A Petri net-based View on the Business Process Life-Cycle

Agnes Koschmider*,[a], Andreas Oberweis[a], Wolffried Stucky[a]

[a] Institute of Applied Informatics and Formal Description Methods, KIT, 76137 Karlsruhe, Germany

Abstract. *During the last 30 years Petri nets have shown a continuous popularity and durability as process modelling language. While some process modelling languages were crossed with others or even disappeared, Petri nets are continuously used as modelling language addressing various purposes in the context of business processes. In this paper, we refer to the success of Petri nets and describe business process modelling extensions as well as approaches for process modelling, simulation, execution and evaluation relying on Petri nets. The variety of Petri net-based extensions shows that Petri nets can be adapted to changing requirements for which these extensions, modifications or variants have been proposed.*

Keywords. Petri Nets • Modelling Language • Modelling Variants

1 Introduction

Process modelling is an important activity that supports all phases of the business process life-cycle, from the early identification of the need for a business process through subsequent phases of design, execution and monitoring. Business process models support the common understanding of the kind of business activities underlying a business process, including their mutual relationships. Plenty of modelling languages exist to design business processes with different representations (e.g., text vs. graphic) or using a different design paradigm (e.g., imperative vs. declarative). The decision in favour of a specific modelling language depends on the requirements that the chosen modelling language should meet (Oberweis 1996). Possible requirements might be the appropriateness and adequacy of modelling constructs (i. e., expressiveness) or an easy learnability of the language (i. e., simplicity). Certainly, the decision in favour of a process modelling language might also be made based upon its practical acceptance, case studies or the cognitive effectiveness of the visual notations (Figl and Recker 2016). Currently, the Business Process Model and Notation (BPMN) is mostly used for process modelling although the language has shortcomings with respect to a comprehensive support of the whole business process lifecycle (Radloff et al. 2015). The Event-Driven Process Chain (Nüttgens and Rump 2002), which was for quite long time the standard language for business process modelling, has lost popularity. Retrospectively, a continuous durability as a process modelling language can be attested only to Petri nets (Reisig 2013). This certainly can be explained due to the *“mathematical analysis techniques allowing for analytical verification of many relevant properties of systems’ behaviour”* (Desel et al. 1998).

Since the doctoral thesis of Carl Adam Petri entitled "Communication with Automata" (Petri 1962) in 1962 there exist also numerous variants for Petri nets addressing various purposes in the context of business processes or system dynamics in general (e. g., different variants of high-level Petri nets). In this paper, we refer to the success of Petri nets and describe business process modelling extensions or approaches to process modelling, simulation, execution and monitoring relying on Petri nets. We will show that Petri nets are capable of expressing changing requirements for which

* Corresponding author.
E-mail. agnes.koschmider@kit.edu

these extensions or variants of Petri nets have been proposed.

The paper is structured as follows. Section 2 summarizes common languages for business process modelling and sketches their current status as modelling language. Section 3 presents exemplary extensions of Petri nets or approaches relying on Petri nets and classifies them according to the business process life-cycle. In each phase we also point to future directions of process modelling, execution and evaluation. The paper ends with a summary in Section 4.

2 Business Process Modelling Languages

The most common language to design business processes are the Business Process Model and Notation (BPMN), Event Driven Process Chains (EPC), Petri nets and the Unified Modelling Language (UML). Technical issues of business process models are rather tackled by orchestration and choreography languages such as XML Process Definition Language (XPDL), the Web Services Business Process Execution Language (WS-BPEL), ebXML or WS-CDL.

BPMN is a graphical modelling language that does not only support business process modelling but also process implementation. Accordingly, the modelling language has primarily been designed with the intention to make the notation easy to understand for all user groups (e.g., analysts creating the first drafts or the technical developers responsible for the implementation). BPMN is divided into five basic categories: Flow Objects, Data, Connecting Objects, Swimlanes and Artefacts. Figure 1 shows a BPMN diagram with three different flow objects, which are events (circles), activities (rectangles) and gateways. The standardization by the OMG in 2005 has certainly led to the high popularity of BPMN and has contributed to BPMN becoming the de-facto standard for business process modelling.

