DOI: 10.15439/2016F328
ACSIS, Vol. 8. ISSN 2300-5963

Overview of Time Issues with Temporal Logics for Business Process Models

Krzysztof Kluza, Krystian Jobczyk, Piotr Wiśniewski, Antoni Ligęza
AGH University of Science and Technology
al. A. Mickiewicza 30, 30-059 Krakow, Poland
E-mail: {kluza,wpiotr,ligeza}@agh.edu.pl

***Abstract*—Process models can specify various aspects of business processes. In this paper, we present an overview of the existing solutions for describing time aspects of such models. We focus on Business Process Model and Notation and provide examples of representing time patterns in this notation. As temporal issues can be specified using temporal logics, we provide a short overview of selected temporal logics which can be used to specify the time patterns in business process models.**

***Index Terms*—BPMN, Business Processes, Temporal Logics, Temporal Issues, Time Patterns**

I. Introduction

BUSINESS Process Management (BPM) [1] is a modern approach to improving organization's workflow, which focuses on reengineering of processes to obtain optimization of procedures, increase efficiency and effectiveness by constant process improvement.

The key aspect of BPM is a Business Process (BP). A BP is usually described as a collection of related activities which transform different kinds of clearly specified inputs to produce a customer value, mainly considered as products or services and organizational goals, as output [2]. Such a process can be represented as a BP model [3]. However, there can be many representation methods for modeling processes. We give an overview of the existing representation methods for business processes, such as Petri nets, EPC, IDEF3, UML AD, YAWL and BPMN diagrams.

As temporal logics can have various applications related to knowledge management [4]–[6] and business process models [7], it is important to consider process representation from the perspective of time issues. Thus, we analyze the presented process representations in terms of time-related elements. The main focus is on the BPMN notation and time issues in this notation. We present how to represent the Allen's algebra relations using BPMN notation. As it was proven empirically in [8], representing relations using the 2D models is efficient in terms of understanding temporal aspects of time intervals.

Moreover, based on the existing time patterns from the literature, we describe how they can be used in BPMN as well as we provide a short overview of temporal logics which can support these time patterns.

The rest of this paper is organized as follows: Section II presents an overview of process modeling notations with with an emphasis on time or temporal aspects.. The most popular BPMN representation is analyzed in details in this context in Section III. In Section IV, we give an overview of temporal logics which can be used to support time patterns for process models. The paper is summarized in Section V.

II. Business Process Representation

As process modeling is an essential part of BPM, in the following subsections, we present the most popular business process representations.

Although there are many process modeling languages, we focus here on the six visual and most successful ones: EPC, IDEF3, UML AD, Petri net, YAWL and BPMN. Processes in these languages consist of activities, which may be decomposed to subactivities. The order of activities defines the sequence of work. In the lower level, each activity transforms some inputs into outputs.

Table I presents a simple yet illustrative example of a car rental process [9] in the above mentioned notations. The process starts with a registering request, and then extra insurance can be added. When check-in is initiated, the customer can select a car; at the same time the driver's license is checked and the customer's credit card is charged. The process ends when the chosen car is supplied to the customer. This exemplary process contains only basic control flow elements, which can be represented in all of these languages. However, it is important to mention that the expressiveness of each of these languages is much higher than the required for this example.

