Information Systems 57 (2016) 38–68

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

# Process time patterns: A formal foundation

Andreas Lanz [a,*], Manfred Reichert [a], Barbara Weber [b]

[a] Institute of Databases and Information Systems, Ulm University, Germany
[b] Quality Engineering Research Group, University of Innsbruck, Austria

ARTICLE INFO

*Article history:*
Received 24 July 2015
Received in revised form
24 August 2015
Accepted 6 October 2015
Recommended by: Dennis Shasha
Available online 13 November 2015

*Keywords:*
Process-aware Information System
Workflow Patterns
Process Time Patterns
Temporal Perspective
Temporal Constraints
Formal Semantics

ABSTRACT

Companies increasingly adopt process-aware information systems (PAISs) to model, execute, monitor, and evolve their business processes. Though the handling of temporal constraints (e.g., deadlines or time lags between activities) is crucial for the proper support of business processes, existing PAISs vary significantly regarding the support of the temporal perspective. Both the formal specification and the operational support of temporal constraints constitute fundamental challenges in this context. In previous work, we introduced *process time patterns*, which facilitate the comparison and evaluation of PAISs in respect to their support of the temporal perspective. Furthermore, we provided empirical evidence for these time patterns. To avoid ambiguities and to ease the use as well as the implementation of the time patterns, this paper formally defines their semantics. To additionally foster the use of the patterns for a wide range of process modeling languages and to enable pattern integration with existing PAISs, the proposed semantics are expressed independently of a particular process meta model. Altogether, the presented pattern formalization will be fundamental for introducing the temporal perspective in PAISs.



## 1. Introduction

Companies strive for comprehensive life cycle support of their business processes [1,2]. In particular, IT support for analyzing, modeling, executing, monitoring, and evolving business processes is becoming increasingly important [3,4]. In this context, process-aware information systems (PAISs) offer promising perspectives by enabling companies to define their business processes in terms of explicit *process schemas* as well as to create, execute and monitor related *process instances* in a controlled and efficient manner [1].

Both the formal specification and the operational support of *temporal constraints* constitute fundamental challenges for PAISs [5–8]. Although there exist many proposals for supporting the temporal process perspective, no comprehensive criteria for systematically assessing its support by a PAIS exist. To foster comparability and to facilitate the selection of PAIS-enabling technologies in a given application environment, *workflow patterns* have been introduced [9–12]. Respective patterns allow analyzing the expressiveness of process modeling languages and tools in respect to different process perspectives, including control flow [9], data flow [10], resources [11], activities [13], exceptions [14], and process changes [12,15,16]. Recently, we extended the workflow patterns by a set of 10 *process time patterns* (*time patterns* for short) suitable for evaluating the support of the temporal perspective in PAISs [17,18]. Examples of time patterns include *Time Lags between Activities*, *Durations*, and *Fixed*

* Corresponding author.
*E-mail addresses:* andreas.lanz@uni-ulm.de (A. Lanz), manfred.reichert@uni-ulm.de (M. Reichert), barbara.weber@uibk.ac.at (B. Weber).

http://dx.doi.org/10.1016/j.is.2015.10.002
0306-4379/

*Date Elements*. Empirical evidence, we gained in case studies [18], has confirmed that the proposed time patterns are common in practice and are required for properly modeling the temporal perspective of processes in many application domains [18]. Finally, we evaluated different approaches and tools in respect to their time pattern support [18].

