

Do We Need a Standard for EPC Modelling?

The State of Syntactic, Semantic and Pragmatic Quality

Michael Fellmann, Sebastian Bittmann, Arne Karhof, Carl Stolze, Oliver Thomas

Information Management and Information Systems,

University Osnabrueck, Germany

{ michael.fellmann | sebastian.bittmann | akarhof | carl.stolze | oliver.thomas }@uni-osnabrueck.de

Abstract: The quality determination of business process models is a complex and demanding task. In literature, a plethora of different quality criteria can be identified and are respectively used by practitioners. The selection of quality criteria depends on the one hand on the respective preferences of the individual modeller. On the other hand, it is prescribed by the modelling language in use, as it has embodied specific criteria e.g. based on its syntax. For the widespread EPC, no comprehensive overview of existing aspects for the evaluation of an EPC model's syntactic, semantic and pragmatic quality exists. With this investigation, we present such an overview and put the identified aspects into a holistic perspective based on the identified and relevant literature.

1 Introduction

Process models are a widespread tool to design, represent, discuss and eventually manage chains of activities in all kinds of businesses – from small companies to global enterprises [HFL11]. Although many methods have been developed to model business processes [BNT10], there is no agreed standard to judge the quality of a process model. Having in mind, there is – besides the variety of methods – no shortage of modelling languages to choose from [TF09], quality criteria either need to be generic, respectively abstract or language-specific. Whilst generic frameworks for process model quality, such as the Generalities of Orderly Modelling [BRS95] or the SIQ-Framework [RMR10], provide general guidelines, they cannot reflect the particularities of specific languages with their individual syntax and semantics. Consequently, the need exists to evaluate particular modelling languages thus going beyond the general guidelines in order to propose implications to the model construction. Thus, in this paper, we focus on the widespread Event-driven Process Chain (EPC) to provide a comprehensive overview of aspects to evaluate the syntactic, semantic and pragmatic quality of models. The EPC is established in academia and practice alike for business process modelling [Fett09; MeRA10], thereby combining practical and theoretical relevance. In addition to this, the EPC language has been investigated for more than 20 years. However, despite its popularity and the large number of modelling tools available, the EPC has never been fully standardised. As no standard for the EPC is available, the quality judgement of the EPC and the resulting EPC models is a demanding task. In this sense, this paper might

contribute to a standard of EPC modelling. Moreover, the language is a candidate to identify important correctness aspects going beyond general frameworks and recommendations. These aspects and the body of research accompanied to them may serve as a reference both for designing new languages and to standardise existing ones.

The remainder of the paper is structured as follows. In Section 2 the literature analysis is described. In Section 3, EPC-specific aspects of syntactic, semantic and pragmatic quality are presented. In Section 4, the results are summarized by providing a comprehensive overview of quality aspects. The paper ends with a conclusion.

2 Preparing and Conducting the Literature Analysis

The scope of our literature analysis is only limited by content. We did not exclude any work based on its age. The content scope was defined as the intersection of the modelling language EPC and evaluation criteria for process model quality. Therefore, only articles mentioning specifically the EPC were considered in the following. We have not limited our search to publications only addressing the basic EPC, instead we also considered publications that focus on quality aspects of eEPC models.

At first, leading and well established databases that are known for publications around the EPC were considered. In applying this approach, we have examined the working paper series of the IWi (Institute for Information Systems at DFKI) – the institute which published the original EPC back in 1992 [KNS92]. The selection of relevant articles from the IWi reports has been made by their titles and abstracts. Initially, all reports dealing with processes were considered (37 papers). From this set, only articles with business process and EPC subjects have been chosen (8 papers). In addition, we used the EPC Community conference proceedings and examined each conference proceeding available. With a similar strategy as previously explained the amount of hits has been too high (65 papers). Therefore, we adjusted our relevance conditions, so that articles that deal not explicitly with either syntactic, semantic or pragmatic quality aspects were excluded. Additionally, redundant contributions were excluded. With this strategy, also applied at the result set of the IWi reports, three relevant EPC Community papers and two IWi reports could be found. To complement our literature analysis, we used the academic search engine Google Scholar with queries such as “EPC syntax” (70 papers), “semantic syntactic pragmatic consistency EPC” (196 papers) or “modelling conventions business process” (213 papers). After applying our conditions for suitable papers, eleven articles have been chosen. Besides this process, we found relevant literature with the “Method of concentric circles” (Snowball system) [K99]. This method consists in evaluating the references of our already selected literature in order to identify often quoted, thus meaningful papers like [Ga10].