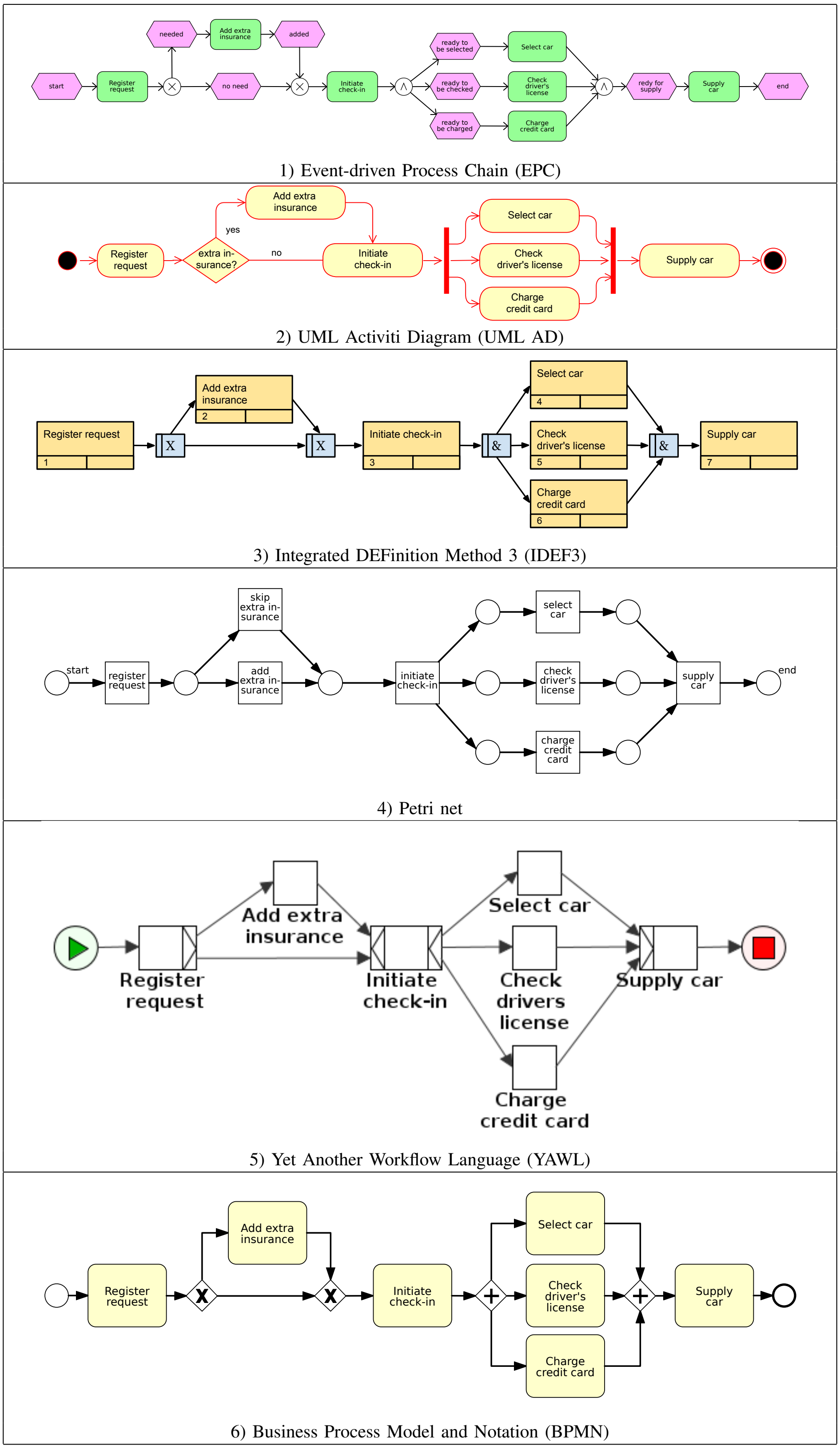
A. Process Modeling Languages

1) Event-driven Process Chain (EPC): Event-Driven Process Chain (EPC) is a simple graphical modeling language for modeling processes introduced by August-Wilhelm Scheer [10]. The EPC models are directed graphs visualizing the control flow [11]. Each EPC consists of events, functions and connectors, starts with at least one event and ends with at least one event, In EPC, each event triggers a function that leads to a new event. The notation supports three types of connectors (AND, XOR, OR) which can be used to model splits and joins.

As events in EPC are rather passive elements which specify under what circumstances a process works or what is a result state of a process, such constructs are rather not suitable for time-related extensions.

The paper is supported by the AGH UST grant.

Table I
AN EXAMPLE OF CAR RENTAL PROCESS IN VARIOUS PROCESS MODELING NOTATIONS



1) Event-driven Process Chain (EPC)

2) UML Activiti Diagram (UML AD)

3) Integrated DEFinition Method 3 (IDEF3)

4) Petri net

5) Yet Another Workflow Language (YAWL)

6) Business Process Model and Notation (BPMN)

2) Integrated DEFinition Method 3 (IDEF3): Integrated Definition (IDEF) is a family of modeling languages that arose in the 1979s out of the U.S. Air Force in order to increase manufacturing productivity [12], [13]. The IDEF suite contains sixteen types of modeling languages in the field of systems and software engineering; however, the majority is still under development and only IDEF0-4 are commonly used in practice. IDEF3 is concerned with modeling the processes of a business or its systems [14], [15]. IDEF3 Process Flow diagram consists of such elements as Unit of Behavior (UOB) boxes, precedence links, junctions (AND, XOR, OR), referents, and notes. An IDEF3 process description organizes the network of relations between situations in a specified scenario from the process-centered perspective.

As in IDEF3, events are not modeled directly, there are no time events. However, IDEF3 specifies constraint precedence links which express simple temporal precedence relations between instances of one UOB and another UOB. They add constraints over and above the activation semantics of simple precedence and can indicate that an instance of the source UOB must be followed or preceded by an instance of the destination UOB [16].

3) UML Activiti Diagram (UML AD): The Unified Modeling Language (UML) [17] from the Object Management Group (OMG) constitutes a standardized notation for modeling software applications [18]. This multipurpose modeling language offers a variety of notations to capture different aspects of software [19], [20]. UML has become the dominant notation among software engineers and attempts to be a universal visual notation for software design. Activity diagram (AD) is a kind of the UML behavior diagrams for modeling dynamic aspects of the system, which focuses on the flow of activities involved in a single process and shows the dependencies among them. Although UML AD can be used for process modeling purposes, its complex nature makes it a barrier for non-technicians and it is not suitable for all aspects of this type of modelling. A simple process consists of a sequence of nodes modeled using control-flow and object-flow. The control-flow comprises two types of nodes: action nodes (activities to be performed or signals to be received/sent by the process) and control nodes, which model sequencing and parallel or alternative branching. The flow of data between nodes can be represented by associations of object nodes with activities.

In the case of UML AD, a time event trigger is supported (notated with an hour glass symbol). On the other hand, UML semantics do not dictate the amount of time between actions or events. However, additional timing constraint elements can be defined using customized stereotypes or specified in OCL (Object Constraint Language), especially as pre- and post-conditions for actions.

4) Petri net: Petri nets [21] offer a graphical notation for modeling processes that include choice, iteration, and concurrent execution. A classical Petri net is a directed graph composed of two types of nodes: places and transitions. An arc in the net may connect a place to a transition or vice versa, but no arc may connect two nodes of the same type. A transition can have a number of input and output places. As Petri nets have an exact mathematical definition of their execution semantics and a well-developed mathematical theory for process analysis, they are suitable for modeling, analysis and simulation of dynamic systems. As generally the execution of Petri nets is nondeterministic, they are well suited for modeling the concurrent behavior of distributed systems [22].

Although the original Petri nets do not model time issues explicitly, there are extensions, such as time Petri nets, which were used in analysis of concurrent systems where behavior was dependent on explicit values of time [23]. A detailed overview of the algorithms that allow for analysing time-dependent Petri nets can be found in the book [24].

5) Yet Another Workflow Language (YAWL): YAWL is a Petri Net based workflow language [25], [26]. Having formal semantics, it supports specification of the control flow and the data perspective of business processes. The language encompasses workflow patterns to guarantee language expressibility [27]. The YAWL language extends the class of workflow nets with multiple instances, OR-joins and cancelations. Workflow net (WF-net) [28] is a subset of Petri net used to model workflows. In a WF-net, there is a unique input place and a unique output place and every other place and transition are on a directed path from input to output place, and WF-nets introduce additional notations for joins and splits (AND and XOR). In the case of YAWL, in contrast to Petri nets and WF-nets, its syntax allows tasks to be directly connected, which helps compress the visual representation of a YAWL model.

