichert, DALEC: a framework for the systematic evaluation of data-centric approaches to process management software, Softw. Syst. Model. 18 (2019) 2679–2716. URL: https://doi.org/10.1007/s10270-018-0695-0. doi:10.1007/s10270-018-0695-0.

[15] A. Meyer, S. Smirnov, M. Weske, Data in Business Processes, EMISA Forum 31 (2011) 5–31.

[16] M. Hewelt, M. Weske, A Hybrid Approach for Flexible Case Modeling and Execution, in: M. L. Rosa, P. Loos, O. Pastor (Eds.), Business Process Management Forum - BPM Forum 2016, Rio de Janeiro, Brazil, September 18-22, 2016, Proceedings, volume 260 of *Lecture Notes in Business Information Processing*, Springer, 2016, pp. 38–54. doi:10.1007/978-3-319-45468-9_3.

[17] R. Hull, E. Damaggio, F. Fournier, M. Gupta, F. F. T. H. III, S. Hobson, M. H. Linehan, S. Maradugu, A. Nigam, P. Sukaviriya, R. Vaculín, Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles, in: M. Bravetti, T. Bultan (Eds.), Web Services and Formal Methods - 7th International Workshop, WS-FM 2010, Hoboken, NJ, USA, September 16-17, 2010. Revised Selected Papers, volume 6551 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 1–24. doi:10.1007/978-3-642-19589-1_1.

[18] V. Künzle, M. Reichert, PHILharmonicFlows: towards a framework for object-aware process management, J. Softw. Maintenance Res. Pract. 23 (2011) 205–244. doi:10.1002/smr.524.

[19] S. W. Sadiq, M. E. Orlowska, W. Sadiq, C. Foulger, Data Flow and Validation in Workflow Modelling, in: K. Schewe, H. E. Williams (Eds.), Database Technologies 2004, Proceedings of the Fifteenth Australasian Database Conference, ADC 2004, Dunedin, New Zealand, 18-22 January 2004, volume 27 of *CRPIT*, Australian Computer Society, 2004, pp. 207–214. URL: http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV27Sadiq.html.

[20] S. X. Sun, J. L. Zhao, J. F. N. Jr., O. R. L. Sheng, Formulating the Data-Flow Perspective for Business Process Management, Inf. Syst. Res. 17 (2006) 374–391. doi:10.1287/isre.1060.0105.

[21] T. Voglhofer, S. Rinderle-Ma, Collection and Elicitation of Business Process Compliance Patterns with Focus on Data Aspects, Bus. Inf. Syst. Eng. 62 (2020) 361–377. doi:10.1007/s12599-019-00594-3.