Volker Nissen, Dirk Stelzer, Steffen Straßburger und Daniel Fischer (Hrsg.):
Multikonferenz Wirtschaftsinformatik (MKWI) 2016
Technische Universität Ilmenau 09. - 11. März 2016
Ilmenau 2016
ISBN 978-3-86360-132-4 (Druckausgabe, Gesamtwerk)

Towards an EPC Standardization – A Literature Review on Exchange Formats for EPC Models

Dennis M. Riehle[1], Sven Jannaber[2], Arne Karhof[2], Patrick Delfmann[1], Oliver Thomas[2], and Jörg Becker[3]

[1] University of Koblenz-Landau, Institute for Information Systems Research, {riehle|delfmann}@uni-koblenz.de

[2] University of Osnabrück, Institute for Information Management and Information Systems, {sven.jannaber|arne.karhof|oliver.thomas}@uni-osnabrueck.de

[3] University of Münster, European Research Center for Information Systems, becker@ercis.uni-muenster.de

Abstract

Event-driven process chains (EPCs) have been used to create business process models from the early 90s and are still used in research and practice today. However, up to today, there is still no accepted standard for the EPC modelling language, which caused several different EPC dialects to appear and disappear over the last decades. To contribute to the development of a future standard for the EPC modelling language, we have conducted a systematic literature review on exchange formats for EPC models. In the paper at hand, we describe seven different exchange and storage formats which we have found in literature and compare their properties and capabilities. We find that the EPC Markup Language (EPML) has the greatest capabilities so far, as it supports the greatest variety of different EPC dialects. With our discussion, we contribute to the development of an EPC standard in the future, by describing how EPML can be adapted as a standardized exchange language for EPC models, which is part of a future standard for the EPC language.

1 Get Things Going – Shedding Light on EPC Exchange Formats

The event-driven process chain (EPC) has been one of the most dominant languages for business process modelling over the last decades and is well established in both research and practice (Knuppertz and Schnägelberger 2008; Fettke 2009; Houy et al. 2009). The maturity of EPCs manifests itself in numerous scientific publications covering a wide range of language aspects. In addition, the EPC has proven its relevance in practice by being implemented in most common business process modelling tools (Drawehn et al. 2014). However, in contrast to languages such as Business Process Model and Notation (BPMN), whose popularity has been significantly boosted by the existence of a defined standard (Recker et al. 2006), no systematic standardization effort for the EPC language has been made yet. Consequently, the absence of a standard hampers EPC usage and diffusion, especially due to difficulties in terms of interoperability, further development and overall acceptance (Ko et al. 2009; Fellmann et al. 2013).

Nowadays, most business process modelling languages have been standardized by respective institutions, for example the Object Management Group (OMG) or the International Organization for Standardization (ISO) (ISO/IEC 15909-1 2004; OMG 2011). The publication of those standards ensures international adherence to specified language components such as syntax, notation or exchange format. Although there have been attempts to provide detailed specifications for certain aspects of the EPCs language (e.g. Nüttgens and Rump 2002; Mendling 2007), an official standardization process guided by a standardization development organization (SDO) has not been initiated to date. In addition, due to the widespread nature of EPCs and extensive previous research in the field, there exists a multitude of contributions ranging from various syntactical or semantical propositions to multiple language extensions (Rittgen 2000; Fettke et al. 2010), ultimately resulting in a mosaic-like EPC landscape. Naturally, this situation renders standard-making a difficult challenge.

The paper at hand aims at addressing this issue by laying ground for a standardization of the EPC language. Since successful standardization endeavours rely heavily on agreement and consensus of a domain community (David and Greenstein 1990; Fomin et al. 2003), this paper tries to lift the fog of previous EPC research by proposing a state-of-the-art analysis on EPC exchange formats discussed in relevant literature. We believe that by focusing on exchange formats, valuable groundwork for further standardization efforts can be gained, since interfaces and thus the seamless exchange of data (e.g. models), is an essential driver for technical standardization (Fomin 2003; Mendling and Nüttgens 2006). Furthermore, exchange formats provide insight into other language components and underline what is of importance to software vendors and practitioners. A specific focus on literature has been chosen, since the foundation for successful EPC standard-making needs to closely consider both scientific and practical concerns. Hence, a research point of view has been applied in this paper, whereas a practical perspective is covered in greater detail in subsequent work. In order to investigate the state-of-the-art, a structured keyword-driven literature review is conducted. The results are classified according to predefined criteria and ultimately consolidated according to their underlying concept.

