On the de-facto Standard of Event-driven Process Chains: How EPC is defined in Literature

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

Abstract: The Business Process Modelling Notation (BPMN) and the Event-driven Process Chain (EPC) are both frequently used modelling languages to create business process models. While there is a well-defined standard for BPMN, such a standard is missing for EPC. As a standard would be beneficial to improve interoperability among different vendors, this paper aims at providing the means for future EPC standardization. Therefore, we have conducted a structured literature review of the most common EPC variants in IS research. We provide a structured overview of the evolution of different EPC variants, describe means and capabilities and elaborate different criteria for decision-making in regard to including EPC variants in a standardization process.

Keywords: event-driven process chain, EPC, process modelling, literature review, EPC variants, EPC dialects, exchange formats, EPC evolution, standardisation

1 Process Modelling with Event-driven Process Chains

To support the management of business processes, a multitude of business process modelling languages (BPML) has emerged over time, for example BPMN or the Unified Modelling Language (UML) [Aa13]. As a result of the growing interest of researchers and practitioners, many BPML have been standardized by respective standard development organizations (SDO) [KLL09]. One of the most dominant languages for business process modelling is the EPC developed in 1992. Consequently, the EPC has been extensively researched and is still ongoing subject of discussion in the business process management (BPM) domain [Fe09, Fe13, HFL09, Ri00]. However, despite its maturity, no attempts for a successful EPC standard-making have been undertaken to this day. As a consequence, EPC loses ground compared to other languages regarding diffusion, usage and acceptance [DKK14, Fe13, KLL09].

A BPML standard typically contains constructs such as modelling elements, a (formal) syntax, a meta-model or a model exchange format [e.g. Om11]. Ultimately, a standard ensures international adherence to those constructs and not only serves as an agreed-upon

[1] University of Koblenz-Landau, Institute for Information Systems Research, Universitätsstraße 1, 56016, Koblenz, riehle@uni-koblenz.de, delfmann@uni-koblenz.de
[2] University of Osnabrück, Institute for Information Management and Information Systems, Katharinenstraße 3, 49069, Osnabrück, arne.karhof@uni-osnabrueck.de, sven.jannaber@uni-osnabrueck.de, oliver.thomas@uni-osnabrueck.de
[3] University of Münster, European Research Center for Information Systems, Leonardo Campus 3, 48149, Münster, becker@ercis.uni-muenster.de

basis for further language refinement, but also facilitates applicability in practice, e.g. by increasing the interoperability of process models [Fo03, MN06]. Although there exist numerous publications that propose specifications for constructs of the EPC language [e.g. MA07, NR02, Ro08], a unified approach towards EPC standardization has not been initiated to date [KLL09]. The challenges of systematic standardization primarily originate in the large amount of scientific contributions that have been published, which have significantly increased the variety of EPC-related propositions. Hence, identifying and determining what language constructs and extensions to include in an EPC standard becomes a difficult task.

The paper at hand aims at providing the groundwork for a successful EPC standardization. The characterized obstacles are tackled by reviewing, ultimately synthesizing and evaluating relevant scientific work in the field of EPCs. Since standard-making heavily depends on agreement and consensus of a domain community [DG90, FKL03], an overview over state-of-the-art in EPC research is necessary to create a common understanding of the EPC language. The main focus of this paper is hereby put on the evolution of the EPC language, more specifically on the various language variants that have been developed over time, which extend the basic EPC language with additional concepts and constructs. We believe that providing transparency over a language's natural evolution is needed for the establishment of a common ground that supports a subsequent successful standard-making procedure.

The conducted literature review has been able to examine the evolution of the EPC in terms of variants and included concepts that have emerged over the last years. In doing so, the paper extends the body of knowledge by providing an overview of the EPC language progression over time. Furthermore, relevant language extensions are introduced and evaluated according to predefined criteria, ultimately resulting in a consolidation of previous work that is able to serve as a basis for an EPC standard. Finally, the paper proposes a suggestion of constructs and extensions to consider for an EPC standard.

The paper is structured as follows. Section 2 introduces basic concepts of the EPC language. Additionally, the evolution of related business process management languages is highlighted briefly. In Section 3, the applied research methodology is characterized. The findings of the literature review are presented in Section 4 and discussed in Section 5. The paper concludes with a summary of the results with respect to the research question and an outlook indicating further work.

2 Theoretical Background and Related Work

At the time when EPCs emerged in the 1990s from a joint work of the Institute for Information Systems in Saarbrücken and SAP [KNS92], the effort in BPM standardization has been a negligible factor. Actually, the first standard published in this particular field has been the Workflow Reference Model by the Workflow Management Coalition and was first released three years after the EPC, in 1995 [Ho95]. If we instance the year 2004, in which the BPMN has been publically released, there have been more than ten organizations, who developed standards in the field of BPM [NM06]. Furthermore, not only the quantity of standardization endeavours has been smaller, but also the extent significantly differed from nowadays standards. While the initial standard of the Workflow Reference Model has been described on approximately 50 pages, the BPMN 2.0 standardization paper goes beyond the constraint of 500 pages [Om11]. Despite this heavy increase of extent, the generally recognized need for standardization in 2004 led to the development of a BPMN standard in only two years, while after more than one decade there is still no received standard for EPC models.