3 Quality Aspects of EPC Models

3.1 The SIQ-Framework applied to EPC Models

Quality aspects for EPC models (other types of models as well) can be divided into different areas. As already mentioned, we only considered articles providing aspects that could be mapped in the three main categories of the SIQ-Framework (Figure 2): *Syntactic*, *semantic* and *pragmatic* quality.

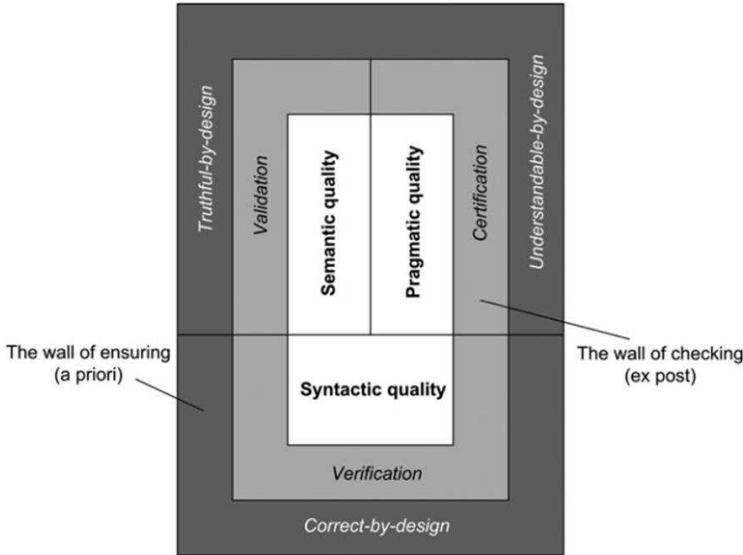


Figure 1: SIQ-Framework [RMR10]

We follow the SIQ-Framework conceptualization of these quality categories [RMR10]:

1. *Syntactic quality* is evaluated by the degree of conformance to a previously defined syntax. Any language defines rules for elements and connections between elements that need to be followed to create a model in this language.
2. *Semantic quality* refers to how well a model is representing the object under consideration. Two sub-goals exist: *completeness* and *validity*. A model is valid if all of its statements are correct and have a relevant reference to the underlying problem. The model is complete if it not only contains relevant references that are correct but also the references that *would be* correct.
3. *Pragmatic quality* is basically a matter of how good a model can be understood by its users. Whilst users might understand a model quite well, it could be still of low semantic quality by leaving out important characteristics of the underlying problem and vice-versa.

The SIQ-framework implicitly shows the relations between the different criteria. For example, syntactic quality is fundamental to semantic and pragmatic quality which is indicated by the fact that the latter two are based on the former. In Figure 2, this is depicted graphically by placing both *semantic quality* and *pragmatic quality* in rectangles on top of the rectangle *syntactic quality*.

3.2 Syntactic Quality

3.2.1 Representation

The notation of the concepts of the EPC is stated by means of Figure 2.