EPC is a graphical, semi-formal modelling language aiming to provide a simple and intuitive way to visualize business process models. There are numerous proposals in the literature formalizing

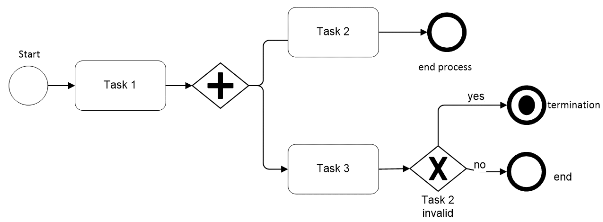

Figure 1: Example BPMN diagram

the EPC language (Kindler 2006; Mendling and Aalst 2006). EPC received a widespread adoption after their integration with the SAP reference model (Nüttgens and Rump 2002) and due to its part of the so-called ARIS house. The main constructs of an EPC are events, functions, and join operators (XOR, AND, OR). Numerous variants of the EPC can be found in the literature such as the extended EPC (eEPC) allowing to define organizational units and input and output parameters. Figure 2 visualizes an eEPC diagram.

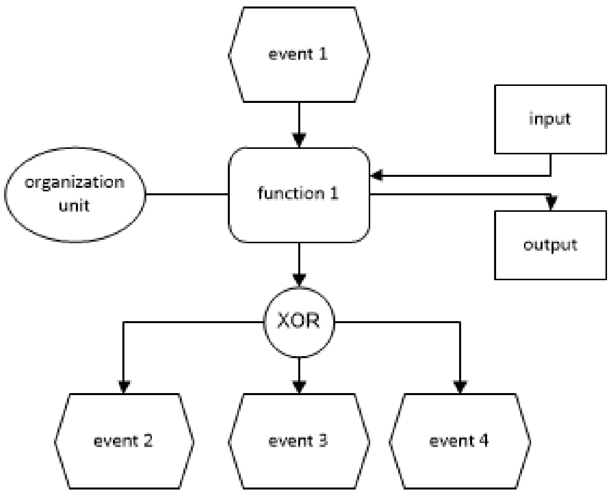

Figure 2: Example eEPC diagram

Due to the lack of standardization through an appropriate organization the popularity of EPCs recently significantly decreases (Karhof et al. 2016).

Petri nets are used in many ways today, for example to model, simulate, analyse and monitor business processes. In doing so, sequential, mutually exclusive and concurrent activities can be modelled with three modelling elements. Petri nets consist of places, which are represented as circles, and of transitions, which are visualized as rectangles, see Figure 3. Places and transitions are linked by arcs representing relationships between both elements (Reisig 2013). Places represent static aspects, such as documents, data or resources. Transitions are used to represent dynamic aspects (i. e., state transitions). Petri nets combine the advantages of a simple graphical representation of business processes with the formal semantics of the uses notation.

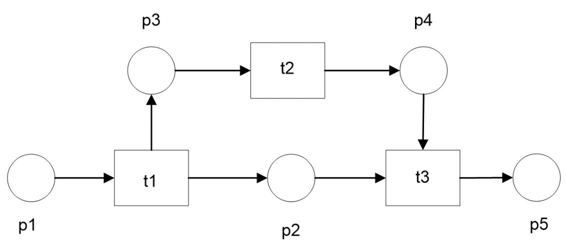

Figure 3: Example Petri net diagram

There have already been many discussions about the theoretical investigations of Petri nets in the academic sector and the practical needs of business process modelling languages in business applications (Desel et al. 1998). In spite of all these discussions Petri nets are the only language that "survived" the last 50 years of modelling and thus a continuous popularity can be definitely attested to Petri nets (Reisig 2013). One variant of Petri nets, which is capable to represent e-Business or web service based aspects are XML nets. XML nets are a variant of high-level Petri nets (Lenz and Oberweis 2003). They have formal semantics, graphical nature, and the strength in exchanging XML-based structured data. In XML nets markings of the places are given as XML documents and places are considered as containers for the documents. The flow of XML documents is defined through occurrences of transitions.