Besides the lack of a de-jure standard for EPC models, there is another ancillary effect ensuing from the historical background – as there never has been a coalition, a committee or any other form of society establishing a standardization process and additionally discussing possible extensions of the EPC, an equivalent amount of proposals to alter the EPC have been developed and published. For partial overviews of these extensions, the reader can refer to [Me08] and [SDL05]. Unfortunately, these papers do not provide insights to the evolution of the EPC-extensions, their dependencies among themselves, or detailed descriptions. Also, the papers' main issues do not concern EPC extensions themselves. Therefore, they are just mentioning or briefly describing them. In consequence of this heterogenous area of EPC interpretations, it is necessary to gain a structured overview of these versions and to cover the mentioned aspects above. Only then it is possible to completely capture all relevant aspects for an EPC standard. There are some publications, which already address partial areas of EPC standardization, either by proposing formal notations of the EPC [Aa99, Me08] or by assembling informal rules for EPC modelling [e.g. Fe13]. Other publications have analysed file-based exchange formats for EPC models [Ri16] and the implementation of EPC in different modelling tools [Ka16]. Unfortunately, such papers neglect the occurrence of EPC-variations and do not discuss their relevance.

In favor of a better understanding of the proposed alterations which are given in section 4, we briefly introduce the initial EPC elements as introduced by [KNS92]. This first version of the EPC only consisted of events, functions and operators. Thereby, the event is defined as an "occurrence of a defined condition", while the function is a "process that converts an input state to an output state". Beginning with events, these entities always have to alternate. Furthermore, the initial EPC provided operators for splitting and joining the control flow of a process model. These operators were distinguished as conjunctive, disjunctive and adjunctive and corresponded to AND, XOR and OR operators respectively.

3 Methodology

We have conducted a structured literature review in the discipline of information systems, as suggested by [WW02] and [Br13]. Since the first publication regarding the EPC was published as a working paper in the working paper series of the institute for information systems at the University of Saarbrücken, we included their working series with a total of 198 papers as a data source in our review process. Additionally, we considered the EPC workshop from 2002 to 2009, whose proceedings with a total of 57 papers were included as our second data source. Lastly, we queried two scientific search engines with the terms "Ereignisgesteuerte Prozesskette" and "event-driven process chain" (in several spellings) to find further publications. We are aware that these generic search queries deliver a huge result set, however we think that there is no simple and more specific query that still includes all publications of different EPC adoptions in different application areas. Altogether, our queries returned 1.806 publications at SpringerLink[4] and 198 publications at ScienceDirect[5]. The large difference is due to SpringerLink including more German publications than ScienceDirect.

Due to the large number of 2.259 publications, we followed [WW02] in considering only titles first and discarded papers, whose titles suggested an application of EPC without contributing to the EPC modelling language itself. Further, for technical reasons, we excluded papers of which we could neither find a full text online nor in the libraries of three different universities. This left us with 316 papers. After removing duplicates, reading the abstracts and where necessary the contents as well, we came to a set of 79 publications in our first iteration.

Next, we conducted a forward and backward search to find additional literature that might not have been found by our search engines. While the backward search was done manually, we used Google Scholar[6] for the forward search. In this iteration we also revised some of our decisions on discarding papers in the previous iteration, e.g. in case a referenced paragraph generated more insight to the papers' content than solely its title. After reading the abstracts of the papers delivered by forward and backward search, and after another removal of duplicates, we found 35 new publications. Therefore, we came to a final set of 114 papers.

4 Different EPC Variants from Literature

More than 20 years ago, the idea of modelling business processes with alternating events and functions was published by [KNS92]. Not surprisingly, event-driven process chains have heavily developed over the years, leading to similar, yet different understandings of EPC in literature. Many authors have suggested extending the EPC in one or the other way and several formalizations of the EPC modelling language have been provided.

Within our set of 144 publications regarding EPCs, we found 14 different variations of the EPC, which we will call EPC dialects. For all these 14 different dialects, the authors described in one or more papers how their variation of the EPC is defined and in which

[4] http://link.springer.com/
[5] http://sciencedirect.com/
[6] http://scholar.google.com/

terms they modified the base EPC by [KNS92]. An overview of these 14 EPC dialects is provided in Tab. 1. Additionally, we identified 8 different exchange formats for EPC models (counting different versions of the same format separately, due to incompatibilities), which are capable of storing EPC models of one or more of the 14 different EPC dialects.

