



Clinical Processes - The Killer Application for Constraint-Based Process Interactions

Jiménez-Ramírez, Andrés; Barba, Irene; Reichert, Manfred; Weber, Barbara; Del Valle, Carmelo

Published in:
Advanced Information Systems Engineering

Link to article, DOI:
[10.1007/978-3-319-91563-0_23](https://doi.org/10.1007/978-3-319-91563-0_23)

Publication date:
2018

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Jiménez-Ramírez, A., Barba, I., Reichert, M., Weber, B., & Del Valle, C. (2018). Clinical Processes - The Killer Application for Constraint-Based Process Interactions. In *Advanced Information Systems Engineering* (pp. 374-390). Springer. Lecture Notes in Computer Science, Vol.. 10816 https://doi.org/10.1007/978-3-319-91563-0_23

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Clinical Processes - The Killer Application for Constraint-Based Process Interactions?

Andres Jimenez-Ramirez¹, Irene Barba¹, Manfred Reichert²,
Barbara Weber^{3,4}, and Carmelo Del Valle¹

¹ Departamento de Lenguajes y Sistemas Informáticos, University of Seville, Spain
{ajramriez, irenebr, carmelo}@us.es

² Institute of Databases and Information Systems, Ulm University, Germany
Manfred.Reichert@uni-ulm.de

³ Department of Computer Science, University of Innsbruck, Austria
Barbara.Weber@uibk.ac.at

⁴ Technical University of Denmark, Denmark
bweb@dtu.dk

Abstract. For more than a decade, the interest in aligning information systems in a process-oriented way has been increasing. To enable operational support for business processes, the latter are usually specified in an imperative way. The resulting process models, however, tend to be too rigid to meet the flexibility demands of the actors involved. Declarative process modeling languages, in turn, provide a promising alternative in scenarios in which a high level of flexibility is demanded. In the scientific literature, declarative languages have been used for modeling rather simple processes or synthetic examples. However, to the best of our knowledge, they have not been used to model complex, real-world scenarios that comprise constraints going beyond control-flow. In this paper, we propose the use of a declarative language for modeling a sophisticated healthcare process scenario from the real world. The scenario is subject to complex temporal constraints and entails the need for coordinating the constraint-based interactions among the processes related to a patient treatment process. As demonstrated in this work, the selected real process scenario can be suitably modeled through a declarative approach.

Keywords: Process flexibility, declarative process model, healthcare process, temporal constraints

1 Introduction

For several years, there has been an increasing interest in aligning information systems in a process-oriented way [6, 30]. Thereby, a business process (BP) consists of a set of activities which jointly realize a business goal and whose execution needs to be coordinated in an organizational as well as technical environment [30]. In this context, process-aware information systems (PAISs) offer promising perspectives by enabling enterprises to define their business processes in terms of

explicit process models as well as to execute the corresponding process instances in a controlled and efficient manner [24, 30].

Declarative approaches are becoming increasingly popular for modeling business processes as they are able to cope with some of the limitations imperative notations are facing [28, 31, 20, 9, 17, 10, 4]. Although declarative modeling languages have been extensively discussed in literature [19, 5], even in the context of healthcare [29, 21, 25], to the best of our knowledge, they have not been used to model complex, real-world scenarios that comprise constraints going beyond control-flow.

In this paper, we propose the use of a declarative language for modeling a sophisticated healthcare process scenario from the real world. Particularly, the latter is subject to complex temporal constraints and entails the need for coordinating the constraint-based interactions among all the processes contributing to the overall patient treatment process. We strongly believe that, if we are able to demonstrate the applicability of declarative modeling languages to clinical process, which can be considered as a kind of killer application for PAISs, the respective approach can be applied to many other sophisticated process scenarios as well.

Although the temporal perspective is present in many real-world process scenarios, it has not received sufficient attention yet. In previous work [14], we systematically derived 10 process time patterns (i.e., solutions for representing commonly occurring temporal constraints in PAISs) by analyzing a large collection of non-trivial process models from various domains. In particular, the time patterns (TPs for short) were defined independently of a specific language or paradigm for BP modeling [13]. Despite the needs for enabling process flexibility and dealing with temporal constraints, most existing approaches are unable to manage both. To fill this gap, we proposed the TConDec-R language [2], a declarative process modeling language that enables the specification of temporal constraints related to the TPs. As demonstrated in this work, TConDec-R is suitable for modeling a sophisticated and real-world process scenario from the healthcare domain.

The remainder of the paper is organized as follows. Section 2 presents backgrounds on temporal constraints in declarative process modeling languages and shows why TConDec-R was selected for modeling the considered scenario. Section 3 motivates the need for coordinating the constraint-based interactions among all the processes in a hospital forming the overall patient treatment process. Section 4 details how TConDec-R is used for modeling such a scenario. Section 5 discusses our work. Finally, Section 6 concludes the paper with a summary and an outlook.