By reviewing relevant literature, several formats to exchange EPCs have been identified. The major contribution of this paper is two-fold: On the one hand, an overview of scientific literature in the field of EPC exchange formats is provided. On the other hand, the identified formats are evaluated against their feasibility to serve as an EPC exchange format that can be adopted for EPC standard-making.

The remainder of the paper is structured as follows. Section 2 introduces theoretical background on business process management with special focus on the EPC. In addition, a brief overview over related efforts to enhance EPC modelling and research conformable to ours further motivates the strived objects. Subsequently, the applied research methodology is carried out in Section 3. In Section 4 the results of the literature review are presented and synthesized, followed by a discussion of results and an outlook on further work. The paper concludes with a summary of the gained insights.

2 Theoretical Background and Related Work

The first definition for EPCs emerged in the 1990s from a joint work of the Institute for Information Systems in Saarbrücken and SAP (Keller et al. 1992). The objective of the project has been to develop a definition of a business process language that would be able to document the SAP R/3 enterprise resource planning system (Melcher 2014). Due to its usability for reference modelling, the EPC evolved to a widely accepted and well-established business process modelling language in practice as well as in academic research (Mendling 2008).

In order to obtain a comprehensive view on EPC exchange formats, it is necessary not just to consider the initial publication, but also to explore the manifold contributions in the field of EPCs. Alongside several standardization approaches (e.g. Nüttgens and Rump 2002; Mendling 2007), many extensions for the EPC language have been proposed. The basic elements initially established by Keller et al. (1992) mainly consist of events, functions and logical connectors. *Events* can either describe post-conditions or pre-conditions in the business process and are presented as a hexagon. Accordingly, a *function* represents an activity and is able to alter these conditions. Functions are portrayed as a rounded rectangle. Finally, *logical connectors* are used to join or split the control flow. This can be done with AND, OR and XOR connectors. As those connector types are also widely established in many other business process modelling languages, we renounce the explanation of their semantics. For an extensive explanation, the reader may refer to e.g. Keller et al. (1992).

On the basis of these modelling concepts, many extensions have been developed and discussed in literature. The most widely known extension might be the eEPC (extended EPC), as it was also implemented in ARIS (Architecture of Integrated Information Systems) and is nowadays often synonymously used for EPC (Becker et al. 2009). Other contributions that have been considered in this research often extend the semantics of EPCs and mostly add new modelling possibilities. We identified configurable EPCs (C-EPC) which extend the basic EPC by two elements and try to capture commonalities between processes (La Rosa et al. 2011). Another extension is the Fuzzy EPCs by Thomas (Thomas 2009), who presents an approach to offset fuzziness that exists for some decisions in business process models. By doing so, the modelling possibilities have been enhanced (Thomas et al. 2002). Nüttgens and Zimmermann (1998) developed the object-oriented EPC (oEPC), which outsources functions and organizational units from the very control flow and rebind them on object classes. Events thereby are directly affiliated by the control flow with these object classes. A likely approach to enrich the basic EPC is semantically annotated EPCs (S-EPC). S-EPCs have been presented by the concept of an ontology and offer the possibility to annotate functions and other elements in an EPC. Due to that, explicit questions regarding the S-EPC model, like “which events triggers which functions”, can automatically be answered (Filipowska et al. 2009). Eventually, we identified Yet another EPC (yEPC) by Mendling et al. (2005a), an extension that enables standard EPCs the support of workflow patterns. Thereby the concepts of empty connectors, multiple instantiation and cancellation is introduced.