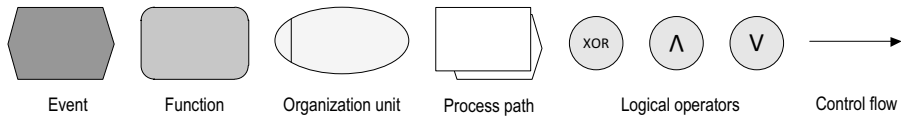


Figure 2: EPC Symbols, c.f. [Sc98, p. 19]

To begin with, there are syntactic rules whose non-conformance does not necessarily lead to a “bad” EPC. So, slight variations in the representation of the EPC elements are possible provided that these are still identifiable (cf. Organization unit in Figure 2). Another possible variation exists in the presentation of the logical operators [St06]. Staud proposes a graphical presentation where the logical operators are modelled as a split circle:

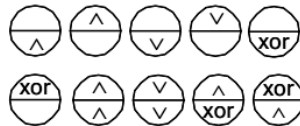


Figure 3: Alternative logical operators [St06]

The upper half represents the incoming control flows and the lower half represents the outgoing control flows. If there is just one out- or incoming control flow the respective half does not show an operator. Furthermore, operators splitting a single control flow to multiple ones are called a *split-operator* and operators joining multiple control flows are called *join-operator* [NR02].

3.2.2 Usage and Composition

In addition to the correct representation of model elements, additional rules are considered with the correct use and composition of these model elements. These rules are dealing with the correct connection between events and functions or the correct usage of logical operators. In the following section these rules will be introduced.

- (1) There is at least one start-event and one end-event [LM09; NR02; St06; KKS04; Ga10].
- (2) Events are able to trigger functions and functions have to be triggered by events. In other words: An event will be followed by a function and a function by an event. The only exception is the start- and end-event. The connection between events and functions are shown as a dashed arrow line [St06; KNS92; NR02; Ri00; KK05; Ga10].
- (3) Information objects and Organization units must be connected with functions. The connection is modelled with a dashed line (organization units) respectively with an arrow (information objects) [KNS92; St06]. An extension of this rule additionally requires a function involving human interactions to be associated with an organisational unit. Analogously, a function that does not require any human participation should be associated with an application system [KK05].
- (4) A logical operator connects several events and functions. The join operator needs to be of the same type as the logical operator that initially split the control flow [St06; KK05; NR02; Ga10].
- (5) Rule number 4 leads to the conclusion that logical operators have at least one incoming and one outgoing control flow arc [LM09]. Different sources expand on this: Split and join operators have either one incoming but multiple outgoing control flows or they have one outgoing but multiple incoming control flows [NR02; St06; KKS04; Ri00; Ga10].
- (6) Furthermore, because of the notation that just logical operators are able to connect one event/function with multiple ones (see rule 4), the guideline that events and functions possess just one incoming and one outgoing control flow arc can be deduced. The only exception is the start- and end-events just possess one arc [NR02; Ga10].
- (7) After an event an OR- or XOR-operator must not follow for the next functions. This results from the lack of decision-making power of an event [St06; NR02; KK05; KKS04; Ri00; Ga10]. But there are two papers mentioning that this opinion is not broadly shared [KKS04; KK05].
- (8) Process pointers are connected with events only [KKS04; NR02; St06].

3.3 Semantic Quality

3.3.1 Linguistic Correctness

An EPC model is linguistically correct if every label of the model elements follow a specified naming convention. The resulting restrictions for the creator of the model aim

at preventing misinterpretations. At first the German naming convention will be presented. The events of an EPC are representing the current state, so the linguistic correctness requires the label to be created from a substantive and a verb in the past participle form. An EPC function represents the active and time consuming part of an EPC model. Thereby a function's name should be created from a substantive and an active. Objects like an organization unit emulate objects of the "real world" and have to be titled with one or compound substantives (e.g. Business Management) [Ba10]. The English naming convention suggestions for model elements differ from the German conventions. In English, events have to be modelled in a substantive form plus a verb in past participle, functions consists of a verb and a substantive [HF09]. The naming conventions are summarized in Table 1.

Table 7: Name convention for events and functions [Ba10]

	Function	Event
German	Substantive(s) + Verb (e.g. Bestellung bearbeiten)	Substantive(s) + Verb in Past Participle (e.g. Bestellung bearbeitet)
English	Verb + Substantive(s) (e.g. Processing order)	Substantive(s) + Verb in Past Participle (e.g. Order processed)

Additionally, to these generally accepted naming conventions, HUMM and FENGEL mention enterprise- and business-specific conventions. Accordingly, naming conventions of EPC model elements, which capture a rather enterprise-specific process, should conform to the according vocabulary of the enterprise (e.g. "Rent vehicle" instead of „Rent car“) [HF09].