2 Background

This section discusses work related to the considered time patterns (cf. Sect. 2.1). It then presents a review of proposals that support temporal constraints in contemporary declarative process modeling languages (cf. Sect. 2.2). Finally,

Table 1. Selected process time patterns.

Cat.	Time Pattern (TP)	Example
I	<i>TP1 (Time Lags between two Activities)</i> enables the definition of different kinds of time lags between two activities.	<i>The time lag between registering a Master thesis and submitting it must not exceed 6 months.</i>
	<i>TP2 (Durations)</i> allows specifying the duration of process activities.	<i>Processing 100 requests must not take longer than 1 second.</i>
	<i>TP5 (Schedule Restricted Element)</i> allows restricting the enactment of a particular activity by a schedule.	<i>Comprehensive lab tests in a hospital can only be done from MO-FR between 8 am and 5 pm.</i>
II	<i>TP6 (Time-based Restrictions)</i> provides support for restricting the number of times a specific process element may be executed within a given timeframe.	<i>For a specific lab test at least 5 different blood samples have to be taken within 24 hrs.</i>

it provides an overview of TConDec-R, a declarative process modeling language with extensive support for temporal constraints (cf. Sect. 2.3).

2.1 Process Time Patterns

In [14, 13], we systematically identified 10 process time patterns (TPs) by analyzing a large collection of process models from various domains. In this context, 4 of the 10 TPs are considered in the present work as they refer to aspects directly related to the considered scenario (cf. Table 4, column *TP*). Specifically, constraints C8, C17, C18, C19, C21, C22, and C23 correspond to TP1; C11, C12 and C20 correspond to TP5; finally, C3, C14, and C5 correspond to TP6. In addition, the duration of the activities (cf. Table 3) corresponds to TP2. We divided the time patterns into two categories according to pattern semantics (cf. Table 1). Category I (*Durations and Time Lags*) provides support for expressing the durations of different process granularities (i.e., activities, activity sets, processes, or sets of process instances) as well as time lags between activities or process events (e.g., milestones). Category II (*Restricting Execution Times*), in turn, allows constraining execution times of single activities or entire processes (e.g., deadlines). All considered time patterns are highly relevant for the support of patient treatment processes.

Note that there exist numerous variants of the time patterns, also denoted as pattern variants [14]. To cope with this variability and to keep the number of patterns manageable, design choices allow for TP parametrization [14]. For example, whether a time lag represents a minimum value, maximum value, or both (i.e., an interval) constitutes a design choice of pattern TP1 (Time Lags between Activities). Additional variance of this time pattern is caused by the fact that time lags may be specified either between the start of two activities, the start of the first and the end of the second activity, the end of the first and the start of the second activity, or the end of the two activities. Other design choices, in turn, cover the fact that different time granularities (e.g., second or hour) or process granularities (i.e., activity, activity set, process, or set of process

instances) may be applied in the context of the time patterns. A complete list of pattern design choices and, hence, pattern variants, is presented in [14].

2.2 Temporal Constraints in Declarative Processes Modeling Languages

This section reviews the works dealing with declarative process modeling languages that support the temporal constraints necessary for modeling the considered scenario.

There exist several proposals for handling the temporal constraints in different domains. We exclude works that are not related to business processes, view temporal constraints solely in the sense of ordering relations, or only mention temporal constraints at an abstract level (i.e., no specific temporal constraints are discussed). In the end, several papers [20, 9, 16, 17, 15, 32, 10, 4, 26, 2, 18] were identified as relevant works in the context of our research.

In order to assess the support each proposal provides for the time patterns, we analyzed the following temporal constraint features:

- F1 Does the proposal allow specifying time lags between activities? (cf. TP1)
- F2 Does the proposal allow specifying the duration of process activities? (cf. TP2)
- F3 Does the proposal consider constraints that may refer to a calendar or schedule? (cf. TP5)
- F4 Does the proposal allow for time-based constraints, restricting the number of times a particular process element may be executed within a pre-specified timeframe? (cf. TP6)

The respective declarative approach allows specifying the temporal constraints related to the considered time patterns (cf. Table 1), if these four features can be checked. Note that we check whether the proposed modeling language *directly* supports the elements needed for assessing the feature. Hence, we do not check for workarounds of the respective approach possible in this context.

Table 2 includes the considered works together with the answers (i.e., yes (✓) or no (X)) to each feature. It indicates that (1) patterns TP1 and TP2 are well supported, (2) pattern TP4 is only supported by [18, 2], and (3) TConDec-R [2] is the only approach supporting TP6. Therefore, to the best of our knowledge, only TConDec-R supports the modeling of business processes enhanced with the temporal constraints related to the time patterns (cf. Table 1). Moreover, for every TConDec-R temporal constraint all relations stated in Allen’s interval algebra [1] (i.e., start-start, start-end, end-start, and end-end) can be specified.