By considering not only the basic EPC, the list of relevant exchange formats could be enhanced while doing research. Another reason for the need of a profound fundament is the lack of existing research in this field. Sarshar et al. (2005) provide an overview of EPC extensions, but neglect exchange formats in their consideration. A similar study has already been conducted by Barborka et al. (2006), who give a short review on EPC exchange formats by comparing Microsoft Visio VDX-files with AML-files of the ARIS Toolset and EPML files. However, they did not conduct a systematic literature review, and their work may lack evolution of the last nine years.

3 Methodology

To review existing literature on EPC exchange formats, we have conducted a structured literature review as suggested by Webster and Watson (2002) and vom Brocke et al. (vom Brocke et al. 2013). Since exchange formats for EPC models might be named in various different ways (e.g. exchange format, file format, XML syntax etc.) and especially because an exchange format could also be defined implicitly by the implementation of a modelling tool, we have decided to keep the scope of our literature review rather wide. Our goal was to gather all papers which address the language EPC as such.

For our literature review, we have considered four different data sources. First, since the event-driven process chain was originally invented at the Institute for Information Systems in Saarbrücken (Keller et al. 1992), we have considered their working paper series consisting of 198 papers in total. Additionally, we have considered the EPC workshop proceedings from 2002 to 2009, leading to another 57 papers. Lastly, we have searched for literature on two different search engines, SpringerLink[1] and ScienceDirect[2]. As search terms, we have used "event-driven process chain", "event driven process chain" and "Ereignisgesteuerte Prozesskette" (each search term was entered separately), which lead to a total of 1.806 results on SpringerLink and 198 results on ScienceDirect.

As suggested by Webster and Watson, we the first took the titles of all 2.259 papers into account and discarded papers which we considered to be not of interest. For the remaining 315 papers, we evaluated the abstracts to get a better understanding of the papers content. Again, we discarded papers which did not cover our topic, so that 150 papers remained. We removed duplicates from this set of papers, finally resulting in 78 papers, which we considered relevant.

We conducted a forward and backward search, to include papers which we may have missed so far. While the backward search was done manually by looking at the references, the forward search was done with the use of Google Scholar[3], because neither SpringerLink nor ScienceDirect provide such a feature. The same principle of title, abstract and paper content was applied, leading to a final set of an additional 27 papers. Therefore, our literature review is based on 105 papers in total.

4 Overview of Existing EPC Exchange Formats

We found that many papers consider the language EPC and elements of that modelling language like events, functions etc. and the relationships between them, but rather few papers consider an exchange format for EPC models. In total, we have identified seven different exchange formats, of which six were developed in academic publication, while one is the file format of proprietary software, the "ARIS Toolset", developed by the IDS Scheer Company – a company of one of the EPC founders, August-Wilhelm Scheer.

In the next subsections, we will describe each exchange format and its capabilities shortly and will further discuss the information, which are summarized in the table above.

[1] http://link.springer.com/
[2] http://www.sciencedirect.com/
[3] https://scholar.google.de/

Name	Language	Type	EPC-specific	Meta-Data	Layout
AML	EPC	XML (Proprietary)	No	Yes	Yes
XML for EPC	EPC	XML	Yes	No	No
EPML	EPC, yEPC and C-EPC	XML (EPML)	Yes	Yes	Yes
Fuzzy-EPML	Fuzzy-EPC	XML (EPML)	Yes	Yes	Yes
oEPML	oEPC	XML (EPML)	Yes	Yes	Yes
GXL	EPC	XML (GXL)	No	No	No
sEPC	S-EPC	Ontology	No	Yes	No

Table 1: Comparison of EPC exchange formats

4.1 ARIS Markup Language

The ARIS Markup Language (AML) is the file format the ARIS Toolset uses when a model is exported to a file. AML is a proprietary file format, which is based on the eXtensible Markup Language (W3C 2008), shortly referred to as XML. The AML format is formally described by a Document Type Definition (DTD), which is a formal declaration of the syntax of an XML document (W3C 2008). The DTD for AML is available together with the user manual of an ARIS Toolset installation. In the following, we refer to ARIS Toolset 7.2, though newer versions are unlikely to be very different.