Although in YAWL there are no separate time elements, any atomic task can be assigned a timer behaviour using a timer property. Such timer can be activated either when a task is enabled (i.e. is offered or allocated) or when it starts. What is more, there are also additional timer predicate expressions which can be used as flow predicates.

6) Business Process Model and Notation (BPMN): BPMN [29], adopted by the OMG group, is the most widely used notation for modeling BPs. The BPMN notation uses a set of models with predefined graphical elements to depict a process and how it is performed. Although the current BPMN 2.0 specification [29] defines three models to cover various aspects of Business Processes, in most cases, using only the Process Model is sufficient. Thus, in this paper we analyze the internal Business Process Model of BPMN, which can be compared to the previously presented approaches.

Table II presents an evaluative analysis of the described Business Process modeling languages (the summary prepared on the basis of [30]–[35]). The symbols in the tables have the following meaning: ❍ – not supported, ● – supported. ◗ – partially supported (not standardized or possible to present but not directly).

As one can see from the Table II, BPMN 2.0, a de facto industry standard for modeling processes, supports most of the listed elements. Thus, in the following sections we will focus on this notation and present time issues related to BPMN.

Table II
COMPARISON OF THE SUPPORTED ELEMENTS IN PROCESS MODELING LANGUAGES

		Business Process modeling languages					
		Petri net	**EPC**	**IDEF3**	**UML AD**	**YAWL**	**BPMN**
Year		1962	1992	1995	1997	2004	2004
Creator		C. Petri	A.-W. Scheer	U.S. Air Force	OMG	van der Aaalst	OMG
Background		Academic	Academic	Industry	Industry	Academic	Industry
Standardised		N.A.	No	Yes	Yes	No	Yes
Metamodel		❍	◗	❍	●	❍	●
Purpose	Formal methods	●	❍	❍	❍	◗	❍
	Graphical	◗	●	●	●	●	●
	Execution	❍	❍	❍	❍	●	●
Activities	Atomic	◗	◗	◗	●	●	●
	Subprocess	◗	◗	◗	●	●	●
Events		●	●	❍	●	●	●
Gateways	AND	◗	●	●	●	●	●
	XOR	◗	●	●	●	●	●
	OR	❍	●	●	●	●	●
	Complex	❍	❍	❍	◗	◗	●
Participants	Internal	❍	●	❍	●	◗	●
	External	❍	❍	❍	●	❍	●
Data	Data flow	❍	●	❍	●	◗	●
	Data objects	❍	◗	◗	●	◗	●
	Data repository	❍	❍	❍	●	❍	◗
Time-related	Events	❍	❍	❍	●	◗	●
	Activities	❍	❍	◗	◗	◗	◗
	Gateways	❍	❍	❍	◗	◗	◗

III. TIME ISSUES IN BUSINESS PROCESS MODELS

A. *Time Representation in BPMN*

If it comes to the time-related issues in BPMN, some of them can be represented directly in the following way:

- Timers are supported as Timer start and Timer intermediate events (see Figure 1) For such an element, one of the following properties can be specified:
  - time date – specifies a fixed date when trigger will be fired,
  - time duration – specifies how long the timer should run before it is fired, or
  - time cycle – specifies repeating interval, which can be useful for starting process periodically, or for sending multiple reminders for overdue user task.
- Event-based Subprocess are subprocesses which are started by an event, such as a timer (see Figure 2).
- Timer boundary events allow for following additional or alternative control flow when the timer is fired. This can either interrupt or not the associated task or subprocess (see Figure 3).
- Event-based gateway with timers works like an exclusive gateway (XOR) as both involve one path in the flow (see Figure 4). In the case of an event-based gateway, however, it is evaluated which event has occurred, not which condition is being met.

Apart from the time issues which are basically supported by the existing BPMN elements, some issues can be modeled indirectly using more complex combination of elements. In Table III, we presented the classic Allen's algebra relations [36] applied to BPMN models. On the left hand side there is a simple model with additional artefacts depicting time relations. On the right hand side, we present a refined model, compliant with the BPMN 2.0 specification [29] which should fulfil the presented relation (this can be also interpreted as a method of imposing some constraints related to Allen's algebra relations). As there are many equivalences in BPMN [37], the presented models are not the only one that are possible. One can also noticed that in some cases, the model is quite complex and requires additional BPMN elements.

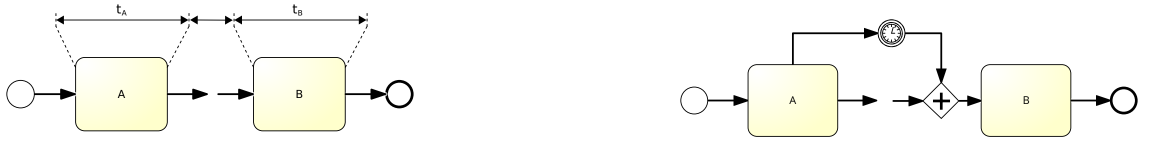
Simpler models for these relations can be found in [38] and [39]. However, they require additional non-standardized elements. Thus, they extend the BPMN notation.

A survey of some time-related aspects of process models conducted by Cheikhrouhou et al. can be found in [40], [41]. Their survey focused on the existing approaches to specifying and verifying temporal aspects of processes, not focusing on processes represented using the BPMN notation, but rather temporal constraints specification methods.