Name	References	Short Abstract
EPC	[KNS92]	The prototype of all event-driven process chains, consisting only of functions, events and connectors
Extended EPC (**eEPC**)	[HKS93] [GS94] [KT97]	An enriched EPC, including organizational units, information objects, IT systems and process refinements
Real-Time EPC (**rEPC**)	[HWS93]	Includes a state machine, which is modelled in parallel to the EPC, events and functions of the EPC are linked to conditions and actions of the state machine
EPC*	[ZR96]	Information objects can be connected to model data flow and conditions can be added to the control flow, e.g. to start events
Object-oriented EPC (**oEPC**)	[SNZ97] [NZ98]	Uses an object-oriented process definition, where activities are carried out on business objects, therefore such objects are modelled instead of traditional functions
Risk EPC	[BO02]	Adds a risk element to the EPC which can be linked with function to visualize potential business risks
Fuzzy EPC	[THA02] [TD06] [Th09]	A new fuzzy connector allows to model fuzzy decision-making in business processes, using variables and decision rules related to the fuzzy connector
Yet another EPC (**yEPC**)	[MNN05]	Extends the EPC to support state-based workflow patterns, multi-instantiation and cancellation
Risk EPC extended	[RM05] [RW08]	Adds different risk events which are triggered when an exception occurs, further risk-management functions can be used to handle the exception
Configurable EPC (**C-EPC**)	[RA07] [Re05]	Functions, events and connectors are configurable, several EPCs can be generated from a single EPC reference model
Semantic EPC	[TF06] [FKS09]	EPC elements are linked to ontologies, to improve semantics and prevent ambiguity
Nautilus EPC (**N-EPC**)	[KUL06]	Allows modelling of several events between two functions, trivial events can make decisions and can hence be followed by an XOR or OR connector
Service EPC	[HW07]	Traditional functions are replaced by services, which can be synchronous or asynchronous, introduce timeouts and message events
Configurable integrated EPC (**C-iEPC**)	[Ro08]	Extends the configuration introduced in the C-EPC to information objects, IT systems and organizational objects, allowing an even greater range of configuration

Tab. 1: Overview of EPC dialects most prominent in literature

While the original EPC only considered the process flow by using events and functions, this basic EPC model was soon extended. A very early extension was the real-time extension to EPC, which included a state machine similar to Petri nets. Events and function of the EPC are linked to conditions and actions of the state machine, enabling a state-based execution of rEPC models [HWS93].

Another early extension of the EPC is referred to as extended EPC (eEPC) in literature. Extended EPCs include organizational units, which are connected to functions to model process responsibilities, information objects, which represent abstract data (documents) and can be used or generated by functions, IT systems, which can be used by functions, and process refinements, which aggregate a set of activities, i.e. a subordinate EPC to a single element [GS94, HKS93, KT97, Ro96]. The eEPC was widely accepted and all further dialects are based heron.

The EPC* extension is provided by [ZR96], who tried to overcome the shorting of traditional EPCs only representing the control flow in a process. Therefore, EPC* introduced relations between information objects to represent the data flow within a single process. Furthermore, conditions can be added to control flows to enable decision-making, and organizational units can be connected with start events to model organizational units that are allowed to initiate the process.

If processes are not understood as a set of activities (function-oriented process definition), but rather as operations on a business object (object-oriented process definition), an object-oriented EPC (oEPC) can be used to model business processes [NZ98, SNZ97]. The oEPC replaces functions which object classes, which includes methods for operating on the business object. Still, organizational units and information objects can be attached to objects.

An idea of explicitly modelling business risks in EPC models was first suggested by [BO02], who add a risk element, which when connected with a function indicates that this function can cause a potential problem. We refer to this dialect as the “Risk EPC”. The idea of modelling risks has been further adapted by [RM05] and [RW08], who propose to separate between different kinds of risks. [RW08] suggest three different risk elements, which can be triggered by functions when an exception occurs. These risk events can be followed by different kinds of risk handling functions. This dialect, which we refer to as “Risk EPC extended”, allows not only to model different kinds of risks, but also to model how unexpected situations are handled in the process.

With the Fuzzy EPC, one is capable of modelling fuzzy decision making based on textual variables and decision rules, for which [THA02] introduce the fuzzy connector. Further publications describe attributes and details of Fuzzy EPCs in more detail [TD06, Th09]. Similarly, the Yet another EPC (yEPC) introduces a new connector as well, namely the empty connector. The yEPC further adds process parameters for multi-instantiation and a cancellation area for cancellation support [MNN05], to make the EPC capable of executing common workflow patterns described in [Aa03].

The suggestion of using configurable process models as a basis for reference modelling

[Aa06] lead to the development of a configurable EPC (C-EPC). The C-EPC includes configurable functions, events and connectors which make the control flow of the EPC configurable [RA07, Re05]. With this technique, several different concrete EPC models can be generated from a single C-EPC, depending on the configuration. This approach has been further extended by [Ro08], who specified configuration for organizational units, information objects and IT systems. This so-called configurable integrated EPC (C-iEPC) enables extremely configurable scenarios and therefore a high reusability of EPC models.

A simplified dialect called nautilus EPC (N-EPC) has been proposed by [KUL06]. Contrasting to other EPC dialects, trivial events are able to make a decision. Therefore, trivial events can be followed by an XOR or an OR operator. Technically, this dialect omits the function succeeding trivial events, as this function is also regarded to be trivial.

Another fairly recent publication adopts the idea of IT services and regards business processes as services, which are consumed during the process execution. Consequently, [HW07] add functions, which refer to service calls that can either be synchronous or asynchronous. We refer to this dialect as “service EPC”. Since the introduction of asynchronous functions requires some kind of message processing when an asynchronously executed function finishes, the service EPC also introduced events to be triggered upon message receipt. This allows a service EPC to start the execution of several asynchronous services first, and then to wait until the service execution finishes.

Besides these 14 different EPC dialects described in literature, we also found different exchange formats. Exchange formats provide a computer-readable data storage for EPC models, which helps to store, transfer and reuse EPC models in different environments. The specification of an exchange format provides an implicit definition of an EPC dialect, by the EPC elements which have been implemented in the concrete exchange format. For all 8 different exchange formats we have identified in literature, we have analysed with which of the 14 different EPC dialects they are compatible with. The evolution of EPC dialects and exchange formats over time is shown in Fig. 1.