With the ARIS Toolset, modellers cannot only create EPC models, but conceptual models of different kinds, which can all be exported to AMF files. This makes the AML format a non-EPC-specific format, i.e. it was created to store different conceptual models and, therefore, is not optimized to structurally represent EPC models. For a meta-model of AML, the reader may refer to Barborka et al. (2006). Basically, in an AMF file, there are object definitions and object occurrences, which are connected by connection definitions and connection occurrences. Distinguishing between definitions and occurrences allows ARIS to identify elements which occur multiple times within a process as the same. For example the activity “check invoice” might occur twice in a process, referring both times to the same activity definition, therefore being identical.

Since early process models were mainly used to visualize and understand processes, and machine use of process models for business process analysis was only introduced later. Subsequently, legacy file formats as the ARIS Markup Language have a focus on visualization. The file structure of an AML file includes much information on how to visualize the EPC model on screen, i.e. it includes a full layout with size (width and height) and position (x and y) of all elements.

Elements in an AMF file do not have a type. Therefore, an immediate identification of an element as activity or event is not possible. Instead, elements have a symbol number, which refers to a number for an ARIS symbol that can be an event, a function or something else. Therefore, the elements, which can be placed within a model, are not restricted by the exchange format, because the XML file can refer to any type of element by an ID. This makes an automated processing of AMF files more complicated as if the elements had types in terms of XML elements.

All in all, the AML file format is rather a graphical representation of conceptual models than an exchange format for event-driven process chains. However, it should not be underestimated, as it was the first file format for EPC models on the market and is still widely supported by current BPM software (e.g. by Signavio[4] and by ARIS Cloud[5]).

4.2 XML for EPC

A first step towards an EPC-specific exchange format was done by Geissler and Krüger (2002), who defined an XML notation for EPC models. They also formally described their XML notation by a Document Type Definition (DTD). An XML file in the format defined by Geissler and Krüger can contain one or more, dependent or independent EPC models. EPC models consist of events, functions, connectors, organizational units, information objects and process markers.

Control flows are used on functions and connectors (but not on events) to define ingoing and outgoing neighbour nodes for these elements. Therefore, events are implicitly connected to functions. To find the neighbour nodes of an event programmatically, one would need to iterate all functions and check if the desired event is referenced in the control flow there.

Besides from simple names for individual processes and resources, which can be assigned to functions, the XML notation by Geissler and Krüger does not allow specification of any meta-data, like informational data of a process modelling editor. Moreover, this XML notation does not include any information on the layout of an EPC model, which makes it hard to display such a model graphically.

The XML notation by Geissler and Krüger is suited to structurally represent an EPC model for further machine processing. It is not useful, if the EPC model should be transferred from one modeller and one modelling environment to another, because of the missing information on visualization of the model.

4.3 EPC Markup Language

The Event-driven Process Chain Markup Language (EPML) was first suggested by Mendling and Nüttgens (2002). EPML has evolved over the years (Mendling and Nüttgens 2003; Mendling and Nüttgens 2004a; Mendling and Nüttgens 2006). In their work, it has also been demonstrated how AML files can be transformed to EPML files (Mendling and Nüttgens 2004b).

Similarly to AML, EPML is also based on XML, making it easy to process EPML files in many different tools and programming languages. However, EPML differs significantly from AML, as EPML was specifically designed for EPC models and is not capable of representing other conceptual models. Therefore, in EPML the nodes of the XML document directly refer to element types of EPC, e.g. events, function, process interfaces, roles, documents and connectors like AND, OR and XOR. Further, in EPML, connections between process elements are named arcs and are modelled as XML elements as well, which is an improvement compared to the XML notation of Geissler and Krüger, because connections can directly be accessed programmatically without the need to iterate over all elements.

To circumstance issues with EPC models not being able to represent state-based workflow patterns, Mendling et al. (2005a) extended the EPC with an empty connector and concepts for multiple instantiation and cancellation. This extension is mostly referred to as yEPC and with version 1.2, Mendling et al. (c.f. id.) added support for yEPC models in EPML.