2.3 The TConDec-R Language

As basis of the TConDec-R language [2], we use Declare [23] for specifying activities and their behavioral (i.e., control-flow) constraints. We consider this

Table 2. Consolidated results of the review.

<i>Work</i>	<i>Features</i>			
	<i>F1</i>	<i>F2</i>	<i>F3</i>	<i>F4</i>
Montali et al. [20]	✓	✓	X	X
Hildebrandt et al. [9]	✓	X	X	X
Maggi et al. [16, 17, 15]	✓	X	X	X
Jiang et al. [10]	✓	✓	X	X
Burattin et al. [4, 26]	✓	X	X	X
Mans et al. [18]	X	✓	✓	X
Barba et al. [2]	✓	✓	✓	✓

✓ = true, X = false

declarative modeling language as appropriate as it enables the specification of a wide range of process models in a flexible way. Respective process models are denoted as *constraint-based*, i.e., they comprise information about (1) the activities that may be performed during process enactment as well as (2) the constraints to be obeyed in this context. The activities of a constraint-based process model may be executed arbitrarily often unless this is restricted by any constraint. Declare constraints can be categorized as follows [23]:

1. **Existence constraints** are unary relationships constraining the number of times a particular activity may be executed in the context of a process instance. For example, *Exactly(A,n)* specifies that activity *A* must be executed exactly *n* times for each process instance.
2. **Relation constraints** are positive binary relationships used to either enforce or allow for the enactment of activities in certain situations. For example, *Precedence(A,B)* specifies that activity *B* may only be executed if *A* is executed before.
3. **Negation constraints** are negative binary relationships used to forbid the enactment of certain activities in specific situations. For example, *NotCoexistence(A,B)* expresses mutual exclusion of *A* and *B*, i.e., *A* must not be executed if *B* is executed, and vice versa.

Definition 1. (*TConDec-R activity*). A *TConDec-R activity* $act = (a, dur)$ refers to a process activity *a* with its estimated duration *dur*.⁵

Definition 2. (*TConDec-R process model*). A *TConDec-R process model* $TCR = (Acts, C_T)$ corresponds to an extended constraint-based process model, where

- *Acts* is a set of *TConDec-R activities* and
- *C_T* is a set of constraints that may include any control-flow constraint supported by *Declare* as well as any temporal constraint related to the time patterns (cf. Table 1).

⁵ Respective estimates can be obtained by interviewing subject matter experts or by analyzing the event logs of completed process instances. Moreover, both approaches can be combined to obtain more reliable estimates.

TConDec-R process model

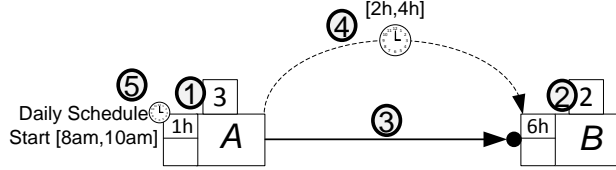


Fig. 1. A simple TConDec-R process model.

TConDec-R constraints are specified according to the graphical notation proposed for Declare constraints [23] and using the graphical notation proposed in [13] for visualizing the temporal constraints.⁶

Example 1. (TConDec-R process model). Figure 1 shows a simple example of a TConDec-R process model:

- $Acts = \{(A, 1h), (B, 6h)\}$ consists of two activities A and B ; A has an estimated duration of 1 hour. In turn, B has an estimated duration of 6 hours;
- C_T comprises the following constraints:
 - (1) $Exactly(A, 3)$, expressing that A shall be executed exactly three times,
 - (2) $Exactly(B, 2)$, expressing that B shall be executed exactly twice,
 - (3) $Precedence(A, B)$, expressing that activity B may only be executed if A is executed before,
 - (4) $TimeLagEndStart(A, B, 2h, 4h)$, representing a temporal constraint corresponding to time pattern TP1; it expresses that for each execution A_i of A , there must be at least one execution B_j of B such that there is a time lag of at least 2 hours ($2h$) and at most $4h$ between the end time of A_i and the start time of B_j , and
 - (5) $DailyScheduleStart(A, 8am, 10am)$, expressing a temporal constraint related to TP5, i.e., each execution of A must start between 8 and 10am.

In order to provide support to the TConDec-R language, a web-based tool has been implemented based on previous approaches [11, 2].⁷ The tool consists of a light-weight web interface that allows for (1) textually modeling declarative business processes using the TConDec-R language or loading already created models, and (2) providing operational support for the models (e.g., generating valid execution traces or checking the conformance of models). For example, Fig. 2 shows the tool where a TConDec-R model has been loaded and a valid trace is generated. The client uses a REST API layer to connect to a server being in charge of all the heavy-duty tasks requested by the user interface.