3 Extensions and Approaches relying on Petri nets

In the following we discuss extensions and approaches relying on Petri nets according to the business process lifecycle (Weske 2012) (see Figure 4). The *design & analysis* phase tackles the creation and validation of business process models. The *configuration* phase within the lifecycle is about technical realization of a process model, including setting specific parameters to tie implementation details to the higher-level model. The *enactment* phase is related to the monitoring of process models. In the business process lifecycle the *evaluation* phase addresses the evaluation of process models using business activity monitoring and process mining techniques.

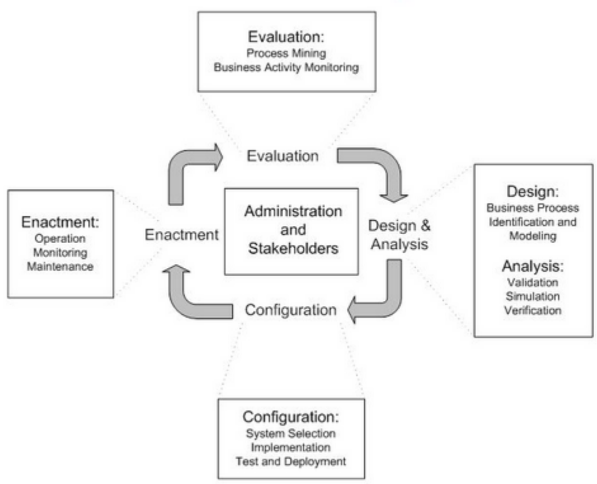

Figure 4: Business process lifecycle

Process mining refers to a set of techniques that analyze (mostly historical) event logs of IT systems in order to derive a model of the process (i. e., mostly a Petri net) that corresponds to these logs. For a process model to be created from such log files, the event log must (at least) store the sequence of events. Figure 5 shows a log file with two traces to which process mining algorithms were applied in order to derive a Petri net (Drescher et al. 2017). Consequently, one can observe that Petri nets also play an important role in the evolving field of process mining.

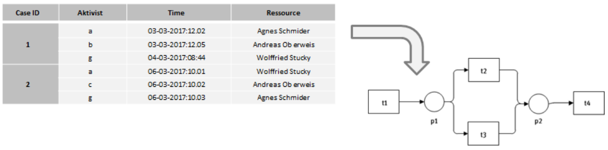

Figure 5: Process mining applied: generating a Petri net from an event log

The next section summarizes Petri net based approaches related to the design and analysis phase.

3.1 Extensions for the Design and Analysis Phase

There are many discussions regarding the modelling approach (i. e., imperative versus declarative modelling) (Fahland et al. 2009), the support of modellers and automatic creation of process models from log files (i. e., process mining). With respect to modelling methods and in particular the understandability of Petri net based models, research has been conducted on how novice modellers are constructing process models. In this area, it is well known from empirical studies that novice modellers struggle to create "good" process models since they tend to forget important model elements. One approach that supports modellers aims at reducing the entry barrier to process modelling by providing an on top layer to process models (Koschmider et al. 2015). It is a lightweight approach to modelling. Particularly, the approach presented in (Koschmider et al. 2015) discusses variables of how to best design diagrams consisting of graphical or textual elements The approach adds an abstract model level ("Layer 0") that can be represented both depictively (iconic) or descriptively (symbolic) with the possibility to seamlessly switch between them. The abstract layer should enable a quick and comprehensive view of the underlying (Petri net-based) process model and in addition should expose basic modelling capabilities. With this layer modelling should be accessible for a larger audience. From the viewer's perspective: both representations allow the process model's viewer to get a quick and comprehensive view of the underlying process model. If the viewer is further interested in the fine-grained representation of the process model he/she can navigate through the process model hierarchy. From the creator's (i. e., process modeller's) perspective: the concepts of this layer abstract from common process modelling languages, and thus, it is expected, that the creation of process models even for inexperienced persons is simplified. Such approaches on top of business process models may pave the way for the detailed specification of requirements and elicitation of further design options and choices. Figure 6 shows the approach allowing an abstraction and concretion from depictive and descriptive models to Petri nets.