A first XML notation for EPC models (XML EPC) has been proposed by [GK02], who provide an XML schema to describe an eEPC, including functions, events, connectors, information objects, organizational units and process refinements. At the same time, a similar yet incompatible XML schema has been described by [MN02], who introduce the EPC Markup Language (EPML) as a general purpose format for exchanging EPC models [MN04].

[WS06] use a different approach by adopting the Graph Exchange Language (GXL) for use with EPC models. GXL is capable of storing any conceptual models as graphs, using nodes to reflect conceptual elements and edges to reflect relations. Though [WS06] do not provide an example for all elements of the eEPC, their exchange format is capable of representing all eEPC elements, as new node types can easily be defined using a custom identifier.

For semantic EPCs, [FKS09] have proposed an ontology based exchange format. By transforming elements of the semantic EPC to ontology concepts, they embed the EPC model within an ontology. We refer to this approach as “XML EPC Ontology”. Similar to the GXL approach, this leads to storage formats where EPC elements are only meaningful by their description, which usually is a string-based identifier. However, for automated processing it is helpful, if EPC elements are defined within the specification on an exchange format.

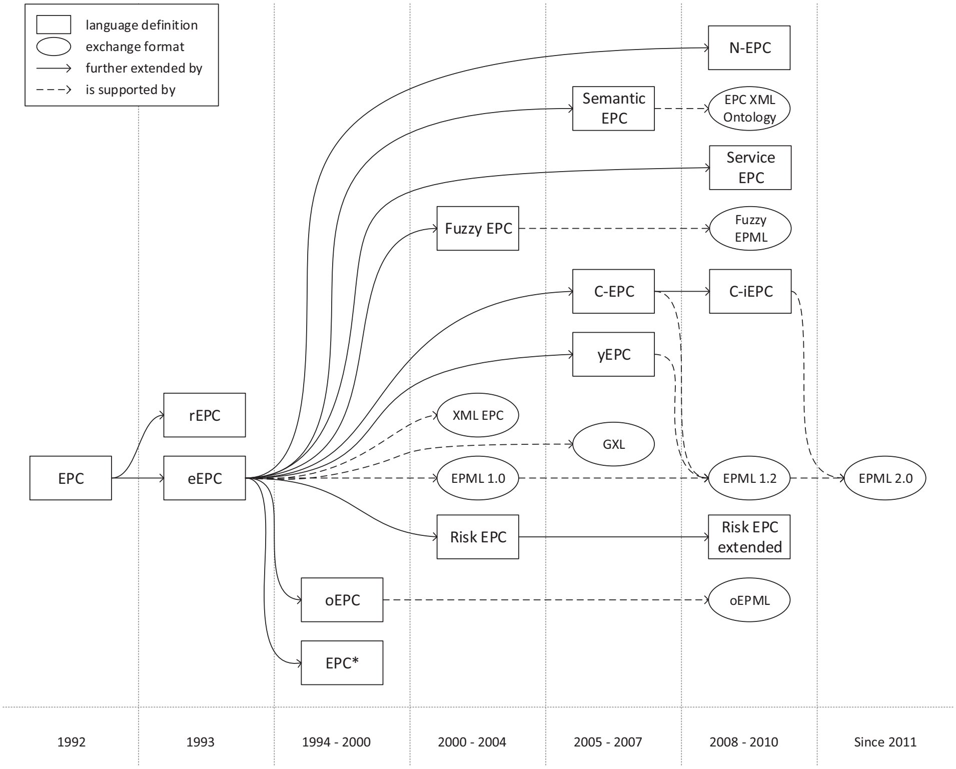

Fig. 1: Evolution of different EPC dialects and exchange formats over time

Besides these four different and independently developed formats, the EPML approach by [MN02] has been adapted several times. A modification called oEPML has been developed by [Ho09], who extends the XML schema for objects of the oEPC dialect. Similarly, [TD08] suggested a modification named Fuzzy EPML, which adds a fuzzy connector to the EPML schema and therefore is capable of representing Fuzzy EPC models. However, Fuzzy EPML and oEPML have not been integrated yet, making these two modifications of EPML incompatible to each other. As a consequence, there is no exchange format that can be used for eEPC, Fuzzy EPC and oEPC models altogether.

After a modification for EPML to support C-EPC models was suggested by [Me05], an updated schema for EPML has been released as EPML 1.2 [Me09], which supports yEPC and C-EPC model. Additionally, with EPML 2.0 [Me11], support for C-iEPC models was added, which makes EPML 2.0 capable of storing eEPC, yEPC, C-EPC and C-iEPC models. From the number of supported EPC dialects, EPML 2.0 supports the largest variety of EPC dialects.

Summing up the results of our structured literature review, we found 14 different EPC dialects and 8 different exchange formats. Most EPC dialects developed in the last 20 years are based on the eEPC dialect, therefore these dialects share a common understanding of not only events, functions and connectors, but also of information objects, IT systems and organizational units.

There are some extensions, which we have not considered as EPC dialects, for example the SEQ connector [Pr95] or the ET and OR1 connector [Ro96]. These connectors focus on providing a simplification for modelling and can be added to any EPC dialect. Additionally, the SEQ, ET and OR1 connectors can be converted into structures consisting of eEPC elements only, i.e. functions, events and XOR, OR and AND connectors [Ru99, p.62 ff.]. Furthermore, we have neither considered modified EPCs (modEPC) nor agent-oriented EPCs (xEPC), as these EPC dialects were only briefly discussed at one single conference and in working papers of the years 1999 and 2000. Unfortunately, we were not able to obtain a full-text as described in the methodology section before.