3 Clinical Process Scenarios

3.1 General Insights into Clinical Practice

In the following, an impression of the constraints driving the interactions between the clinical workflows of a patient treatment process is given to emphasize

⁶ A complete formalization of the TConDec-R constraints is available at <http://azarias.lsi.us.es/TCR/Formalization.pdf>

⁷ It is available at <http://azarias.lsi.us.es/TCR/ModelChecker>

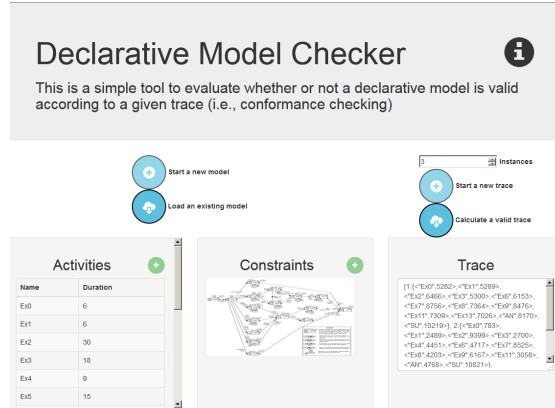


Fig. 2. TConDec-R client interface.

under which conditions constraint-based solutions need to operate in a healthcare environment.

In the context of a patient treatment process, numerous clinical procedures have to be planned, ordered and prepared, appointments be made, and results be obtained and evaluated. Moreover, many procedures need preparatory measures of various complexity. Before a surgery may take place, a patient has to undergo numerous preliminary examinations (i.e., smaller processes), each of them requiring additional preparations. While some of them are known in advance, others may have to be scheduled dynamically, depending on the individual patient and her state of health.

In general, the various clinical workflows to be performed in a patient treatment process as well as their tasks may have to consider complex temporal constraints. After an injection with contrast medium was given to a patient, for example, some other tests cannot be performed within a certain period of time. In contemporary healthcare environments, physicians still have to coordinate the tasks related to their patients manually, taking into account all the constraints existing in this context. Therefore, changing a schedule is not trivial and requires time-consuming communication.

For other procedures, medical staff from various departments have to collaborate. Thus, the process is subdivided into organization-oriented views leading to several problems. First, patients have to wait, because resources (e.g., physicians, rooms or technical equipment) are not available due to insufficient coordination. Second, medical procedures cannot be performed as planned if information is missing, preparations are omitted, or a preceding procedure is postponed, canceled or requires latency time. Depending procedures might then have to be re-scheduled resulting in time-consuming phone calls. Third, if urgently needed results are missing, clinical procedures may have to be performed repeatedly causing unnecessary costs and burdening patients.

For all these reasons, from both the patient and the hospital perspective undesired effects occur: hospital stays can take longer than required and costs or even invasiveness of patient treatment increase. Therefore, healthcare process

Table 3. Processes relevant in the context of the considered clinical scenario.

<i>ID</i>	<i>Description</i>	<i>Dur</i>	<i>Unit/Dep</i>	<i>%Req</i>
Ex0	First visit and examination of the patient	30m	WH	100
Ex1	Pelvic Ultra-sound Imaging	30m	UU	100
Ex2	Cystoscopy & Rectoscopy	2h30m	ED	100
Ex3	Uretero Pyelography	1h30m	RD	100
Ex4	CT scanning	45m	RD	60
Ex5	Magnetic Resonance Imaging	1h15m	RD	40
Ex6	Colonoscopy	2h15m	CCC	100
Ex7	Colon Contrast Imaging	3h30m	CCC	40
Ex8	X-ray of the gastrointestinal tract	1h15m	RD	35
Ex9	Chest X-ray	30m	RD	85
Ex10	Blood test	10m	WH	70
Ex11	Laparoscopy	1h	LU	100
Ex12	Doppler examinations	30m	UU	20
Ex13	Medical council with the Otolaryngology Department	60m	OD	20
Ex14	Medical council with the Neurology Department	30m	ND	10
AN	The patient is examined and interviewed by an anesthetist	1h	WH	100
SU	The surgery is performed	2-6h	WH	75

h = hours, m = minutes

support, in particular regarding the many constraint-based interactions between processes, would be highly welcome by medical staff.

3.2 A Concrete Treatment Scenario

This section deals with the considered clinical scenario, which we derived by interviewing subject matter experts as well as by analyzing process-relevant information (e.g., patient records, process handbooks) [27, 22].

This clinical scenario deals with surgeries (including their preparations) in the context of ovarian carcinoma [27, 22]. Usually, patients with ovarian carcinoma are treated in the women’s hospital (WH); on average 2 patients with this diagnosis emerge to the WH per day. After admitting a patient to the gynecological ward, one of the two ward physicians visits and examines the patient. Afterwards, the physician orders and schedules a number of medical examinations, which need to be performed before the surgery may take place. Additionally, the patient needs to be examined by an anesthetist who may then request additional medical examinations before the surgery.