3.3.2 Formal Semantics

Due to the possible automatic execution of EPC models, formal semantics require such models to ensure formal correctness, namely to avoid endless loops (livelock) or the unexpected halt of the execution (deadlock) [St06]. These two semantic mistakes will be shown in Figure 4. The model can be started with event E1 being activated resulting in a token which will be passed to function F1. After that function, the AND-split-operator delivers tokens to both control flows. So between the first event E1, the first function F1 and the AND-operator, a livelock occurs. In the subsequent control flow, the XOR-operator delivers just one token to one event, based on the decision in F1. But the following AND-operator expects both tokens, so the end-event will never be reached and a deadlock occurs [GLK].

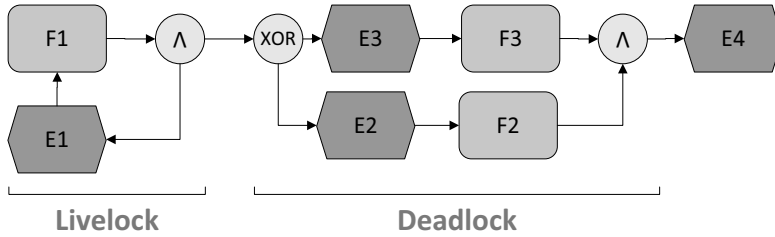


Figure 4: Deadlock and Livelock in an EPC model

A further criterion regarding the semantic correctness is *soundness*. It originally has been created for workflow nets and comprises three criteria: “option to complete”, “proper completion” and “no dead transitions”. Since these three requirements were designed for workflow nets, where just one source and one sink exist, the soundness requirements cannot be applied to EPC directly since they allow for multiple start- and end-events. Hence a slight variation is necessary which has led to the notion of *EPC soundness* [ME09]. EPC soundness also comprises three criteria: (i) a set of start-events exists so that the EPC is not a cycle and (ii) for every possible combination of activation of start-events proper completion is guaranteed, i.e. regardless of which start-event combinations are activated, the model terminates. Furthermore, (iii) the third point is derived from the *relaxed soundness*. For a particular set of final nodes the model has to be sound, i.e. every end-event in the model that is accessible from the start-events must be able to receive a final token. If this requirement is violated, the proper completion would not be guaranteed in the case that all start-events are activated. A taxonomy of soundness terms can be developed by the consideration of the different sets of anomalies. These sets of anomalies can be detected with criteria such as soundness, relaxed soundness and EPC soundness. While the soundness criteria are able to identify all errors, e.g. deadlocks, in an EPC model with just one start and one end node, the relaxed soundness criteria just demands the proper use of the soundness criteria at a particular part of the model (The part that is the desired behavior of the modeler). Therefore it is not possible to identify all errors in the model. The second soundness variation, the EPC soundness, deals with the case of multiple start- and end nodes. So by applying the EPC soundness criteria, it is possible to identify the errors in a model that occur for all possible start combinations. It could therefore be concluded that the original soundness term is the strictest criterion, followed by the EPC soundness and the relaxed soundness as the weakest criterion. Thus, soundness subsumes EPC soundness which in turn subsumes relaxed soundness. Beyond soundness, GRUHN and LAUE present different additional types of semantic anomalies and suggest solutions that will be presented in the following [GL09].

- (1) If two events occur before or after a join-operator with the identical content, the model often can be simplified. However, if the join-operator is a XOR-operator it shall be presumed that a modelling mistake occurred because a XOR-operator merely should be used for mutually exclusive events.