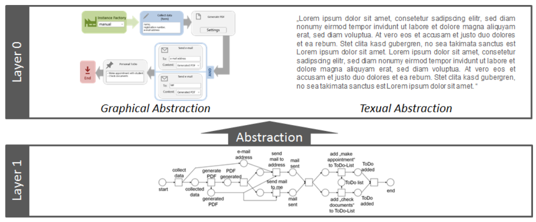

Figure 6: Two modalities of the abstraction layer ("Layer 0")

The design of business process models (for novices) can also be supported with a recommendation-based modelling support system (Koschmider et al. 2011). Such a system suggests how to finish appropriately a partially developed but incomplete business process model. The user can invoke the modelling support either via a query interface or s(he) can use a function automatically suggesting appropriate (Petri net based) process model fragments by unveiling the modelling intention of a user at process modelling time. Additionally, a social software-based extension was added to the recommendation-based modelling support in order to take into account a process builder's modelling context and the modelling history of a community of users (Koschmider et al. 2010). The system also suggests (Petri net based) process model parts to the user, that may help him achieving an individual modelling goal. Such features potentially improve the modelling process and, as such, the modelling outcome, that is, the quality of the process model. To support location- and device-independent modelling a Petri net based tool has been suggested (Alpers and Hellfeld 2016). The pne.fzi.de tool supports continuous work on a business process model through intelligent synchronization mechanisms, even when changing devices. The list of Petri net-based extensions addressing the design phase can be completed with approaches addressing security and privacy issues or sustainability aspects in business process models (Betz et al. 2017). Petri net-based approaches for simulation were suggested in (Li and Oberweis 2009) and (Eichhorn et al. 2009). The approach proposed in (Eichhorn et al. 2009) tackles a 3D support for graphical business process simulation. A risk-aware simulation based on XML nets is presented in (Betz et al. 2011). In the future, process modelling languages should be capable of providing links to bind connected devices enabling sensing, (re-)acting, collecting and exchanging data via various communication networks including the Internet. So far, the design of business processes was dominated by a so-called Model-Enact paradigm, i. e., the process has been depicted as a (graphical) process model that afterwards could be executed by a Business Process Management System (BPMS) (Janiesch et al. 2017). The Internet-of-Things paradigm requires approaches that tackle the whole stack from sensor data to event data to process activities. Physical devices sense their environment and produce events that have to be correlated in order to distil complex events. Complex events are then used as input for process mining algorithms. Benefits and challenges of the integration of IoT into business process modelling are discussed in (Janiesch et al. 2017) and (Soffer et al. 2018). The list of future research fields can be complemented by the blockchain technology and the combination of augmented or virtual reality and business process modelling. 3D technologies open up new possibilities for modelling business processes. They provide higher plasticity and eliminate some deficits of conventional 2D process modelling such as the limitation of the amount of information to be integrated into a process model in an understandable way (Betz et al. 2008). The next section sketches approaches for the business process configuration phase relying on Petri nets.

3.2 Languages for the Configuration Phase

Technical realization of a process model is tackled by orchestration and choreography languages. WS-BPEL is the widely-used standard for this phase. It lays the foundation for a process engine automating the execution of business processes specified as (web) services. BPEL provides an XML notation and semantics for describing the behaviour of business processes. XML net based approaches were suggested since XML nets allow formal verification for the composition of services (Che et al. 2009). Particularly, with XML nets messages can be modelled and manipulated as place tokens for message passing, and the labels in arcs can be used to model constraints for web service discovery and selection. A further XML net based extension for web service composition was presented in (Koschmider and Mevius 2005) and (Lenz and Oberweis 2004). The paper (Koschmider and Mevius 2005) introduces so-called Web service nets allowing a process model driven deduction of BPEL. Web service nets describe control flows of web services and derive the description of web services including behavioural, functional, and interface-based information. For this purpose, four different types of flow structures are distinguished. Figure 7 shows the sequence, the alternative, parallelism and synchronization of activities. For instance, by the use of alternative, one branch of a set of choices can be selected.