5 Towards an EPC Standard

The diversity of EPC dialects and exchange formats calls for a standardization procedure to develop a specification for an integrated EPC language and exchange format. As mentioned in our introduction, a standardized modelling language can be more easily implemented by tool vendors and hence can spread faster in the BPM community.

It is worth mentioning that an EPC standard would not only be beneficial for companies that start modelling business processes with event-driven process chains, but also for companies that already have modelled EPCs in the past and still store these models as legacy models. With an EPC standard, such legacy models could be reused in any modelling environment supporting the prospective EPC standard. Possibly some legacy models have to be revised, but it is safe to assume that basic EPC construct will find their way into the standard. Moreover, procedures for migrating one modelling language to another, e.g. EPC to BPMN as suggested by [DT09], could be implemented independently of a modelling tool.

While the benefits of an EPC standard are numerous and rather obvious, there are several open questions yet to be answered. Most importantly, it has to be decided which EPC dialects should be part of an EPC standard. There are several criteria that one might consider. On the one hand, one could evaluate the dominance of an EPC dialect in the literature, e.g. for the rEPC there is – to the best of our knowledge – only one publication [HWS93], which might be an indicator for the rEPC being less relevant to researchers. On the other hand, one might consider the compatibility of different EPC dialects. For example, the N-EPC breaks with the formal semantics of traditional EPCs, as it allows trivial events to make decisions, which is not allowed in any other EPC dialect presented in this paper. Therefore, one might exclude the N-EPC for compatibility reasons.

Besides regarding EPC dialects in the literature, one might also consider the spread of EPC dialects in practice. Since an EPC standard should be implemented and used by practitioners, such a standard should include all artefacts which are needed in practice. Therefore, one should perform a market analysis of BPM modelling tools to get an understanding of EPC dialects which are commonly used in practice. The results of such a market analysis should be included in the process of deciding which EPC dialects are becoming parts of an EPC standard.

Since practitioners can only use the EPC dialects which have been implemented in EPC modelling tools, there might be a discrepancy between what practitioners would like to see in an EPC model and what they can actually model with their modelling tool. Therefore, a survey among practitioners might deliver further insights, especially in terms of useful EPC dialects which have not made it into modelling tools yet.

To further circumstantiate the integration or exclusion of certain EPC dialects and artefacts in an EPC standard, one could interview EPC experts. Interviews with individuals might deliver more detailed results than a survey and provide better arguments for decision-making. Together with a group of experts, a final decision on what to be included in an EPC standard should be made. This should be done once modelling tools have been analysed and practitioners have been surveyed.

For a first step towards standardization, we have analysed the type of EPC extension, the type of specification of that extension, the impact in research and the degree of generalization for each 14 EPC dialects. An overview is provided in Tab. 2. To estimate the impact in research, we have used Google Scholar to conduct a forward search for each paper and to count the number of citations each paper has received (column “Cites” in Tab. 2). For EPC dialects that consist of multiple papers, we have conducted a forward search for each paper, and have built an outer set, i.e. eliminated all duplicates in the citations. That number is listed in brackets in the column “Impact”. We have classified the impact as “high” if there were more than 100 citations, “noticeable” if there were more than 50 citations, “medium” if there were more than 25 citations and “low” otherwise.

In regard to the way an EPC dialect was specified, we distinguished between an enumerative specification, i.e. the authors provided a textual or visual list of all elements or extensions they provided, a formal semantic specification, i.e. a specification in terms of mathematical equations based on set theory and a meta-model based specification, where the authors provided a meta-model for their EPC variant.

The column “Type of Extension” in Tab. 2 lists the means of each EPC variant, for example if the authors provide additional elements to enrich the information of an EPC model or if the authors aim at a different goal like achieving execution of workflows.

Lastly, the column “Generalizable” describes whether the variant is applicable to business process models modelled in EPC in general or if the dialect aims at a certain application domain. As the N-EPC breaks with formal semantics of other EPC dialects, it is not generally applicable to other EPC models.

Dialect	Reference	Cites	Impact	Specification	Type of Extension	Generalizable
EPC	[KNS92]	993	high	enumerative	-	generally applicable
eEPC	[HKS93]	67	high (163)	enumerative	additional elements	generally applicable
	[GS94]	19		enumerative		
	[KT97]	84		enumerative		
rEPC	[HWS93]	10	low	formal semantics	execution of processes	generally applicable
EPC*	[ZR96]	35	medium		execution of workflows	generally applicable
oEPC	[SNZ97]	64	noticeable (74)	enumerative	additional elements	for modelling business objects
	[NZ98]	18		enumerative		
Risk EPC	[BO02]	25	low	enumerative	additional element	generally applicable
Risk EPC extended	[RM05]	150	high (166)	enumerative	additional elements	generally applicable
	[RW08]	17		meta-model		
Fuzzy EPC	[THA02]	24	medium (34)	meta-model	additional elements and decision tables	generally applicable
	[TD06]	10		formal semantics		
	[Th09]	9		meta&formal		
yEPC	[MNN05]	41	medium	enumerative	execution of workflows, additional elements	generally applicable
C-EPC	[RA07]	505	high (526)	formal semantics	reference modelling	generally applicable
	[Re05]	42		enumerative		
Semantic EPC	[TF06]	5	low (22)	enumerative	annotation w/ontologies	generally applicable
	[FKS09]	17		enumerative		
N-EPC	[KUL06]	22	low	formal semantics	simplification	conflicts with formalization of common EPC dialects
Service EPC	[HW07]	8	low	enumerative	additional elements	for modelling business services
C-iEPC	[Ro08]	67	noticeable	formal semantics	extended configuration	generally applicable