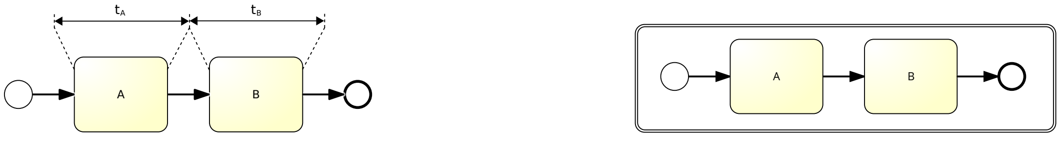
B. *Time Patterns in Process Models*

The objective of time patterns is to facilitate the analysis and comparison of Process-Aware Information Systems (PAIS). Their classification and selection criteria were presented in [42] and further developed in [38]. In [43], a dedicated tool for PAIS management and support was demonstrated. Time constraints related to specific patterns can be

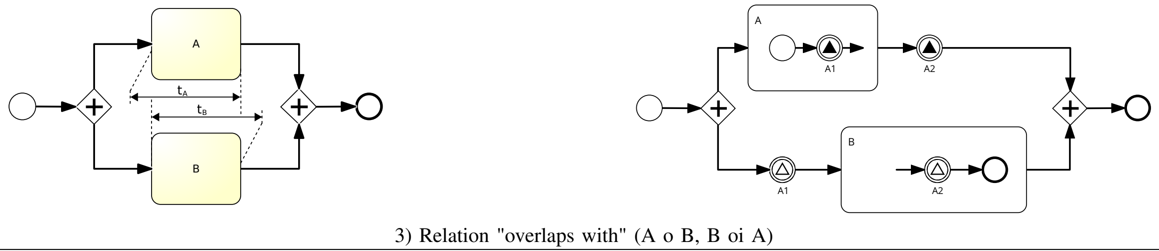
Table III
ALLEN'S RELATIONS IN BPMN: LEFT: STANDARD MODEL WITH TWO TASK (WITHOUT IMPOSED RESTRICTIONS), RIGHT: REFINED MODEL FULFILLING THE RELATION



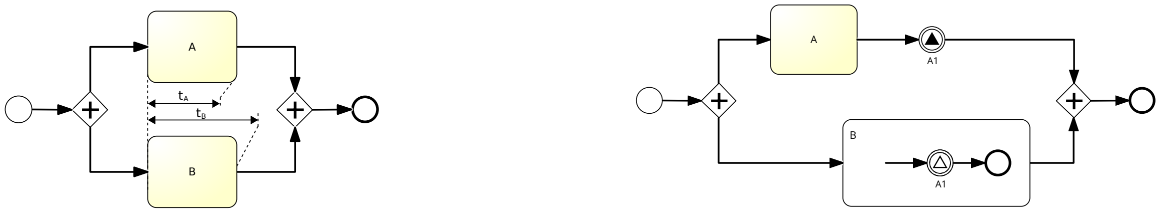
1) Relation "takes place before" (A < B, B > A)



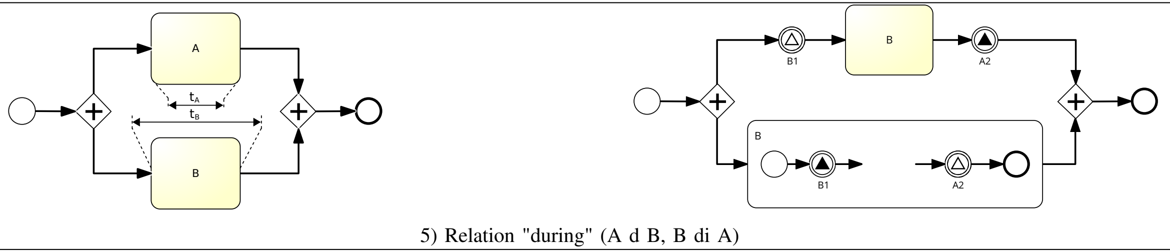
2) Relation "meets" (A m B, B mi A)



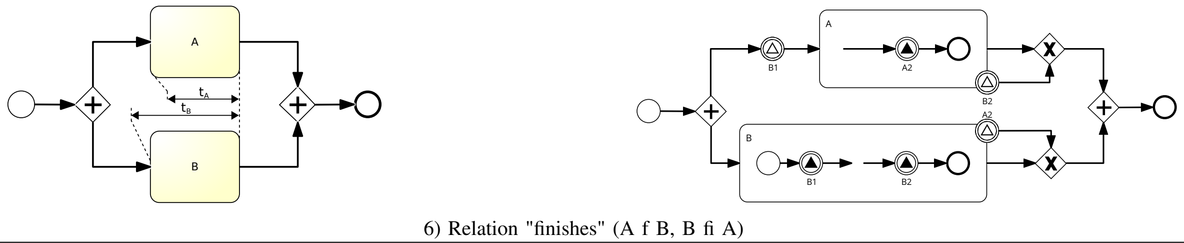
3) Relation "overlaps with" (A o B, B oi A)



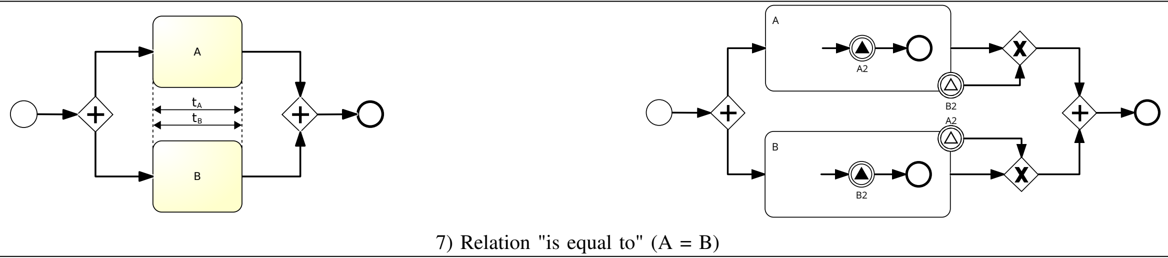
4) Relation "starts with" (A s B, B si A)



5) Relation "during" (A d B, B di A)



6) Relation "finishes" (A f B, B fi A)



7) Relation "is equal to" (A = B)

used in generating optimized plans for declarative process models [44], where a user determines the final purpose instead of an explicit task sequence. Another approach is to present temporal information along with resources using a modified Petri Net [45], where validity periods as well as maximal processing times are determined.