Some of the examinations are not carried out in the WH itself, but are provided by other clinical departments, which may be either internal or external (i.e., placed at a different location) to the WH. In the given scenario, five external departments (i.e., Endoscopy Department (ED), Radiology Department (RD), Comprehensive Cancer Centre (CCC), Otolaryngology Department (OD), and Neurology Department (ND)) are involved as well as an internal one comprising two units (i.e., Ultrasound Unit (UU) and Laparoscopy Unit (LU)). Each department has limited resources that provide services to many other departments and clinics respectively (including WH). In order to ensure a fair use of the shared resources, a particular department may only order a maximum number of a specific examination from the respective provider per day.

All relevant processes and activities of the considered clinical scenario (i.e., appointments to be made with the physician and anesthetist, medical exami-

nations to be performed, and the surgery to be carried out) are summarized in Table 3: Column *ID* contains the identifier of the activity, *Dur* its average duration, and *Unit/Dep* the unit/department the activity is performed at. Finally, *%Req* indicates the frequency with which the respective examination is requested for a patient (e.g., *%Req* = 100 expresses that the examination is requested once for every patient, i.e., in 100% of all cases). Estimates regarding the average duration of the activities were obtained by interviewing subject matter experts. Considering such indications, an example of the different processes involving 14 patients for the considered scenario can be generated using the provided tool (cf. Sect. 2.3).⁸

Concerning the surgery of a particular patient and required preparations (i.e., medical examinations), several constraints (including temporal ones) need to be obeyed. Examples include chronological orders of activities and time lags between them. Corresponding constraints are informally summarized in Table 4.

4 Modeling the Clinical Scenario with TConDec-R

This section shows how the considered scenario (cf. Sect. 3.2) can be properly described with TConDec-R. Note that all constraints related to the clinical scenario can be modeled using TConDec-R constraints. The resulting TConDec-R constraints are depicted in the right column of Table 4. When capturing the scenario with TConDec-R, we obtain the model depicted in Fig. 3.

On one hand, each process activity is characterized by its estimated duration according to Def. 2. In Fig. 3, the department/hospital performing the activities is depicted for the sake of clarity (e.g., *Ex0* is performed in the *WH*). Note that this information is useful, for example, to check whether patient transportation is needed between two directly succeeding activities. In such cases, a time lag of 20 minutes is added to properly cover the time needed for transportation.

On the other, each constraint is labeled with the related constraint ID (cf. Table 4). In addition, for the sake of clarity, in Fig. 3 some related constraints are depicted together. For example, constraints *Precedence(A,B)* and *TimeLagEndStart(A,B,LB,UB)* are depicted as a *Precedence* constraint with a clock and the related time lag [LB,UB] on top of it (e.g., constraint *C8*, *Precedence(Ex6,Ex7)*, and *TimeLagEndStart(Ex6,Ex7,6d,-)*). For the sake of readability, when the same constraint applies to a set *S* of activities in Table 4, it is depicted as only one constraint over *S*, e.g., in constraint *C1*, constraints *Exactly(Ex0,1)*, *Exactly(Ex1,1)*, *Exactly(Ex2,1)*, *Exactly(Ex3,1)*, *Exactly(Ex6,1)*, and *Exactly(AN,1)* are represented by *Exactly({Ex0,Ex1,Ex2,Ex3,Ex6,AN},1)*. Further, in the TConDec-R model from Fig. 3, the *TimeBasedExclusive1of2Daily* constraints (e.g., constraint *C3*) are not depicted as each of them relates one activity with (almost) all other activities and, hence, their inclusion would affect readability of the figure.

⁸ A graphical example is depicted in <http://azarias.lsi.us.es/TCR/PlanEx1.pdf>.

Table 4. Constraints related to the clinical scenario.