Tab. 2: Attributes of different EPC dialects for decision-making in a standardization process

Besides fostering a common understanding of EPC elements, an EPC standard should also provide an exchange format for EPC models to improve interoperability among different modelling tools. We think that such an exchange format should be focused on EPC models only and not on process models in general. Approaches like GXL, which can be applied to different process modelling languages as EPC or BPMN, are more complex, thus harder to implement for tool developers and, in most cases, do only support portions of modelling languages, namely those concepts which are supported by all considered languages. Contrastingly, an EPC-specific exchange format like EPML has a well-defined set of element types it can represent, namely those of the underlying EPC dialect. This makes adoption of such an exchange format in modelling tools much easier.

An exchange format for EPC models should be designed carefully. If an EPC standard is extensible, as discussed above, an exchange format needs to be extensible as well. Therefore, an exchange format should be designed in a way that additional information can be added without breaking backward compatibility, so that a tool, which can import a basic EPC model, can also import an EPC model with extensions it is not aware of. Of course, such extensions the modelling tool is not aware of will not be displayed or in any way handled by the modelling tool. The important point is that a modelling tool needs to recognise information in the exchange format it does not know as an unknown EPC extension that has not been implemented in the tool. Then, the modelling tool can simply ignore the additional information. Obviously, in this scenario, an EPC model might become incomplete if imported in a modelling tool which does not implement all EPC extension used within the EPC model. However, this way, a modelling tool could still import as much as possible from a model, providing the greatest possible benefits to users in terms of reusing models.

6 Conclusion

In this paper, we have performed a structured literature review on business process modelling with event-driven process chains and have identified 14 different EPC dialects in the literature. While an overview of EPC dialects has already been provided in [Me08, p.28 ff.] and [SDL05], we have found more EPC dialects and we have described all EPC dialects we found in more detail. Additionally, we have described the evolution of different EPC dialects over time, which – to the best of our knowledge – has not been done to this day. Therefore, we provide a more recent and greater overview of EPC variants.

Furthermore, we have considered exchange formats for EPC models, of which we have found 8 different exchange formats in literature. We have related these 8 different exchange formats to the 14 different EPC dialects we found and have analysed, which exchange format can be used for storing EPC models of which dialect. With this analysis, we have shown that for some EPC dialects there are no well-known exchange formats and that there are several incompatible exchange formats. Only a few exchange formats support more than one EPC dialect.

Lastly, we have analysed the 14 different EPC dialects in four different criteria, their impact in research, the type of specification, the type and mean of the variant and the generalizability. By this, we have shown that some EPC variants can be considered being more prominent than others and therefore being more adequate for inclusion in a future EPC standard.

With our research, we have contributed towards the development of a standardized EPC modelling language and a standardized EPC exchange format, shortly referred to as an EPC standard. While a standard is beneficial for research and practice under several aspects, such a standard has not been developed yet. With our research, we have provided parts of a research agenda towards a successful EPC standardization.

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.

References

[Aa03] Aalst, W.M.P. van der, Hofstede, A.H.M. ter, Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14, pp. 5–51. 2003.

[Aa06] Aalst, W.M.P. van der, Dreiling, A., Gottschalk, F., Rosemann, M., Jansen-Vullers, M.H.: Configurable Process Models as a Basis for Reference Modeling. In: C.J. Bussler, A. Haller (eds.) Business Process Management Workshops - BPM 2005 International Workshops. pp. 512–518. Springer, Berlin 2006.

[Aa13] Aalst, W.M.P. van der.: Business Process Management: A Comprehensive Survey. ISRN Software Engineering 2013, pp. 1–37. 2013.

[Aa99] Aalst, W.M.P. van der.: Formalization and verification of event-driven process chains. Information and Software Technology 41, pp. 639–650. 1999.

[BO02] Brabänder, E., Ochs, H.: Analyse und Gestaltung prozessorientierter Risikomanagementsysteme mit Ereignisgesteuerten Prozessketten. In: EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens, pp. 1–5. 2002.

[Br13] Brocke, J.M. vom, Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A.: Reconstructing the giant: On the importance of rigour in documenting the literature search process. In: 17th European Conference on Information Systems, pp. 1–13. Verona, Italy 2013.

[DG90] David, P.A., Greenstein, S.: The Economics of Compatibility of Standards: An Introduction to Recent Research. Economics of Innovation and New Technology 1, pp. 3–41. 1990.

[DKK14] Drawehn, J., Kochanowski, M., Kötter, F.: Business Process Management Tools 2014. Stuttgart, Germany 2014.

[DT09] Decker, G., Tscheschner, W.: Migration von EPK zu BPMN. In: EPK 2009. 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitkreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK). Gesellschaft für Informatik, pp. 91–109. 2009.