Thus, in [38], based on the representative set of business process models from different domains, 10 time patterns (divided to four categories) were identified and described:

I Duration and Time Lags
- TP1 Time Lags between two Activities
- TP2 Durations
- TP3 Time Lags between Arbitrary Events

II Restricting Execution Times
- TP4 Fixed Date Elements
- TP5 Schedule Restricted Elements
- TP6 Time-based Restrictions
- TP7 Validity Period

III Variability
- TP8 Time-dependent variability

IV Recurrent Process Elements
- TP9 Cyclic Elements
- TP10 Periodicity

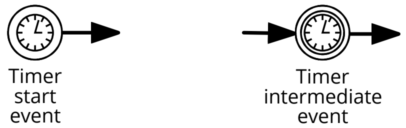


Figure 1. BPMN Timers

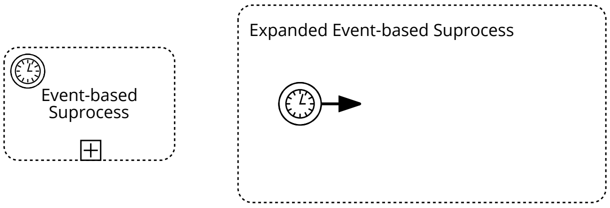


Figure 2. BPMN Event-based Subprocess

Each of these patterns can be used in various forms and applied to different elements (design choices). Thus, some of such choices of the patterns TP 1-4, 8 and 9 are supported by BPMN constructs as well:

- TP1 – supported by timers or combination of signals, as in diagrams in Table III,
- TP2 – supported by boundary timer events, as in Figure 3,
- TP3 – supported by timer intermediate event, as presented in Figure 1,
- TP4 – supported by timer start event, as in Figure 1 and 2,
- TP8 – supported by event-based gateway with timers, as in Figure 4,

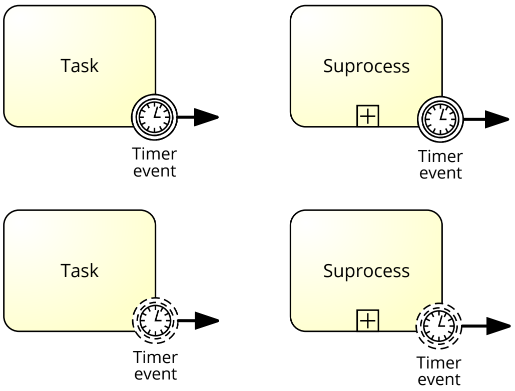


Figure 3. BPMN Timer boundary events

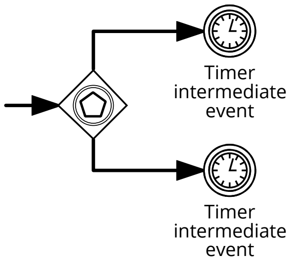


Figure 4. BPMN Event-based gateway with timers

- TP9 – supported either by a loop modeled using control flow with a timer intermediate event or by multi instance loop task or subprocess.

C. Analysis and Verification of Time Related Process Models

An important issue in process modeling is verification process models [46]–[48]. Among the existing papers concerning time issues in business processes, there are also papers related to analysis and verification of process and workflow models. There are several methods that can be used for validation of time related processes [49]–[56]. One of the existing methods concerns dynamic detection of temporal violations and providing possible solutions to a specific problem [49]. An analysis of time compatibility of web services with respect to time constraints, correlated with data and caused by message interaction between services, was conducted in [51]. As an extension to this solution a technique of analyzing and validating a set of processes was presented in [52]. It includes checking if a process choreography fulfils the declared time requirements and is based on Business Process Execution Language models which are then interpreted in the Fiacre formal verification language. Another method used to verify temporal constrains is based on new formal language, XTUS-Automata, which can be used to specify both relative and absolute time properties and also include deadlock verification [53]. This solution use temporal specification patterns, which are provided for different types of properties. In [54], a solution where an extended BPMN model is mapped onto timed automata and then verified using UPPAAL model checker is proposed. Temporal requirements

Table IV
COMPARISON OF THE EXISTING PROCESS VALIDATION METHODS

Authors	Du et al. [49]	Guermouche [51]	Guermouche et al. [52]	Kallel et al. [53]	Watahiki et al. [54]	Makni et al. [55]	Wong et al. [56]
Year	2011	2010	2012	2009	2011	2010	2009
Model	Time WF-net	Timed automata	Modified BPMN and Fiacre	XTUS automata	Extended BPMN and timed automata	Timed Petri Net	CSP algebra
Model checker	UPPAAL	UPPAAL	TINA	UPPAAL	UPPAAL	TINA	FDR
Graphical model	●	●	●	◑	●	●	○
Deadlock detection	◑	●	●	●	●	○	○

represented by deadline constraints within inter-organizational workflows, where certain private processes remain unexposed for other users, can be verified using modified Petri Nets [55]. Another approach is base on building a semantic model of a business process, including the occurring delays, and its verification using CSP process algebra [56]. Table IV presents a summary of selected process validation techniques.

IV. Temporal logics for time patterns in process models

Based on the presented time patters [38], we analyzed selected temporal logics [57], [58] in order to assess their suitability to represent these patterns.

A. Linear Temporal Logic

Linear Temporal Logic (LTL) [59] is a temporal logic system for representing of time linearity. Its language is obtained from standard propositional language (with the Boolean constant $\top$) by adding temporal-modal operators such as: *always in a past* (H), *always in a future* (G), *eventually in the past* (P), *eventually in the future* (F), *next and until* ($\mathcal{U}$) and *since* ($\mathcal{S}$) – co-definable with "until".

B. Computation Tree Logic

LTL is traditionally interpreted in models based on the point-wise time-flow frames $\mathcal{F} = \langle T, < \rangle$, so in a poitwise semantics. If we also admit two additional quantifiers: A ('along all paths)' and E (read: 'there exists at least one path such that'), we obtain a new system – commonly called *Computation Tree Logic* (CTL) [59] as a branching-time logic.

The construction way of these systems demonstrates that these systems are rather suitable for describing situations, somehow, temporally 'open' in a past or in a future.