ID	Constraints Informally Described	TP	Related TComDec-R Constraints
C1	Activities Ex0, Ex1, Ex2, Ex3, Ex6, and AN are always performed once for all patients.	-	<i>Exactly</i> ({ <i>Ex0</i> , <i>Ex1</i> , <i>Ex2</i> , <i>Ex3</i> , <i>Ex6</i> , <i>AN</i> }, 1)
C2	Ex0 is always the first and Ex1 the second activity to be performed.	-	<i>Succession</i> (<i>Ex0</i> , <i>Ex1</i>) and <i>Precedence</i> (<i>Ex1</i> , { <i>Ex2</i> , <i>Ex3</i> , <i>Ex8</i> , <i>Ex10</i> , <i>Ex11</i> , <i>Ex13</i> , <i>Ex14</i> })
C3	No other examination may be scheduled for the day at which Ex2 is performed.	TP6	<i>TimeBasedExclusive1of2Daily</i> (<i>Ex2</i> , { <i>Ex0</i> , <i>Ex1</i> , <i>Ex3</i> , <i>Ex4</i> , ..., <i>Ex14</i> }, 1d)
C4	Ex3 has to be performed in the morning; i.e., it takes place between 8am and 12am.	TP5	<i>DailyScheduleStart</i> (<i>Ex3</i> , 8h, 12h) and <i>DailyScheduleEnd</i> (<i>Ex3</i> , 8h, 12h)
C5	Either Ex4 or Ex5 shall be performed for a particular patient, but not both examinations.	-	<i>Exclusive1of2</i> (<i>Ex4</i> , <i>Ex5</i>)
C6	Ex3 needs to be completed before Ex4 or Ex5 may be started.	-	<i>Precedence</i> (<i>Ex3</i> , <i>Ex4</i>) and <i>Precedence</i> (<i>Ex3</i> , <i>Ex5</i>)
C7	Ex4 or Ex5 needs to be finished before starting Ex6.	-	<i>Response</i> (<i>Ex4</i> , <i>Ex6</i>) and <i>Response</i> (<i>Ex5</i> , <i>Ex6</i>)
C8	If Ex7 is ordered, it must be performed after Ex6 and there must be a time lag between the completion of Ex6 and the start of Ex7 of at least 6 days (i.e., minimum time lag).	TP1	<i>Precedence</i> (<i>Ex6</i> , <i>Ex7</i>) and <i>TimeLagEndStart</i> (<i>Ex6</i> , <i>Ex7</i> , 6d, -)
C9	The day before Ex6 is performed, no other examination may be scheduled after 12am. The day Ex6 is performed, no other examination may be scheduled before Ex6 is started.	TP6	<i>TimeBasedNotPrecedenceDaily</i> ({ <i>Ex0</i> , <i>Ex1</i> , <i>Ex2</i> , ..., <i>Ex14</i> }, <i>Ex6</i> , 1d, -)
C10	The day Ex6 is performed, no other examination, except Ex10, may be scheduled.	TP6	<i>TimeBasedExclusive1of2</i> (<i>Ex6</i> , { <i>Ex0</i> , <i>Ex1</i> , <i>Ex3</i> , <i>Ex4</i> , ..., <i>Ex9</i> , <i>Ex11</i> , ..., <i>Ex14</i> }, 1d)
C11	Ex6 shall take place in the morning, i.e., between 8am and 12am.	TP5	<i>DailyScheduleStart</i> (<i>Ex6</i> , 8h, 12h) and <i>DailyScheduleEnd</i> (<i>Ex6</i> , 8h, 12h)
C12	Ex10 is performed early in the morning, i.e., between 6am and 7am.	TP5	<i>DailyScheduleStart</i> (<i>Ex10</i> , 6h, 7h) and <i>DailyScheduleEnd</i> (<i>Ex10</i> , 6h, 7h)
C13	Ex11, Ex13 and Ex14 are performed before the appointment with the anesthetist (AN) takes place.	-	<i>Response</i> (<i>Ex11</i> , <i>AN</i>), <i>Response</i> (<i>Ex13</i> , <i>AN</i>), and <i>Response</i> (<i>Ex14</i> , <i>AN</i>)
C14	Ex13 must not be performed more than once per day.	TP6	<i>TimeBasedCrossInstanceAbsence</i> (2, <i>Ex13</i> , 1d)
C15	Ex14 must not be performed more than once per day.	TP6	<i>TimeBasedCrossInstanceAbsence</i> (2, <i>Ex14</i> , 1d)
C16	Ex9 and Ex12 are performed after the appointment with the anesthetist (AN) took place.	-	<i>Precedence</i> (<i>AN</i> , <i>Ex9</i>) and <i>Precedence</i> (<i>AN</i> , <i>Ex12</i>)
C17	AN is going to be performed at least 1 day after completing Ex8.	TP1	<i>Response</i> (<i>Ex8</i> , <i>AN</i>) and <i>TimeLagEndStart</i> (<i>Ex8</i> , <i>AN</i> , 1d, -)
C18	All examinations need to be completed at least 3 days before the surgery may take place.	TP1	<i>Response</i> (<i>Ex2</i> , <i>SU</i>), <i>TimeLagEndStart</i> (<i>Ex2</i> , <i>SU</i> , 3d, -), <i>Response</i> (<i>Ex7</i> , <i>SU</i>), <i>TimeLagEndStart</i> (<i>Ex7</i> , <i>SU</i> , 3d, -), <i>Response</i> (<i>Ex9</i> , <i>SU</i>), <i>TimeLagEndStart</i> (<i>Ex9</i> , <i>SU</i> , 3d, -), <i>Response</i> (<i>Ex10</i> , <i>SU</i>), <i>TimeLagEndStart</i> (<i>Ex10</i> , <i>SU</i> , 3d, -), <i>Response</i> (<i>Ex12</i> , <i>SU</i>), and <i>TimeLagEndStart</i> (<i>Ex12</i> , <i>SU</i> , 3d, -)
C19	No other examination must take place at the day Ex13 or Ex14 is performed.	TP1	<i>TimeBasedExclusive1of2Daily</i> (<i>Ex13</i> , { <i>Ex0</i> , <i>Ex1</i> , <i>Ex3</i> , <i>Ex4</i> , ..., <i>Ex14</i> }, 1d) and <i>TimeBasedExclusive1of2Daily</i> (<i>Ex14</i> , { <i>Ex0</i> , <i>Ex1</i> , <i>Ex3</i> , <i>Ex4</i> , ..., <i>Ex13</i> }, 1d)
C20	SU needs to be started in the morning (between 8am and 12am).	TP5	<i>Schedule</i> (<i>SU</i> , { <i>W</i> , <i>T</i> }), <i>DailyScheduleStart</i> (<i>SU</i> , 8h, 12h), and <i>DailyScheduleEnd</i> (<i>SU</i> , 8h, 12h)
C21	Each department allows each other department to perform a maximum of 2 examinations per day.	TP1	<i>TimeBasedCrossInstanceAbsence</i> (3, { <i>Ex2</i> , <i>Ex3</i> , <i>Ex4</i> , <i>Ex5</i> , <i>Ex6</i> , <i>Ex7</i> , <i>Ex8</i> , <i>Ex9</i> }, 1d)
C22	Examinations may only be performed between 8am and 6pm (except Ex10, see C12).	TP1	<i>DailyScheduleStart</i> ({ <i>Ex0</i> , <i>Ex1</i> , <i>Ex2</i> , <i>Ex4</i> , <i>Ex5</i> , <i>Ex7</i> , <i>Ex8</i> , <i>Ex9</i> , <i>Ex11</i> , <i>Ex12</i> , <i>Ex13</i> , <i>Ex14</i> , <i>AN</i> }, 8h, 18h) and <i>DailyScheduleEnd</i> ({ <i>Ex0</i> , <i>Ex1</i> , <i>Ex2</i> , <i>Ex4</i> , <i>Ex5</i> , <i>Ex7</i> , <i>Ex8</i> , <i>Ex9</i> , <i>Ex11</i> , <i>Ex12</i> , <i>Ex13</i> , <i>Ex14</i> , <i>AN</i> }, 8h, 18h)
C23	No activity must be performed on Saturday or Sunday.	TP1	<i>Schedule</i> ({ <i>Ex0</i> , <i>Ex1</i> , <i>Ex2</i> , <i>Ex3</i> , <i>Ex4</i> , <i>Ex5</i> , <i>Ex6</i> , <i>Ex7</i> , <i>Ex8</i> , <i>Ex9</i> , <i>Ex10</i> , <i>Ex11</i> , <i>Ex12</i> , <i>Ex13</i> , <i>Ex14</i> , <i>AN</i> }, { <i>M</i> , <i>Tu</i> , <i>W</i> , <i>T</i> , <i>F</i> })