[Fe09] Fettke, P.: Ansätze der Informationsmodellierung und ihre betriebswirtschaftliche

Bedeutung: Eine Untersuchung der Modellierungspraxis in Deutschland. Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung (zfbf) 61, pp. 550–580. 2009.

[Fe13] Fellmann, M., Bittmann, S., Karhof, A., Stolze, C., Thomas, O.: 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), pp. 103–116. St. Gallen, Switzerland 2013.

[FKL03] Fomin, V., Keil, T., Lyytinen, K.: Theorizing about standardization: integrating fragments of process theory in light of telecommunication standardization wars. Sprouts: Working Papers on Information Environments, Systems and Organizations 3, pp. 29–60. 2003.

[FKS09] Filipowska, A., Kaczmarek, M., Stein, S.: Semantically annotated EPC within semantic business process management. In: Lecture Notes in Business Information Processing, pp. 486–497. Milano, Italy 2009.

[Fo03] Fomin, V.: The Role of Standards in the Information Infrastructure Development. MISQ Special Issue Workshop 1993, pp. 302–313. 2003.

[GK02] Geissler, M., Krüger, A.: Eine XML-Notation für Ereignisgesteuerte Prozessketten (EPK). In: Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises “Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK),” pp. 81–86. Trier, Germany 2002.

[GS94] Galler, J., Scheer, A.-W.: Workflow-Management - Die ARIS-Architektur als Basis eines multimedialen Workflow-Systems. In: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), No. 108, Universität des Saarlandes 1994.

[HFL09] Houy, C., Fettke, P., Loos, P.: Stilisierte Fakten der Ereignisgesteuerten Prozesskette – Anwendung einer Methode zur Theoriebildung in der Wirtschaftsinformatik. In: EPK 2009. 8. Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitkreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK). Gesellschaft für Informatik, pp. 22–41. Bonn, Germany 2009.

[HKS93] Hoffmann, W., Kirsch, J., Scheer, A.-W.: Modellierung mit Ereignisgesteuerten Prozeßketten. In: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), No. 101, Universität des Saarlandes 1993.

[Ho09] Hogrebe, F., Nüttgens, M., Kern, H., Kühne, S.: 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), 2009.

[Ho95] Hollingsworth, D.: The Workflow Reference Model. Management am, pp. 1–55. 1995.

[HW07] Huth, S., Wieland, T.: Geschäftsprozessmodellierung mittels Software-Services auf Basis der EPK. In: V. Nissen, M. Petsch, H. Schorcht (eds.) Service-orientierte Architekturen. pp. 61–76. Deutscher Universitäts-Verlag, Wiesbaden 2007.

[HWS93] Hoffmann, W., Wein, R., Scheer, A.-W.: Konzeption eines Steuerungsmodells für Informationssysteme–Basis für die Real-Time-Erweiterung der EPK (rEPK). In: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), No. 106, Universität des Saarlandes 1993.

[Ka16] Karhof, A., Jannaber, S., Riehle, D.M., Thomas, O., Delfmann, P., Becker, J.: On the de-facto Standard of Event-driven Process Chains: Reviewing EPC Implementations in Process Modelling Tools. In: Proceedings of the Modellierung 2016, Karlsruhe, Germany 2016.

[KLL09] Ko, R.K.L., Lee, S.S.G., Lee, E.W.: Business process management (BPM) standards: a survey. Business Process Management Journal 15, pp. 744–781. 2009.

[KNS92] Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage “Ereignisgesteuerter Prozeßketten (EPK).” In: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), No. 89, Universität des Saarlandes 1992.

[KT97] Keller, G., Teufel, T.: SAP R/3 prozeßorientiert anwenden. Addison-Wesley, Bonn 1997.

[KUL06] Kopp, O., Unger, T., Leymann, F.: Nautilus Event-driven Process Chains: Syntax, Semantics, and their mapping to BPEL. In: Proceedings of the 5th GI Workshop on Event-Driven Process Chains (EPK 2006), pp. 85–104. 2006.

[MA07] Mendling, J., Aalst, W.M.P. van der.: Formalization and Verification of EPCs with OR-Joins Based on State and Context. Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE’07) 4495, pp. 439–453. 2007.

[Me05] Mendling, J., Recker, J., Rosemann, M., Aalst, W.M.P. van der.: 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. 2005.

[Me08] Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. 2008.

[Me09] Mendling, J.: EPML Schema 1.2. 2009. Available at: http://www.mendling.com/EPML/EPML_12.xsd, accessed 15. Oct 2015.

[Me11] Mendling, J.: EPML Schema 2.0. 2011. Available at: http://www.mendling.com/EPML/EPML_2.0.xsd, accessed 15. Oct 2015.

[MN02] Mendling, J., Nüttgens, M.: Event-Driven-Process-Chain-Markup-Language (EPML): Anforderungen zur Definition eines XML-Schemas für Ereignisgesteuerte Prozessketten. In: Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises “Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)”2, pp. 87–94. 2002.

[MN04] Mendling, J., Nüttgens, M.: XML-based Reference Modelling: Foundations for an EPC Markup Language. In: J. Becker, P. Delfmann (eds.) Referenzmodellierung. pp. 51–72. Physica-Verlag, Heidelberg 2004.