C. Halpern-Shoham logic

The logic of Halpern-Shoham (HS) – introduced in [60] forms a multi-modal system suitable to represent the 13 well-known Allen's relations between intervals [36], and constitutes a concurrent approach to temporal reasoning w. r. t. the Computational Tree Logic (CTL) or the Linear Temporal Logic LTL – the more traditional and pointwise approaches. More, precisely, HS forms a modal representation of temporal relations: **a**fter (or **m**eets, **l**ater), **b**egins (or start), **d**uring, **e**nd and **o**verlap and they are rendered in HS by corresponding modal operators: $\langle A \rangle$ for **a**fter, $\langle B \rangle$ for **b**egins, $\langle D \rangle$ for **d**uring, etc. The full syntax of HS-entities ϕ is defined by:

$$\phi := p|\neg\phi|\phi \wedge \phi|\langle X \rangle|\langle \bar{X} \rangle\,, \tag{1}$$

where p is a propositional variable and $\langle \bar{X} \rangle$ denotes a modal operator for the inverse relation. Such logic is more suitable for representing some temporally 'closed' events, actions and processes, their mutual temporal relations or time lags.

Table V presents the overview of these logics in terms of time pattern representation for process models.

V. Summary

Process models can specify various aspects of business processes, among them the temporal ones. In this paper, we present the existing solutions for describing time aspects of process models. Although there are many notations for modeling processes, the main focus is on the BPMN notation. We provide several examples of representing time patterns in BPMN as well as discuss temporal issues with temporal logics for such specifications. Thus, the original contribution of this paper is threefold – we present:

1) the overview of business process modeling notations focusing on time-related elements,
2) the process models in BPMN notation which fulfil the Allen's relations,
3) the assessment of temporal logics in terms of using them for representing time patterns in process models.

The research presented in this paper is a proposal for further studies related to time issues in BPMN process models. Our future work will focus on practical assessment of process models with the time related logic specification and the possibility of validation and verification of such models [61] or compliance checking [62], especially with the existing tools which uses temporal knowledge [63], [64]. One of the possible directions is related to integration of timed process models with timed rules models for complex reasoning on contex data [65], [66]. Important issue is also practical design of timed models of processes and rules [67]. As we focus on analysis of a single process model, additional research for analyzing process models of processes changing over time can be an important issue [68].

Table V
SUITABILITY OF REPRESENTING TIME PATTERNS [38] USING TEMPORAL LOGICS AND ALLEN'S ALGEBRA

Pattern Type/ Logic type	**LTL/CTL**	**Allen's algebra**	**Halpern-Shoham logic**
Patter 1 (Duration and Time Lags)	◑	●	●
Pattern 2 (Duration)	◑	●	●
Pattern 3 (Time Lags between arbitrary Events)	◑	◑	◑
Pattern 4 (Fixed Data Element)	●	❍	❍
Pattern 5 (Schedule Restricted Element)	❍	●	●
Pattern 6 (Time Based Restrictions)	◑	❍	❍
Pattern 7 (Validity Period)	◑	●	●
Pattern 8 (Variability)	◑	●	●
Pattern 9 (Cyclic Elements)	◑	❍	❍
Pattern 10 (Periodicity)	❍	❍	❍

REFERENCES

[1] M. Weske, *Business Process Management: Concepts, Languages, Architectures 2nd Edition*. Springer, 2012.

[2] A. Lindsay, D. Dawns, and K. Lunn, "Business processes – attempts to find a definition," *Information and Software Technology*, vol. 45, no. 15, pp. 1015–1019, Dec 2003, elsevier.

[3] F. Hunka and R. Belunek, "Transaction based business process modeling," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015, pp. 1397–1402.

[4] M. Mach-Król, "Perspectives of using temporal logics for knowledge management," in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2012.

[5] M. Mach-Król and K. Michalik, "Selected aspects of temporal knowledge engineering," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1091–1096.

[6] M. Owoc and K. Marciniak, "Knowledge management as foundation of smart university," in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*. IEEE, 2013, pp. 1267–1272.

[7] W. M. P. Aalst, H. T. Beer, and B. F. Dongen, "Process mining and verification of properties: An approach based on temporal logic," in *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005, Agia Napa, Cyprus, October 31 - November 4, 2005, Proceedings, Part I*, R. Meersman and Z. Tari, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 130–147.

[8] Y. Qiang, M. Valcke, P. De Maeyer, and N. Van de Weghe, "Representing time intervals in a two-dimensional space: an empirical study," *Journal of Visual Languages & Computing*, vol. 25, no. 4, pp. 466–480, 2014.

[9] W. M. P. van der Aalst, "Business process management: A comprehensive survey," *ISRN Software Engineering*, vol. 2013, 2013.

[10] A. W. Scheer, *Aris: Business Process Modeling*, 3rd ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2000.

[11] M. Rosemann and W. M. P. van der Aalst, "A configurable reference modelling language," *Information Systems*, vol. 32, no. 1, pp. 1–23, 2007.

[12] C. Menzel and R. J. Mayer, "The IDEF family of languages," in *Handbook on Architectures of Information Systems*, ser. International Handbooks on Information Systems, P. Bernus, K. Mertins, and G. Schmidt, Eds. Springer Berlin Heidelberg, 1998, ch. 10, pp. 209–241.

[13] C. Badiča, A. Badiča, and V. Litoiu, "A new formal IDEF-based modelling of business processes," in *Proc. of the 1st Balkan Conference in Informatics, Thessaloniki, Greece*, 2003, pp. 535–549.

[14] C. P. Menzel, R. J. Mayer, and D. D. Edwards, "IDEF3 formalization report," DTIC Document, Texas TX USA, Tech. Rep. KBSL-89-1007, October 1991.

[15] C. Badiča and C. Fox, "Hybrid DEF0/IDEF3 modelling of business processes: Syntax, semantics and expressiveness," in *Romanian-Austrian Workshop on Computer-Aided Verification of Information Systems: A Practical Industry-Oriented Approach. Timisoara, Romania*, 2004, pp. 20–22.