[4] See http://www.signavio.com/
[5] See http://www.ariscloud.com/

Since some processes have multiple variants, and modellers do not want to create a different process model for each variant of that process, configurative EPC models were introduced (Recker et al. 2005; Mendling et al. 2005b), mostly referred to as C-EPC models. With C-EPCs changing slightly over the years, the C-EPC as presented by La Rosa et al. (2011) was used to create EPML 2.0, which is capable of representing EPC, yEPC and C-EPC models.

Another approach to EPML was done by Thomas and Dollmann (Thomas and Dollmann 2008; Thomas 2009), who discussed fuzzy process engineering, which allows modelling imprecise decision making in EPC models. Thomas also suggested a modification of EPML to add fuzzy attributes to the file format. However, the EPML version by Thomas has not yet been integrated into EPML 2.0, meaning that fuzzy EPML currently is a standalone file format, even though it has many commonalities with EPML 2.0.

Similarly, Hogrebe et al. (2009) defined a version of EPML which is capable of storing oEPC models. Their approach is an extension of EPML 1.2 and has not yet been integrated into EPML 2.0 as well, making it yet another standalone file format.

The above-mentioned EPML formats do all include a variety of information on the graphical representation of EPC models. While the specification of such a graphical representation is not mandatory in EPML, all elements can be stored with their x and y position on the screen. This enables the reconstruction of the EPC model in different modelling tools, ensuring that the model looks the same in every different modelling environment.

All in all, EPML includes the greatest varieties of EPC dialects of all exchange formats discussed in this paper and represents an open, XML-based approach to exchange EPC models between different modellers and different modelling tools.

4.4 GXL

Winter and Simon (2006) suggested using the Graph Exchange Language (GXL) to exchange business process models. While it might be generally beneficial to regard process models as graphs, e.g. for business process analysis, such a storage format is rather impracticable to exchange EPC models between different modelling tools.

The GXL allows nodes and edges to be stored within a GXL file. Therefore, all process elements like events and activities are converted to nodes and all arcs between process elements are converted to edges. In order to keep information on which node is an event and which a function, additional string-based type declarations are added to each node. While this has the advantage of being able to store any kind of element types in the GXL file (i.e. configurable C-EPC nodes could be added, though not implicitly handled by the authors), it bears the danger of naming conflicts, because element types are not handled by the format. Different vendors might name different elements in different ways, leading to confusion for humans and probably errors for machine interpretation.

Additionally, the GXL format does not include any information on how the EPC model should look like in a modelling tool, so there is no information on a layout included. This makes the GXL format impractical for exchanging EPC models between modellers or modelling environments.

4.5 sEPC Ontology

Semantically annotated EPCs (sEPC) were introduced by Filipowska et al. (2009), who apply ontology concepts to EPC models. To exchange such annotated models, no known exchange format has been capable of including ontology concepts, which is why they stored the EPC model within the ontology. They included only the basic EPC elements defined by Keller et al. (1992), disregarding several recent EPC extensions.

Further, their ontology does not include any information on graphical representation, which is why the authors state that the EPC layout will be lost when an EPC is transformed from EPML to their ontology. For the exchange of EPC models, this exchange format seems to be inappropriate.

5 Discussion and Outlook

In the previous chapter, we have described seven different exchange formats for EPC models, of which three exchange formats were based on the EPC Markup Language suggested by Mendling and Nüttgens (2002). Still, these three EPML variants are incompatible with each other, as there is no integrated specification which includes all three dialects.

Comparing the seven exchange formats (c.f. Table 1), most of them are based on the extensible markup language (XML) – which is comprehensible, since XML documents are widely adopted for storing and processing structured data, and XML processing libraries are available for many programming languages and development environments, enabling tool developers to support XML processing without much effort. However, to allow tool developers and easy adoption of an EPC exchange format, such a format should not be proprietary, i.e. an exchange format should be well documented and the documentation should be freely available on the internet. This is not the case for the AMF format, which makes the AMF format an improper candidate for an exchange standard in an EPC standard.