### *1.1. Problem statement*

Our evaluation of approaches and tools incorporating the temporal process perspective has revealed the need for precise formal semantics of the time patterns. In particular, if such formal semantics are not present, patterns may be interpreted differently and ambiguities regarding informal pattern descriptions will result. In turn, this would hamper both pattern implementation and a pattern-based comparison of PAISs. Only when providing these precise formal semantics, it can be ensured that different implementations of a particular time pattern share the same semantics and, hence, have the same effects during process enactment. Precise formal semantics further constitute a prerequisite for verifying the temporal perspective of a business process at both build- and run-time [5–7,19,20], i.e., to check whether the temporal constraints of a process are satisfiable. Moreover, formal semantics are needed to be able to detect temporal inconsistencies in a process schema caused by interactions among different time pattern occurrences. Example 1 illustrates how such interactions may result in hidden effects. In particular, note that the occurrence of a time pattern within a process schema can never be treated in isolation. This significantly differentiates the time patterns from related patterns (e.g., workflow or data patterns [9,10]), making a formal specification of their semantics indispensable. Only then a robust and error-free process execution becomes possible. Precise formal semantics are further required to achieve a common understanding of process schemas using the time patterns.

**Example 1** (*Interactions between temporal constraints*). Fig. 1 depicts process schema $S_1$ consisting of three activities and two control gateways. Each activity is associated with a minimum and maximum duration (Pattern: *Duration*). Furthermore, time lags exist between the end of activity $A_1$ and the start of activity $A_3$, between the end of $A_1$ and the start of $A_4$, and between the end of $A_3$ and the end of $A_4$ (Pattern: *Time Lags between Activities*). At first glance, the process schema seems to be sound. However, when taking a closer look at it, one realizes that $S_1$ can never be executed without violating at least one of its temporal constraints. In particular, $A_3$ may be started the earliest 20 time units after completing $A_1$ and takes at least 30 time units to complete, i.e., it completes at least 50 time units after completing $A_1$. In turn, $A_4$ must start the latest 25 time units after completing $A_1$ and takes at most 10 time units to complete. Thus, $A_4$ completes at most 35 times units after completing $A_1$. However, this violates the time lag between $A_3$ and $A_4$. Particularly, it is not possible to complete $A_3$ within 10 time units after completing $A_4$.

In order to tackle the issues and limitations outlined above, formal semantics need to specify how the various time patterns interact with the elements of the control flow perspective, i.e., control flow patterns like loop, XOR-split, or AND-join. Moreover, in the context of loops and concurrent data access, for time patterns referring to process instance data (e.g., appointments made during run-time), it must be precisely defined which version of the data value shall be used for a specific pattern instance. For example, if a data object may be modified more than once during run-time, it must always be clear which version of the data value shall be used when evaluating a particular temporal constraint referring to the data object.

Since a pattern is defined as a reusable solution to a commonly occurring problem, the time patterns should be applicable to a wide range of application scenarios. A particular challenge is to provide a formal description of the time patterns, which is independent of a specific process modeling language or paradigm. Only then time patterns as well as their formal semantics will be widely accepted. Finally, this is required to enable PAIS engineers to integrate the time patterns without need to cope with language-specific issues.

### *1.2. Contribution*

This paper complements our previous work on time patterns [17,18] by providing *precise formal semantics* for them. These semantics are defined independent of a specific process modeling language or paradigm. Furthermore, we illustrate the pattern semantics through realistic examples and detailed explanations.

To define the pattern semantics independent of any process modeling language, while still closely intertwined with process execution semantics, we use *execution traces* as the basis for the formalization [21]. An execution trace may be considered as logical representation of the *execution history* of a process instance, i.e., it reflects what happened during the execution of a particular process instance. We formally describe time pattern semantics by characterizing which traces are producible on a process schema that contains a particular time pattern, i.e., we formally describe which traces are *temporally compliant* with the pattern semantics. This enables the implementation of techniques for checking the conformance of a process instance [22] with respect to a given process schema and its temporal constraints (i.e., occurrences of the time patterns). For pattern occurrences depending on run-time data of the process instance (i.e., on process instance data), we define which version of a data value shall be valid for a specific pattern instance. Finally, based on the presented formal semantics, we provide a reference implementation of selected time pattern variants.

As will be shown, the provided formal semantics contribute to overcome the problems discussed in Section 1.1. In particular, they will foster the integration of the temporal perspective into PAISs broadening their application scope significantly.

Section 2 provides background information and summarizes the time patterns. Section 3 discusses the research method we applied for defining and evaluating the proposed