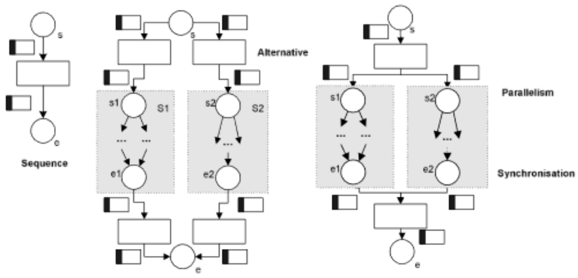

Figure 7: Flow structures of Web service nets

The execution of a business process might also depend on location constraints. A Petri net based method for defining location constraints in business process models was suggested in (Decker 2009), which is suitable for mobile business processes. Location constraints restrict the place where a process activity must be executed (positive restriction) or is not allowed to be executed (negative restriction). Figure 8 allows the execution of the activity "record order" only within the local instance "Berlin".

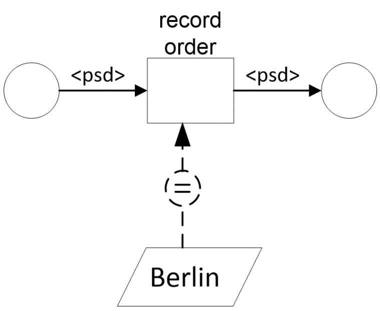

Figure 8: Location constraints of mobile business processes

One stream of future research for business process execution might stem from digital platforms, which require software architectures being capable of quickly adopting to changing conditions. In the future we will see research exploiting benefits of microservices for business process model execution. Microservice "...*is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms*" (Lewis and Fowler n.d.). In the context of Business Process Management (BPM) microservices could complement or even replace WS-BPEL for business process model execution. Contrary to WS-BPEL the characteristics of microservices allow executing highly distributed business process models without any centralized management. Particularly, process model activities or events could be specified as a microservice addressing its own technology stack. Challenges of a microservice-based business process execution are discussed in (Koschmider 2017b).

The next section sketches a Petri net-based approach for the enactment phase.

3.3 Languages for the Enactment Phase

One issue common to business process monitoring is the inability to link the business process schema to the information gathered during relevant phases (Mevius 2008). Therefore, (Lenz et al. 2005) presents an XML net based approach for process-oriented business performance management. In particular, the approach allows analysing potential exceptional states of performance indicators and matches counter measures in target-oriented simulation scenarios. Performance indicators are represented by XML schemas. Places that are typed by a performance indicator schema are interpreted as containers for the performance indicator value. The calculation of the performance indicator value is modelled by parts of an XML net and performance indicator violations can be described with alert or repair transitions. Figure 9 depicts the alert transition. An alert occurs if the ratio falls below a critical value.

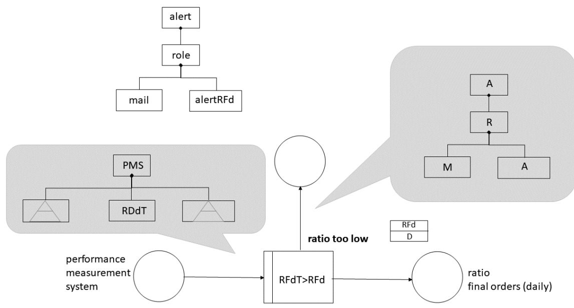

Figure 9: Alert transition to describe performance indicator violations

3.4 Languages for the Evaluation Phase

The evaluation phase is dominated by approaches for business activity monitoring (Friedenstab et al. 2012) and process mining (Aalst 2016).

The paper (Koschmider 2017a) describes a novel approach for clustering event traces by their behavioural similarity. Existing process mining algorithms reportedly generate spaghetti models from event logs of flexible processes, which are largely incomprehensible. The technique presented in (Koschmider 2017a) can compare the control-flow of event traces rather than deriving a unique process model encompassing all traces. The comparison is related to time, duration and further exogenous factors such as temperature. The clustering technique is based on two algorithms, which classify behaviours and allow us to identify behavioural shifts. The improved classification of event traces makes the technique superior to existing approaches, allowing more efficient identification and analysis behavioural deviations (Koschmider 2017a). In the future process mining approaches are required allowing to combine heterogeneous sensor and event types.