Our review has further shown that there are approaches which are specific to the EPC language, while some approaches are capable of storing arbitrary conceptual models. While such flexible exchange formats may be beneficial in some cases, they have a severe disadvantage from a programmer's point of view. When such an EPC model is to be imported into a modelling environment, the type of content within that file cannot be guessed from the file format. Extensive parsing needs to be done, and maybe also assumptions need to be made, in order to import a model from a file which may contain arbitrary conceptual models. Additionally, since the type of elements is usually not limited to a certain set, such a file might contain elements, which are not part of an EPC standard and therefore are unknown to the modelling tool. Hence, an exchange format should be specifically designed for EPC models, to limit the file contents to a well-defined standard. Consequently, the GXL format and sEPC ontologies are improper candidates for an exchange standard as well.

Since an EPC model does not only consist of the structural process, but also the visual representation of that process, an exchange standard for EPC models should include a visual representation of an EPC. This ensures that an EPC can be displayed in the very same way across different modelling tools, which helps users to better identify and exchange their process models. Additionally, an EPC exchange format should include fields for meta-data, where modelling tools can store individual information, for example annotations for process model elements. Ideally, this information is stored in something similar to a key-based hash map, where arbitrary values can be stored under a specific identifier. If modelling tools adhere to a convention like prefixing all identifiers with a tool-specific vendor prefix, several different modelling tools could store meta-information in the same model without conflicting with each other.

Over the decades that EPCs have been discussed in literature, several variations of the EPC modelling language have been developed. Obviously, an exchange format for EPC models should include all EPC variants, which can be considered relevant in research or practice. While further research on the relevance of the individual EPC variants may be necessary, our review shows that the EPML family (including EPML, oEPML and Fuzzy EPML) already includes all EPC variants that we discovered with our systematic literature review. Since the EPML family fulfils the two previously mentioned criteria as well, namely including a visual process representation and meta-data storage, we suggest building an exchange standard for EPC models upon EPML.

Currently, the EPML exchange format is splintered, since there are three different incompatible EPML adoptions. Further research should consider these three EPML variants and aim at merging them back into a single EPML specification. Subsequently, an integrated version of EPML should be capable of storing EPC models, yEPC models, C-EPC models and depending on further relevance studies also Fuzzy EPC models and/or oEPC models. If all relevant variants of EPML are merged back together into a single EPML specification, EPML seems to be a solid base for an exchange format in an EPC standard in the future.

6 Conclusion

In this paper, we have conducted a structured literature review on event-driven process chains and have analysed 105 papers in terms of EPC exchange formats. We have found seven different exchange formats which were used in research and practice in the past. We have further analysed all seven exchange formats in terms of which EPC dialects they can handle, on what kind of storage type they are based, whether they were specifically designed for EPC-models or if they can be used for arbitrary conceptual models and finally, if these exchange formats can store meta-data and a visual representation of the process model.

Afterwards we shortly discussed characteristics, which a standardized exchange format for EPC models should fulfil and demonstrated to which degree the seven exchange formats known from literature meet these characteristics. We came to the conclusion that only the EPML formats meet all requirements, hence making the EPML exchange format a proper candidate for a future exchange standard. However, currently there are different adoptions of EPML, which should be united before EPML can be used as an exchange standard for EPC models.

With our paper, we have contributed to the ongoing discussion whether a standard for EPC is needed and how such a standard needs to look like. Since EPC standard-making is an ongoing part of our research, the paper at hand can be seen An EPC standard is necessary, especially in terms of exchange formats for EPC models, since this would greatly improve the user experience when EPC models need to be transferred from one user to another or from one modelling environment to another. However, as our review has shown, there are different exchange formats and different EPC dialects. Therefore, an EPC standard would greatly help to define an exchange format.

Acknowledgement:

The research presented in this paper is part of the SPEAK project and is funded by the Federal Ministry for Economic Affairs and Energy (BMWi) under grant number 01FS14030.