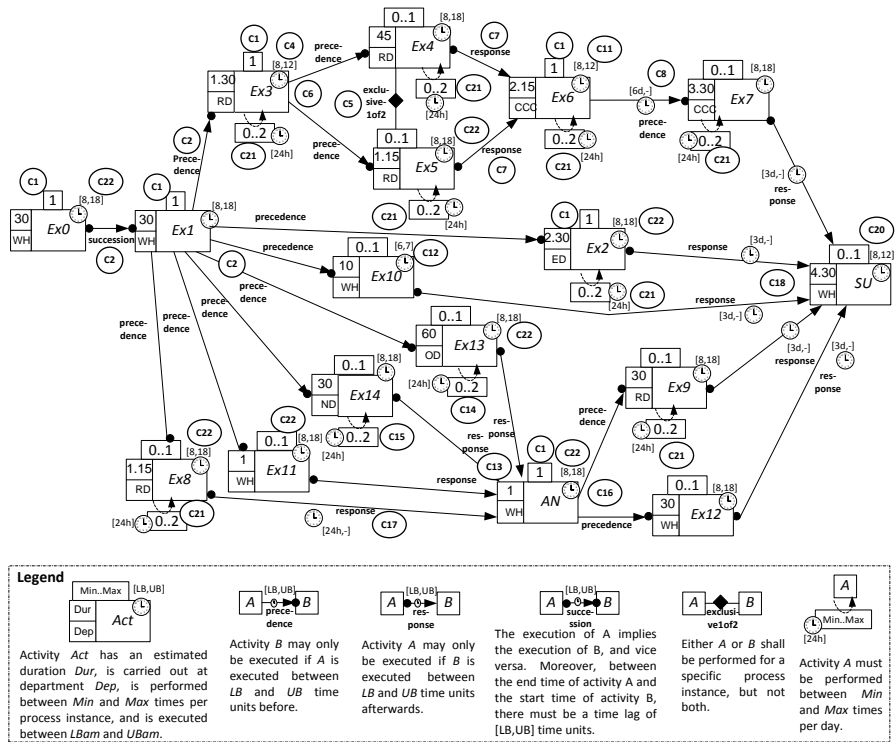


Fig. 3. Simplified TConDec-R model for the scenario.