[16] R. J. Mayer, C. Menzel, M. Painter, P. S. deWitte, T. Blinn, and B. Perakath, "Information integration for concurrent engineering (IICE) IDEF3 process description capture method report," Knowledge Based Systems, Inc., Tech. Rep. AL-TR-1995-XXXX, 1995.

[17] OMG, "Unified Modeling Language (OMG UML) version 2.2. superstructure," Object Management Group, Tech. Rep. formal/2009-02-02, February 2009.

[18] J. Hunt, *Guide to the Unified Process featuring UML, Java and Design Patterns*. Springer, 2003.

[19] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd ed. Addison-Wesley Professional, 2003.

[20] D. Pilone and N. Pitman, *UML 2.0 in a Nutshell*. O'Reilly, 2005.

[21] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[22] M. Szpyrka, *Sieci Petriego w modelowaniu i analizie systemów współbieżnych*. Warszawa: Wydawnictwa Naukowo-Techniczne, 2008.

[23] B. Berthomieu and M. Diaz, "Modeling and verification of time dependent systems using time petri nets," *IEEE transactions on software engineering*, vol. 17, no. 3, p. 259, 1991.

[24] L. Popova-Zeugmann, *Time and Petri Nets*. Springer, 2013.

[25] A. H. M. ter Hofstede, W. M. P. van der Aalst, M. Adams, and N. Russell, Eds., *Modern Business Process Automation: YAWL and its Support Environment*. Springer, 2010.

[26] W. M. P. van der Aalst and A. H. M. ter Hofstede, "YAWL: Yet another workflow language," *Information Systems*, vol. 30, no. 4, pp. 245–275, 2005.

[27] W. M. P. van der Aalst and A. H. M. ter Hofstede, "Workflow patterns: On the expressive power of (petri-net-based) workflow languages," in *Proceedings of the Fourth International Workshop on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark, August 28-30, 2002*, K. Jensen, Ed., University of Aarhus. DAIMI PB-560, Aug 2002, pp. 1–20.

[28] W. M. P. van der Aalst, "Verification of workflow nets," in *Application and Theory of Petri Nets 1997*, ser. Lecture Notes in Computer Science, P. Azema and G. Balbo, Eds. Springer Berlin Heidelberg, 1997, vol. 1248, pp. 407–426.

[29] OMG, "Business Process Model and Notation (BPMN): Version 2.0 specification," Object Management Group, Tech. Rep. formal/2011-01-03, January 2011.

[30] B. List and B. Korherr, "An evaluation of conceptual business process modelling languages," in *Proceedings of the 2006 ACM symposium on Applied computing*. ACM, 2006, pp. 1532–1539.

[31] W. M. P. van der Aalst, L. Aldred, M. Dumas, and A. H. M. ter Hofstede, "Design and implementation of the YAWL system," in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science, A. Persson and J. Stirna, Eds. Springer Berlin Heidelberg, 2004, vol. 3084, pp. 142–159.

[32] J. Recker, M. Rosemann, M. Indulska, and P. F. Green, "Business process modeling – a comparative analysis," *Journal of the Association for Information Systems*, vol. 10, no. 4, 2009.

[33] F.-R. Lin, M.-C. Yang, and Y.-H. Pai, "A generic structure for business process modeling," *Business Process Management Journal*, vol. 8, no. 1, pp. 19–41, 2002.

[34] W. Wang, H. Ding, J. Dong, and C. Ren, "A comparison of business process modeling methods," in *Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics, 2006. SOLI '06*, 2006, pp. 1136–1141.

[35] O. Svatoš, "Conceptual process modeling language: Regulative approach," in *Proceedings of the 9th Undergraduate and Graduate Students eConf. and 14th Business & Government Executive Meeting on Innovative Cross-border eRegion, Univ. of Maribor*, 2007.

[36] J. Allen, "Maintaining knowledge about temporal intervals," in *Communications of ACM*, 26(11)1983, pp. 832–843.

[37] K. Kluza and K. Kaczor, "Overview of BPMN model equivalences: towards normalization of BPMN diagrams," in *8th Workshop on Knowledge Engineering and Software Engineering (KESE2012) at the at the biennial European Conference on Artificial Intelligence (ECAI 2012): August 28, 2012, Montpellier, France*, J. Canadas, G. J. Nalepa, and J. Baumeister, Eds., 2012, pp. 38–45. [Online]. Available: http://ceur-ws.org/Vol-949/

[38] A. Lanz, B. Weber, and M. Reichert, "Time patterns for process-aware information systems," *Requirements Engineering*, vol. 19, no. 2, pp. 113–141, 2012.

[39] D. Gagne and A. Trudel, "Time-bpmn," in *2009 IEEE Conference on Commerce and Enterprise Computing*, July 2009, pp. 361–367.

[40] S. Cheikhrouhou, S. Kallel, N. Guermouche, and M. Jmaiel, "A survey on time-aware business process modeling," in *International Conference on Enterprise Information Systems (ICEIS)*, July 2013, p. 10p.

[41] S. Cheikhrouhou, S. Kallel, N. Guermouche, and M. Jmaiel, "The temporal perspective in business process modeling: a survey and research challenges," *Service Oriented Computing and Applications*, vol. 9, no. 1, pp. 75–85, 2015.

[42] A. Lanz, B. Weber, and M. Reichert, "Workflow time patterns for process-aware information systems," in *Enterprise, Business-Process and Information Systems Modeling: 11th International Workshop, BPMDS 2010, and 15th International Conference, EMMSAD 2010, held at CAiSE 2010, Hammamet, Tunisia, June 7-8, 2010. Proceedings*. Berlin, Heidelberg: Springer, 2010, pp. 94–107.

[43] A. Lanz, U. Kreher, M. Reichert, and P. Dadam, "Enabling process support for advanced applications with the Aristaflow BPM suite," in *Proceedings of the Business Process Management 2010 Demonstration Track*, September 2010, no. 615.