- (2) If an AND- or OR-operator is followed by two events whose content is mutually exclusive, it is a logical mistake because events after an AND- or OR-operator can occur together. Mostly a XOR-operator is the right choice.
- (3) A XOR-operator is followed by three events and two of them are their negation to each other. Therefore, the third event would eventually be needless because a XOR-operator normally models the decision between mutually exclusive events.
- (4) If an operator is used to compare a value (e.g. with a XOR-operator), the issue is often forgotten that the value remains constant.
- (5) If a function models a polar question that just can be answered with yes or no, the function has to be followed by an XOR-operator.

In an older publication GRUHN and LAUE elaborated on further aspects (styling rules) [GL05].

- (6) Functions and events of an EPC model must not be instantiated more than once, i.e. for example a function which is already active must not be activated by another token. Therefore it should be ensured that a control flow arc cannot be reached by several tokens.
- (7) XOR-joins do not block if multiple tokens are possible to arrive at the join, rather the incoming token will be passed to the outgoing arc. This also corresponds to the intention of EPC modellers in the practice, because they assume that anyway just one token will arrive.
- (8) Each split-operator must have a corresponding join-operator. Such models are also called *structured models*.

Other criteria are specified by GRUHN, LAUE, KERN and KÜHNE [GLK08] with reference to notion of soundness that has been introduced originally by VAN DER AALST.

- (9) Of each EPC model element that can be reached from the start-event it must be possible to reach an end-event.
- (10) If a model element does not have a following element, it must be an end-event.
- (11) There must be no element in an EPC model, for which there is no control flow from a start-event to an end-event.

In addition to these EPC-specific semantic rules, the literature often just discuss about concepts to transfer an EPC to a Petri net in order to detect anomalies relating to the formal semantics. Since these approaches do not focus on checking the semantic quality of EPC models directly, they are not covered in this paper due to space limitations.

3.3.2 Compliance

Beyond formal semantics, in the area of compliance checking, the correctness of the *models content* (i.e. what is represented in the model) is of central importance. Compliance can be understood as the conformity of something such as a process model to the entirety of relevant legal liabilities, directives and rules as well as to the internal guidelines and best practices of an enterprise [WL08]. Compliance rules can result from scenario-specific, project-specific or general requirements [DF09]. Often, compliance rules target the behavioural aspect of a model that is all possible execution traces. An example of such a compliance rule is: “After the decision that a service is free of charge, no fee calculation can take place”.

- (1) The behaviour of a model should comply with rules specified w.r.t the semantics of individual model elements [SPH04; SM06].

Further distinctions can be made (a) between compliance rules focusing on the *occurrence* of specific nodes in a process graph or the structure of the model and (b) regarding the type of nodes in the process graph between basic flow elements such as functions and events and elements representing resources such as organizational units [FHT11]. These distinctions lead to element flow rules, resource usage rules, element occurrence rules and resource occurrence rules [FHT11]. An example of a simple element flow rule is: “Within 2-3 process steps after order rejection, the customer has to be notified about the decision”.

- (2) The occurrence of model elements and the connections between model elements in an EPC model should comply with rules specified w.r.t the semantics of individual model elements [FHT11].

The attributes of model elements may also be the target of compliance rules [DF09]. An example of such a compliance rule is: “For every object which requires persistence, modifications have to be saved each time the document is read”.

- (3) The model should comply with rules covering both the model elements and their attributes [DF09].

3.4 Pragmatic Quality

An important criterion leading to a high pragmatic quality of an EPC model is the ease of understanding of the model for the intended audience. So an EPC model will suffice this demand if it e.g. follows the guideline of clearness comprises by the GoM [BRS95]. Among others, these are included in the following points:

- (1) There should not be overlapping arcs in the generated EPC [Br11].
- (2) If possible, a change of the flow direction should be avoided. In the best case the flow direction goes from the top to the bottom across the whole model [Br11].

- (3) The naming conventions should be applied [Br11].
- (4) Not every function in an EPC model receives an assignment to an organization unit. A new correlation should be done if the organization unit changes [BRS95].
- (5) Very large process chains should be divided with the aid of process pointers in order to promote clarity [St06].