7 References

Barborka P, Helm L, Köldorfer G, et al (2006) Integration of EPC-related Tools with ProM. In: Nüttgens M, Rump FJ, Mendling J (eds) Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)." Vienna, Austria, pp 105–120

Becker J, Mathas C, Winkelmann A (2009) Geschäftsprozessmanagement, 7th edn. Springer, Berlin/Heidelberg, Germany

David PA, Greenstein S (1990) The Economics of Compatibility of Standards: An Introduction to Recent Research. Economics of Innovation and New Technology 1:3–41.

Drawehn J, Kochanowski M, Kötter F (2014) Business Process Management Tools 2014. Fraunhofer, Stuttgart, Germany

Fellmann M, Bittmann S, Karhof A, et al (2013) Do We Need a Standard for EPC Modelling? The State of Syntactic, Semantic and Pragmatic Quality. In: 5th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA). St. Gallen, Switzerland

Fettke P (2009) Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung (zfbf) 61:550–580.

Fettke P, Houy C, Loos P (2010) Zur Bedeutung von Gestaltungswissen für die gestaltungsorientierte Wirtschaftsinformatik − Ergänzende Überlegungen. Universität des Saarlandes

Filipowska A, Kaczmarek M, Stein S (2009) Semantically annotated EPC within semantic business process management. In: Lecture Notes in Business Information Processing. Milano, Italy, pp 486–497

Fomin V (2003) The Role of Standards in the Information Infrastructure Development. MISQ Special Issue Workshop 1993:302–313.

Fomin V, Keil T, Lyytinen K (2003) Theorizing about standardization: integrating fragments of process theory in light of telecommunication standardization wars. Sprouts: Working Papers on Information Environments, Systems and Organizations 3:29–60.

Geissler M, Krüger A (2002) Eine XML-Notation für Ereignisgesteuerte Prozessketten (EPK). In: Nüttgens M, Rump FJ (eds) Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)." Trier, Germany, pp 81–86

Hogrebe F, Nüttgens M, Kern H, Kühne S (2009) Towards an Integrated Product and Process Modelling : oEPC Markup Language (oEPML) for object-oriented Event-driven Process Chains (oEPC). In: Informatik 2009: Im Focus das Leben, Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI).

Houy C, Fettke P, Loos P (2009) Stilisierte Fakten der Ereignisgesteuerten Prozesskette – Anwendung einer Methode zur Theoriebildung in der Wirtschaftsinformatik. In: Nüttgens M, Rump FJ, Mendling J (eds) EPK 2009. 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitkreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK). Gesellschaft für Informatik. Bonn, Germany, pp 22–41

ISO/IEC 15909-1 (2004) Systems and sotware engineering--High-level Petri nets--Part1: Concepts, definitions and graphical notation

Keller G, Nüttgens M, Scheer A-W (1992) Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)." Universität des Saarlandes

Knuppertz T, Schnägelberger S (2008) Status Quo Prozessmanagement 2007/2008 – Ergebniszusammenfassung. Komptenzzentrum für Prozessmanagement

Ko RKL, Lee SSG, Lee EW (2009) Business process management (BPM) standards: a survey. Business Process Management Journal 15:744–781

La Rosa M, Dumas M, ter Hofstede AHM, Mendling J (2011) Configurable multi-perspective business process models. Information Systems Special Issue: Semantic Integration of Data, Multimedia, and Services 36:313–340

Melcher J (2014) Process Measurement in Business Process Management : Theoretical Framework and Analysis of Several Aspects. KIT Scientific Publishing

Mendling J (2007) Detection and prediction of errors in EPC business process models. PhD Thesis, Institute of Information Systems and New Media, University of Vienna

Mendling J (2008) Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness

Mendling J, Neumann G, Nüttgens M (2005a) Yet another event-driven process chain. In: Lecture Notes in Computer Science, pp 428–436

Mendling J, Nüttgens M (2006) EPC markup language (EPML): An XML-based interchange format for event-driven process chains (EPC). Information Systems and e-Business Management 4:245–263.