[44] I. Barba, A. Lanz, B. Weber, M. Reichert, and C. del Valle, "Optimized time management for declarative workflows," in *13th BPMDS'12 working conference, Lecture Notes in Business Information Processing*. Berlin, Heidelberg: Springer, 2012, pp. 195–210.

[45] J. Xie, Y. Tang, Q. He, and N. Tang, "Research of temporal workflow process and resource modeling," in *Proceedings of the 9th international conference on computer supported cooperative work in design*, 2005, vol. 1, pp. 530–534.

[46] M. Szpyrka, G. J. Nalepa, A. Ligęza, and K. Kluza, "Proposal of formal verification of selected BPMN models with Alvis modeling language," in *Intelligent Distributed Computing V. Proceedings of the 5th International Symposium on Intelligent Distributed Computing – IDC 2011, Delft, the Netherlands – October 2011*, ser. Studies in Computational Intelligence, F. M. Brazier, K. Nieuwenhuis, G. Pavlin, M. Warnier, and C. Badica, Eds. Springer-Verlag, 2011, vol. 382, pp. 249–255. [Online]. Available: http://www.springerlink.com/content/m181144037q67271/

[47] A. Ligęza, K. Kluza, and T. Potempa, "Ai approach to formal analysis of bpmn models. towards a logical model for bpmn diagrams," in *Proceedings of the Federated Conference on Computer Science and Information Systems – FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012*, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., 2012, pp. 931–934. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6354394

[48] M. Szpyrka, P. Matyasik, and M. Wypych, "Alvis language with time dependence," in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*. IEEE, 2013, pp. 1565–1570.

[49] Y. Du, P. Xiong, Y. Fan, and X. Li, "Dynamic checking and solution to temporal violations in concurrent workflow processes," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1166–1181, 2011.

[50] R. Klimek and P. Szwed, "Verification of archimate process specifications based on deductive temporal reasoning," in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*. IEEE, 2013, pp. 1109–1116.

[51] N. Guermouche, "Etude des Interactions Temporisées dans la Composition de Services Web," PhD Thesis, Université Henri Poincaré - Nancy, Jun. 2010.

[52] N. Guermouche and S. D. Zilio, "Towards timed requirement verification for service choreographies," in *2012 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2012, pp. 117–126.

[53] S. Kallel, A. Charfi, T. Dinkelaker, M. Mezini, and M. Jmaiel, "Specifying and monitoring temporal properties in web services compositions," in *7th IEEE European Conference on Web Services, ECOWS '09*, 2009, pp. 148–157.

[54] K. Watahiki, F. Ishikawa, and K. Hiraishi, "Formal verification of business processes with temporal and resource constraints," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, 2011, pp. 1173–1180.

[55] M. Makni, S. Tata, M. Yeddes, and N. Ben Hadj-Alouane, "Satisfaction and coherence of deadline constraints in inter-organizational workflows," in *On the Move to Meaningful Internet Systems: OTM 2010: Confederated International Conferences: CoopIS, IS, DOA and ODBASE, Hersonissos, Crete, Greece, October 25-29, 2010, Proceedings, Part I*. Berlin, Heidelberg: Springer, 2010, pp. 523–539.

[56] P. Y. H. Wong and J. Gibbons, "A relative timed semantics for BPMN," *Electronic Notes in Theoretical Computer Science*, vol. 229, no. 2, pp. 59–75, 2009.

[57] K. Jobczyk and A. Ligeza, "Systems of temporal logic for a use of engineering. toward a more practical approach," in *Intelligent Systems for Computer Modelling*. Springer, 2016, pp. 147–157.

[58] K. Jobczyk and A. Ligeza, "Why systems of temporal logic are sometimes (un) useful?" in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2016, pp. 306–316.

[59] A. Pnueli, "The temporal logic of programs," *Proceedings of the 18th Annual Symposium on Foundation of Computer Science*, 1977:46-57.

[60] J. Halpern and Y. Shoham, "A propositional modal logic of time intervals," *Journal of the ACM*, vol. 38, pp. 935–962, 1991.

[61] M. Mach-Krol and K. Michalik, "Validation and verification of temporal knowledge as an important aspect of implementing a temporal knowledge base system supporting organizational creativity," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE, 2015, pp. 1315–1320.

[62] A. Awad, G. Decker, and M. Weske, "Efficient compliance checking using bpmn-q and temporal logic," in *Business Process Management: 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings*, M. Dumas, M. Reichert, and M.-C. Shan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 326–341.

[63] M. Mach-Król, "Tools for building a temporal knowledge base system supporting organizational creativity," *Procedia Computer Science*, vol. 65, pp. 1031–1037, 2015.

[64] E. Kucharska, K. Grobler-Dębska, J. Gracel, and M. Jagodziński, "Idea of impact of erp-aps-mes systems integration on the effectiveness of decision making process in manufacturing companies," in *International Conference: Beyond Databases, Architectures and Structures*. Springer, 2015, pp. 551–564.

[65] G. J. Nalepa and S. Bobek, "Rule-based solution for context-aware reasoning on mobile devices," *Computer Science and Information Systems*, vol. 11, no. 1, pp. 171–193, 2014.

[66] S. Bobek, M. Slazynski, and G. J. Nalepa, "Capturing dynamics of mobile context-aware systems with rules and statistical analysis of historical data," in *Artificial Intelligence and Soft Computing*, ser. Lecture Notes in Computer Science, L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, Eds., vol. 9120. Springer International Publishing, 2015, pp. 578–590. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-19369-4_51

[67] G. J. Nalepa and A. Ligęza, *Software engineering: evolution and emerging technologies*, ser. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, 2005, vol. 130, ch. Conceptual modelling and automated implementation of rule-based systems, pp. 330–340.

[68] D. Luengo and M. Sepúlveda, "Applying clustering in process mining to find different versions of a business process that changes over time," in *Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I*, F. Daniel, K. Barkaoui, and S. Dustdar, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 153–158.