Furthermore, there are a few rules given by [MRA10] that are aiming to a higher pragmatic quality. The relevant guidelines are listed as follows:

- (6) Use as few elements as possible. The bigger the model, the more confusing (and error-prone) it is.
- (7) Minimize the amount of control flows of an element. The more incoming and outgoing control flow arcs there are, the harder it is to interpret the model.
- (8) If possible, use just one start- and one end-event. In addition to the reduction of possible mistakes, it increases the intelligibility.
- (9) Design the EPC model as structured as possible, e.g. by reuniting every split-operator with its corresponding join-operator.
- (10) Avoid OR-operators because of their ambiguity in the semantics.
- (11) Apply the verb-object naming convention in order to prevent ambiguities.
- (12) Split your model if it contains more than 50 elements.

4. EPC Quality Aspects reflected in the Literature

In the following the quality aspects will be reflected based on the identified literature. The explicit consideration of syntactic aspects by the different authors is summarized in Table 8.

Table 8: Syntactic Quality Aspects in Literature

<i>Citation</i>	<i>Authors Names</i>	<i>Syntactic Aspects</i>							
		<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>
[Ga2010]	GADATSCH	•	•		•	•	•	•	
[KK05]	KAHL, KUPSCH			•	•	•			•
[KNS92]	KELLER, NÜTTGENS, SCHEER		•	•					
[KKS04]	KLEIN, KUPSCH, SCHEER	•				•		•	•
[LM09]	LAUE, MENDLING	•				•			
[MRA10]	MENDLING, REIJERS, VAN DER AALST								
[NR02]	NÜTTGENS, RUMP	•	•		•	•	•	•	•
[Ri00]	RITTGEN		•			•		•	
[St06]	STAUD	•	•	•	•	•		•	•

As it can be inferred the literature is to a certain degree clear about the syntactical aspects of an EPC model. A unique position is hold by rule number six, as this rule is only mentioned by two authors. However, rule number six is often implicitly given, as it can be deduced from other rules.

Table 9: Semantic Quality Aspects in Literature

Citation	Authors Names	Semantic Aspects															
		Naming		Formal Semantics										Compliance			
		D	E	1	2	3	4	5	6	7	8	9	10	11	1	2	3
[Ba10]	BARTSCH	•						•	•		•	•	•	•			
[DF09]	DRAWEHN, FEJA															•	•
[FHT11]	FELLMANN, HOGREBE, THOMAS														(•)	•	
[GL05]	GRUHN, LAUE									•	•	•					
[GL09]	GRUHN, LAUE				•	•	•	•	•								
[GLK08]	GRUHN, LAUE, KERN, KÜHNE												•	•	•		
[HF09]	HUMM, FENGEL		•														
[SM06]	SIMON, MENDLING															•	•
[SPH04]	SPECK, PULVERMÜLLER, HEUZEROTH															•	•

The coverage of semantic aspects is summarized by Table 3. In the column “Naming”, “D” and “E” represent German or English naming conventions. There is only a small overlap between the papers considered because there are no fixed EPC conventions in the area of semantics in particular. The coverage of pragmatic aspects is summarized by Table 4.

Table 10: Pragmatic Quality Aspects in Literature

<i>Citation</i>	<i>Authors Names</i>	<i>Pragmatic Aspects</i>									
		<i>1</i>	<i>2</i>	<i>3/11</i>	<i>4</i>	<i>5/12</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>
[BRS95]	BECKER, ROSEMAN, SCHÜTTE					•	•				
[Br11]	BRÜCK		•	•	•						
[MRA10]	MENDLING, REIJERS, VAN DER AALST				•		•	•	•	•	•