4 Conclusion

In this paper we sketched business process modelling approaches relying on Petri nets. The approaches were classified following the business process life-cycle. Petri net based approaches for the modelling phase support modellers location- and device-independently in the construction of process models. The support ranges from tools that add an on top layer, to process models or automatically suggest process model parts for the completion of a process model editing activity. Approaches for the configuration phase use XML nets, a variant of high-level Petri nets, for web service-based compositions and executions of process models. Location constraints restricting the execution can also be defined. An extension of XML nets was also presented for the enactment phase allowing to describe performance indicator violations. Our contribution to the evaluation phase is a novel clustering technique for event traces.

In the future, process modelling languages should be capable of providing links to IoT devices and thus considering IoT parameters in process execution and evaluation (Janiesch et al. 2017).

References

van der Aalst W. M. P. (2016) Process Mining - Data Science in Action, Second Edition. Springer

Alpers S., Hellfeld S. (2016) Werkzeug zur mobilen Modellierung von Geschäftsprozessen mittels Petri-Netzen. In: Betz S., Reimer U. (eds.) Modellierung 2016, 2.-4. März 2016, Karlsruhe - Workshopband. LNI Vol. 255. GI, pp. 147–152

Betz S., Eichhorn D., Hickl S., Klink S., Koschmider A., Li Y., Oberweis A., Trunko R. (2008) 3D Representation of Business Process Models. In: Loos P., Nüttgens M., Turowski K., Werth D. (eds.) Modellierung betrieblicher Informationssysteme. LNI Vol. 141. GI, pp. 73–87

Betz S., Fritsch A., Oberweis A. (2017) TracyML - A Modeling Language for Social Impacts of Product Life Cycles. In: Proceedings of the ER Forum 2017 and the ER 2017 Demo Track. CEUR Workshop Proceedings Vol. 1979. CEUR-WS.org, pp. 179–192

Betz S., Hickl S., Oberweis A. (2011) Risk-Aware Business Process Modeling and Simulation Using XML Nets. In: 13th IEEE Conference on Commerce and Enterprise Computing, CEC 2011. IEEE Computer Society, pp. 349–356

Che H., Li Y., Oberweis A., Stucky W. (2009) Web Service Composition Based on XML Nets. In: 42st Hawaii International International Conference on Systems Science (HICSS-42 2009)). IEEE Computer Society, pp. 1–10

Decker M. (2009) A Location-Aware Access Control Model for Mobile Workflow Systems. In: IJITWE 4(1), pp. 50–66

Desel J., Oberweis A., Reisig W., Rozenberg G. (1998) Petri Nets and Business Process Management. Dagstuhl seminars. Tech report.

Drescher A., Koschmider A., Oberweis A. (2017) Modellierung und Analyse von Geschäftsprozessen: Grundlagen und Übungsaufgaben mit Lösungen. DeGruyter Oldenbourg

Eichhorn D., Koschmider A., Li Y., Stürzel P., Oberweis A., Trunko R. (2009) 3D Support for Business Process Simulation. In: Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference, COMPSAC. IEEE Computer Society, pp. 73–80

Fahland D., Lübke D., Mendling J., Reijers H. A., Weber B., Weidlich M., Zugal S. (2009) Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In: 10th International Workshop on Enterprise, Business-Process and Information Systems Modeling, pp. 353–366

Figl K., Recker J. (2016) Exploring cognitive style and task-specific preferences for process representations. In: Requir. Eng. 21(1), pp. 63–85

Friedenstab J.-P., Janiesch C., Matzner M., Müller O. (2012) Extending BPMN for Business Activity Monitoring. In: 45th Hawaii International International Conference on Systems Science (HICSS-45), pp. 4158–4167

Janiesch C., Koschmider A., Mecella M., Weber B., Burattin A., Di Ciccio C., Gal A., Kannengiesser U., Mannhardt F., Mendling J., Oberweis A., Reichert M., Rinderle–Ma S., Song W., Su J., Torres V., Weidlich M., Weske M., Zhang L. (2017) The Internet-of-Things Meets Business Process Management: Mutual Benefits and Challenges. In: CoRR abs/1709.03628