Mendling J, Nüttgens M (2002) Event-Driven-Process-Chain-Markup-Language (EPML): Anforderungen zur Definition eines XML-Schemas für Ereignisgesteuerte Prozessketten. In: Nüttgens M, Rump FJ (eds) Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)", pp 87–94

Mendling J, Nüttgens M (2003) Konzeption eines XML-basierten Austauschformates für Ereignisgesteuerte Prozessketten (EPK) XML-basierter Modelldatenaustausch. Wirtschaftsinformatik 10:89–103.

Mendling J, Nüttgens M (2004a) Exchanging EPC Business Process Models with EPML. In: Nüttgens M, Mendling J (eds) XML Interchange Formats for Business Process Management (XML4BPM 2004). Marburg, Germany, pp 61–79

Mendling J, Nüttgens M (2004b) Transformation of ARIS Markup Language to EPML. In: Nüttgens M, Rump FJ (eds) Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)." Luxemburg, pp 27–38

Mendling J, Recker J, Rosemann M, van der Aalst WMP (2005b) Towards the Interchange of Configurable EPCs: An XML-based Approach for Reference Model Configuration. In: Proceedings Workshop Enterprise Modelling and Information Systems Architectures, pp 8–21

Nüttgens M, Rump FJ (2002) Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen P-21:64–77.

Nüttgens M, Zimmermann V (1998) Geschäftsprozeßmodellierung mit der objektorientierten Ereignisgesteuerten Prozeßkette (oEPK). In: Maicher M, Scheruhn H-J (eds) Informationsmodellierung - Referenzmodelle und Werkzeuge. Gabler, Wiesbaden, Germany, pp 23–35

OMG (2011) Business Process Model and Notation (BPMN) Version 2.0. http://www.omg.org/spec/BPMN/2.0/. Accessed 27 Sep 2015

Recker J, Indulska M, Rosemann M, Green P (2006) How Good is BPMN Really? Insights from Theory and Practice. In: Proceedings of the 14th European Conference on Information Systems. Goeteborg, Sweden

Recker JC, Rosemann M, van der Aalst WMP, Mendling J (2005) On the Syntax of Reference Model Configuration – Transforming the C-EPC into Lawful EPC Models. In: Business Process Management Workshops: BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM, WSCOBPM, BPS. Nancy, France, pp 60–75

Rittgen P (2000) Quo vadis EPK in ARIS? Ansätze zu syntaktischen Erweiterungen und einer formalen Semantik. Wirtschaftsinformatik 42:27–35

Sarshar K, Dominitzki P, Loos P (2005) Einsatz von Ereignisgesteuerten Prozessketten zur Modellierung von Prozessen in der Krankenhausdomäne – Eine empirische Methodenevaluation. EPK 2005 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, pp 97–116

Thomas O (2009) Fuzzy Process Engineering. Gabler Edition Wissenschaft, Wiesbaden, Germany

Thomas O, Dollmann T (2008) Towards the Interchange of Fuzzy-EPCs: An XML-based Approach for Fuzzy Business Process Engineering. In: Multikonferenz Wirtschaftsinformatik. Garching, Germany, pp 1999–2010

Thomas O, Hüsselmann C, Adam O (2002) Fuzzy-Ereignisgesteuerte Prozessketten - Geschäftsprozessmodellierung unter Berücksichtigung unscharfer Daten. EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens, pp 7–16.

Vom Brocke JM, Simons A, Niehaves B, et al (2013) Reconstructing the giant: On the importance of rigour in documenting the literature search process. In: 17th European Conference on Information Systems, pp 1–13

W3C (2008) Extensible Markup Language (XML) 1.0 (Fifth Edition). http://www.w3.org/TR/REC-xml/. Accessed 27 Sep 2015

Webster J, Watson R (2002) Analyzing the past to prepare for the future: writing a literature review. Management Information Systems Quarterly 26:8–13.

Winter A, Simon C (2006) Using GXL for exchanging business process models. Information Systems and e-Business Management 4:285–307.