It is possible to summarize rule three and eleven as well as five and twelve according to their content. Through the presence of generally accepted notations for model design pragmatic aspects are barely discussed in literature. It is possible to deduce every rule mentioned in section 3 from the guideline of clearness stated in [BRS95] and [MRA10]. Overall, extensions like the eEPC, the SEQ-Operator [P95] or ET-Operator [R96] and the discussion about their implications for correctness and quality are disproportionately underrepresented in literature. Up to now, discussion about correctness has astonishingly just referred to the “simple” EPC with events, functions and operators.. In addition to the results apparent from the tables, we noticed that the events in the debate about correctness criteria are less important. They are playing a key role regarding the syntactic quality but while discussing the semantic and pragmatic quality of EPC models they are almost not even mentioned. Out of this observation we can point out two ideas for further modelling rules: Prospective modelling languages may not use elements similar to EPC events and insignificant events in EPC models might be abolished. This rule would just be applied on trivial events like “Check contract” – “Contract checked” and could both increase the pragmatic and semantic quality.

5. Some Considerations on Standardisation

As the use of EPCs in the practice is widespread, standardisation seems beneficial in order to support the users at modelling correct models and facilitate the implementation of conformant software tools. Especially, a standardisation is required for domain- or subject-specific extensions of the EPC. As far as it concerns the syntax, respectively the syntactic quality, there mostly is consensus among the authors regarding syntactical rules an EPC models has to follow. The problem is more dramatic regarding the semantics of an EPC model. Since the EPC was firstly introduced in 1992 without a complete semantic specification, nowadays there partly is a disagreement about the interpretation of different EPC elements (e.g. does an XOR-operator allow receiving several tokens, how the OR-operator does know when he has received all required

tokens etc.). Above all there is the discussion whether some rules restrict the freedom of modelling the EPC provides. Hence, the literature often looks for a way out by Petri nets whose semantic is sufficiently formalized and defined. The field of pragmatic quality is less strongly discussed in recent articles, presumably because some general guidelines have been published whose compliance already leads to a high pragmatic quality like the „Generalities of Orderly Modelling“ [BRS95] or the „Seven Process Modelling Guidelines“ [MRA10]. Although these were not only published specifically for EPC models, they can easily be applied to it. So ultimately, in order to increase the utility of the EPC as a language for business process modelling, a standard seems to play an important role. To name just two benefits from such a standardisation, first the cooperation between enterprises by means of the EPC would be increased, as common concept will have a common meaning and secondly, the derivation of software artefacts based from an EPC model in the sense of model-driven software engineering would be beneficial as well, as the more semantically enriched foundation tackles both: the humans that execute the business process and the technology that support and automate the execution. However, it is not likely that an “one size fits all”-approach to standardization will significantly increase the utility of the EPC and related tools. Instead, it can be assumed that different project requirements (e.g. documentation, re-engineering or automation) and modeller skills (e.g. novice modeller, intermediate or expert) will require different priorities and levels of conformance that can be consolidated to different EPC profiles in the future.

5. Conclusion

We have investigated the state of syntactic, semantic and pragmatic quality of EPC-modelling by conducting a literature analysis. The results show that there is some consensus regarding syntactic aspects, but less regarding semantic and pragmatic aspects. Finally, we advocate for the standardisation of the EPC which should reflect the needs and purposes in different modelling contexts and by different groups of modellers.

Acknowledgement: The research presented in this paper is part of the WISMO project and is funded by the German Federal Ministry of Education and Research (BMBF) within the framework concept “KMU-innovativ” (grant number 01IS012046B).

References

- [Ba10] Bartsch, M.: Realisierung der semantischen Annotierung für pModeler, einem System zu semantischen Prozessanalyse. Linz: Johannes Kepler Universität Linz – Institut für Wirtschaftsinformatik, 2010
- [BNT10] Becker, Jörg; Niehaves, Björn; Thome, Irina: How Many Methods Do We Need ? – A Multiple Case Study Exploration into the Use of Business Process Modeling Methods in Industry. In: AMCIS 2010 Proceedings, 2010, p. 534.
- [Br11] Brück, S.: Modellierungskonventionen – Regeln bei der Erstellung eines Modells. Saarbrücken: Universität des Saarlandes: Institut für Wirtschaftsinformatik, 2011