Karhof A., Jannaber S., Riehle D. M., Thomas O., Delfmann P., Becker J. (2016) On the de-facto Standard of Event-driven Process Chains: Reviewing EPC Implementations in Process Modelling Tools. In: Modellierung 2016, pp. 77–92

Kindler E. (2006) On the semantics of EPCs: Resolving the vicious circle. In: Data Knowl. Eng. 56(1), pp. 23–40

Koschmider A. (2017a) Clustering Event Traces by Behavioral Similarity. In: ER 2017 Workshops. Lecture Notes in Computer Science Vol. 10651. Springer, pp. 36–42

Koschmider A. (2017b) Microservices-based Business Process Model Execution. In: 8th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA), pp. 158–161

Koschmider A., Caporale T., Fellmann M., Lehner J., Oberweis A. (2015) Business Process Modeling Support by Depictive and Descriptive Diagrams. In: 6th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA), pp. 31–44

Koschmider A., Hornung T., Oberweis A. (2011) Recommendation-based editor for business process modeling. In: Data Knowl. Eng. 70(6), pp. 483–503

Koschmider A., von Mevius M. (2005) A Petri Net Based Approach for Process Model Driven Deduction of BPEL Code. In: OTM 2005 Workshops. Lecture Notes in Computer Science Vol. 3762. Springer, pp. 495–505

Koschmider A., Song M., Reijers H. A. (2010) Social software for business process modeling. In: JIT 25(3), pp. 308–322

Lenz K., von Mevius M., Oberweis A. (2005) Process-Oriented Business Performance Management with Petri Nets. In: 2005 IEEE International Conference on e-Technology, e-Commerce, and e-Services. IEEE Computer Society, pp. 89–92

Lenz K., Oberweis A. (2003) Inter-organizational Business Process Management with XML Nets. In: Petri Net Technology for Communication-Based Systems - Advances in Petri Nets. Lecture Notes in Computer Science Vol. 2472. Springer, pp. 243–263

Lenz K., Oberweis A. (2004) Workflow Services: A Petri Net-Based Approach to Web Services. In: International Symposium on Leveraging Applications of Formal Methods, ISoLA 2004. Technical Report Vol. TR-2004-6, pp. 35–41

Lewis J., Fowler M. (n.d.) Microservices: a definition of this new architectural term.. https://martinfowler.com/articles/microservices.html

Li Y., Oberweis A. (2009) A Petri Net-Based Software Process Model for Developing Process-Oriented Information Systems. In: Proceedings of Information Systems Development ISD. Springer, pp. 27–39

Mendling J., van der Aalst W. M. P. (2006) Towards EPC Semantics based on State and Context. In: 5. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)", pp. 25–48

von Mevius M. (2008) A novel modeling language for tool-based business process engineering. In: Proceedings of the 2008 ACM Symposium on Applied Computing (SAC). ACM, pp. 590–591

Nüttgens M., Rump F. J. (2002) Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen. LNI Vol. 21. GI, pp. 64–77

Oberweis A. (1996) Modellierung und Ausführung von Workflows mit Petri-Netzen. Teubner-Reihe Wirtschaftsinformatik. Teubner

Petri C. A. (1962) Fundamentals of a Theory of Asynchronous Information Flow. PhD thesis, pp. 386–390

Radloff M., Schultz M., Nüttgens M. (2015) Extending different Business Process Modeling Languages with Domain Specific Concepts: The Case of Internal Controls in EPC and BPMN. In: Proceedings of the 6th Int. Workshop on Enterprise Modelling and Information Systems Architectures (EMISA), pp. 45–58

Reisig W. (2013) Understanding Petri Nets - Modeling Techniques, Analysis Methods, Case Studies. Springer

Soffer P., Hinze A., Koschmider A., Ziekow H., Ciccio C. D., Koldehofe B., Kopp O., Jacobsen A., Sürmeli J., Song W. (2018) From event streams to process models and back: Challenges and opportunities. In: Information Systems to appear

Weske M. (2012) Business Process Management - Concepts, Languages, Architectures, 2nd Edition. Springer