[MN06] Mendling, J., Nüttgens, M.: EPC markup language (EPML): An XML-based interchange format for event-driven process chains (EPC). Information Systems and e-Business Management 4, pp. 245–263. 2006.

[MNN05] Mendling, J., Neumann, G., Nüttgens, M.: Yet another event-driven process chain. In: Lecture Notes in Computer Science, pp. 428. 2005.

[NM06] Nickerson, J. V, Muehlen, M. zur.: The Ecology of Standards Processes: Insights from Internet Standard Making. MIS Quarterly 30, pp. 467–488. 2006.

[NR02] Nüttgens, M., Rump, F.J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen P-21, pp. 64–77. 2002.

[NZ98] Nüttgens, M., Zimmermann, V.: Geschäftsprozeßmodellierung mit der objektorientierten Ereignisgesteuerten Prozeßkette (oEPK). In: M. Maicher, H.-J. Scheruhn (eds.) Informationsmodellierung - Referenzmodelle und Werkzeuge. pp. 23–35. Gabler, Wiesbaden, Germany 1998.

[Om11] OMG.: Business Process Model and Notation (BPMN) Version 2.0. 2011. Available at: http://www.omg.org/spec/BPMN/2.0/, accessed 27. Sep 2015.

[Pr95] Priemer, J.: Entscheidungen über die Einsetzbarkeit von Software anhand formaler Modelle. Pro-Universitate-Verlag, Sinzheim 1995.

[RA07] Rosemann, M., Aalst, W.M.P. van der.: A configurable reference modelling language.

Information Systems 32, pp. 1–23. 2007.

[Re05] Recker, J.C., Rosemann, M., Aalst, W.M.P. van der, Mendling, J.: 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, pp. 60–75. Nancy, France 2005.

[Ri00] Rittgen, P.: Quo vadis EPK in ARIS? Ansätze zu syntaktischen Erweiterungen und einer formalen Semantik. Wirtschaftsinformatik 42, pp. 27–35. 2000.

[Ri16] Riehle, D.M., Jannaber, S., Karhof, A., Delfmann, P., Thomas, O., Becker, J.: Towards an EPC Standardization – A Literature Review on Exchange Formats for EPC Models. In: Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI 2016), Ilmenau, Germany 2016.

[RM05] Rosemann, M., Muehlen, M. zur.: Integrating Risks in Business Process Models. Australasian Conference on Information Systems (ACIS) , pp. 62–72. 2005.

[Ro08] Rosa, M. La, Dumas, M., Hofstede, A.H.M. ter, Mendling, J., Gottschalk, F.: Beyond Control-Flow: Extending Business Process Configuration to Roles and Objects. 5231, pp. 199–215. 2008.

[Ro96] Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen. Gabler, Wiesbaden 1996.

[Ru99] Rump, F.J.: Geschäftsprozeßmanagement auf der Basis ereignisgesteuerter Prozeßketten. B. G. Teubner, Suttgart 1999.

[RW08] Rieke, T., Winkelmann, A.: Modellierung und Management von Risiken. Ein prozessorientierter Risikomanagement-Ansatz zur Identifikation und Behandlung von Risiken in Geschäftsprozessen. Wirtschaftsinformatik 50, pp. 346–356. 2008.

[SDL05] Sarshar, K., Dominitzki, P., Loos, P.: Einsatz von Ereignisgesteuerten Prozessketten zur Modellierung von Prozessen in der Krankenhausdomäne – Eine empirische Methodenevaluation. EPK 2005 -Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten , pp. 97–116. 2005.

[SNZ97] Scheer, A.-W., Nüttgens, M., Zimmermann, V.: Objektorientierte Ergenisgesteuerte Prozeßkette (oEPK) - Methode und Anwendung. In: Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), No. 141, Universität des Saarlandes 1997.

[TD06] Thomas, O., Dollmann, T.: Attributierung und Regelintegration. Proceedings of the 5th GI Workshop on Event-Driven Process Chains (EPK 2006) , pp. 49–68. 2006.

[TD08] Thomas, O., Dollmann, T.: Towards the Interchange of Fuzzy-EPCs: An XML-based Approach for Fuzzy Business Process Engineering. In: Multikonferenz Wirtschaftsinformatik, pp. 1999–2010. Garching, Germany 2008.

[TF06] Thomas, O., Fellmann, M.: Semantische Ereignisgesteuerte Prozessketten. Data Warehousing , pp. 205–224. 2006.

[Th09] Thomas, O.: Fuzzy Process Engineering. Gabler Verlag, Wiesbaden 2009.

[THA02] Thomas, O., Hüsselmann, C., Adam, O.: Fuzzy-Ereignisgesteuerte Prozessketten - Geschäftsprozessmodellierung unter Berücksichtigung unscharfer Daten. EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings des GI-Workshops und Arbeitskreistreffens , pp. 7–16. 2002.

[WS06] Winter, A., Simon, C.: Using GXL for exchanging business process models. Information Systems and e-Business Management 4, pp. 285–307. 2006.

[WW02] Webster, J., Watson, R.: Analyzing the past to prepare for the future: writing a literature review. Management Information Systems Quarterly 26, pp. 8–13. 2002.

[ZR96] Zukunft, O., Rump, F.: From Business Process Modelling to Workflow Management - An Integrated Approach. In: B. Scholz-Reiter, E. Stickel (eds.) Business Process Modelling. pp. 3–22. Springer, Berlin 1996.