- [BRS95] Becker, Jörg; Rosemann, Michael; Schütte, Reinhard: Grundsätze ordnungsmäßiger Modellierung. In: Wirtschaftsinformatik 37 (1995) 5, pp. 435–445.
- [DF09] Drawehn, J.; Feja, S.: Anwendung von grafischen Validierungsregeln bei der Entwicklung von IT-Integrationsprozessen. In: Münch, J.; Liggesmeyer, P. (Hrsg.): Tagungsband des Workshops "Modellgetriebene Softwarearchitektur – Evolution, Integration und Migration" (MSEIM 2009), 3. März, Kaiserslautern. Bonn : Köllen (GI LNI P150), 2009, S. 367–375
- [FHT11] Fellmann, M.; Högbe, F.; Thomas, O.: Semantic Verification of Business Process Models: Prospects and Limitations. In: Smolnik, S.; Teuteberg, F.; Thomas, O. (Hrsg.): Semantic Technologies for Business and Information Systems Engineering: Concepts and Applications. IGI Global, Hershey PA, USA, 2011, pp. 150–168
- [Ga10] Gadatsch, A.: Grundkurs Geschäftsprozessmanagement Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. Vieweg+Teubner, Wiesbaden, 2010
- [GL05] Gruhn, V.; Laue, R.: Einfache EPK-Semantik durch praxistaugliche Stilregeln. In (Nüttgens, M.; Rump, F.J., Hrsg.): Geschäftsprozessmanagement mit Ereignisgesteuerten Prozesskette. Hamburg, 2005, S.176–189
- [GL09] Gruhn, V.; Laue, R.: Ein einfaches Verfahren zur Erkennung häufiger Fehler in EPKs. Universität Leipzig: Lehrstuhl für Angewandte Telematik und E-Business, 2009
- [GLK08] Gruhn, V.; Laue, R.; Kern, H.; Kühne, S.: EPK-Validierung zur Modellierungszeit in der bflow* Toolbox.. Universität Leipzig
- [HF09] Humm, B. G.; Fengel, J.: Assessing Semantic Consistency of Business Process Models. Hochschule Darmstadt: Zentrum für Forschung und Entwicklung, 2009
- [HFL11] Houy, Constantin; Fettke, Peter; Loos, Peter; Aalst, Wil M. P.; Krogstie, John: Business Process Management in the Large. In: Business & Information Systems Engineering, 2011, vol. 3, no. 6, pp. 385–388
- [K99] Krämer, Walter: Wie schreibe ich eine Seminar- oder Examensarbeit? Frankfurt a.M.: Campus 1999
- [KK05] Kahl, T.; Kupsch, F.: Transformation und Mapping von Prozessmodellen in verteilten Umgebungen mit der ereignisgesteuerten Prozesskette. Saarbrücken: Institut für Wirtschaftsinformatik, 2005
- [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Saarbrücken: Institut für Wirtschaftsinformatik, 1992
- [KKS04] Klein, R.; Kupsch, F.; Scheer, A.-W.: Modellierung inter-organisationaler Prozesse mit Ereignisgesteuerten Prozessketten. In (A.-W. Scheer, Hrsg.): Veröffentlichung des Instituts für Wirtschaftsinformatik. Heft 178. 2004
- [LM09] Laue, R.; Mendling, J.: Structuredness and its significance for correctness of process models. Springer-Verlag, Berlin, 2009
- [ME09] Mendling, J: Empirical Studies in Process Model Verification. Heidelberg u.a., Springer, 2009
- [MRA10] Mendling, Jan; Reijers, Hajo A.; van der Aalst, Wil M.P.: Seven process modeling guidelines (7PMG). In: Information and Software Technology 52 (Feb. 2010) 2, pp. 127–136.
- [NR02] Nüttgens, M.; Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: Promise 2002 – Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen. Potsdam, 2002, S.14
- [P95] Priemer, J.: Entscheidungen über die Einsetzbarkeit von Software anhand formaler Modelle. Sinzheim: Universität Münster 1995