Finally, constraints requiring that no activity may be performed on Saturday or Sunday (i.e., constraint *C23*) are added by stating corresponding schedule relations over all activities.

We consider the case as a proper clinical scenario for illustrating the proposed approach. It represents a real-world process, includes process-relevant perspectives (i.e., control flow and time), and requires dealing with many activities and numerous constraints of diverse nature. Altogether, the modeling of such scenario entails a rather high complexity, hence it may be considered as killer scenario for constraint-based languages. With the proposed approach, unlike with previous proposals, it becomes possible to model the scenario without the need for any workarounds or extensions.

The considered scenario has been modeled using the proposed web-based tool to generate valid execution traces. Such traces have then be used as a first validation of the model by checking that each constraint of the scenario (cf. Table 4) is fulfilled.⁹

⁹ A set of traces generated by the web-based tool can be accessed at <http://azarias.lsi.us.es/TCR/instances.zip>

5 Discussion and Lessons Learned

In the current work, we build upon a constraint-based business process modeling language that allows specifying sophisticated temporal constraints, i.e., the TConDec-R language[2]. On one hand, the fact that TConDec-R is framed in the declarative modeling paradigm allows, unlike imperative proposals, for the specification of flexible scenarios. On the other, there are many scenarios from various domains (e.g., healthcare, logistics, and engineering) requiring support for temporal constraints, particularly in the context of long-running processes. More precisely, this work deals with a sophisticated healthcare process scenario from the real world. We derived the scenario by interviewing subject matter experts as well as by analyzing process-relevant information in this context (e.g., patient records, process handbooks) [27, 22]. We consider the clinical processes as a proper case for illustrating the proposed approach as they were derived from a real hospital environment, include all process-relevant perspectives, and require dealing with numerous activities and constraints of diverse nature.

Although there exists related work on declarative BP modeling [28, 31, 20, 9, 17, 10, 4], only few approaches pay attention to the temporal perspective from a wider point of view, i.e., beyond viewing temporal constraints solely just in the sense of ordering relations [20, 9, 17, 16, 32, 15, 26, 10, 4]. To assess to what extent these works may support the temporal constraints required by the considered scenario, we performed a review which revealed that there exists other works that partially addresses some of the requirements related to the time patterns (i.e., activity durations and time lags between activities). Unlike TConDec-R, existing works do not consider other requirements that may refer to a calendar or schedule, and repetitions of process elements within a pre-specified timeframe. Therefore, TConDec-R opens a new opportunity for (1) further scenarios that were not supported by the other declarative BP languages, and (2) existing models that can be refined by including the TConDec-R constraints in order to improve the accuracy of the model, i.e., the similarity between the real and the modeled behavior.

Although there are other promising and interesting scenarios where declarative paradigm plays an important role in their effective management [8], we choose a clinical process scenario as it is on the one hand so challenging with respect to the constraints to be supported that the current state of the art was unable to effectively deal with it, and on the other hand not so "exotic" that the tool support needed would not be useful for other application domains as well, making clinical processes the 'killer application' for most declarative approaches currently available.

To demonstrate the applicability and expressiveness of the presented approach, the healthcare scenario is modeled with TConDec-R. The modeling of such scenario entailed a rather high complexity. Despite this complexity, the TConDec-R language could be successfully used for modeling the considered scenario without need for any workaround or extension. Nonetheless, the proposed approach is not only suitable for the illustrated clinical scenario, but for a class of processes with (1) flexible nature, i.e., scenarios for which declarative

specifications fit better than imperative ones when designing the process, and (2) temporal constraints playing an important role, i.e., the enactment of process activities must obey complex temporal relations. Hence, we strongly believe that the proposed approach can be successfully applied to many real-world scenarios for enabling flexible process support.

Additionally, to provide a validation of the language, a web-based tool has been implemented to support the TConDec-R.¹⁰ Such a tool allows both (1) modeling the clinical scenario of this work as well as other scenarios which require constraints related to the TPs [13], and (2) providing support for generating valid execution traces according to the modeled scenario. The latter has been conducted based on previous work [11, 2].

Altogether, we can ensure both the relevance and the novelty of the proposed BP modeling language, i.e., TConDec-R is a declarative approach that is able to cope with all temporal constraints related to the time patterns.

Note that the presented approach has revealed limitations as well. First, designers must deal with a new language for the constraint-based specification of processes. Thus, training is required to make them familiar with TConDec-R. Moreover, although declarative approaches enable a high degree of flexibility, problems in understanding and maintaining the respective process models often impede their adoption [33, 7]. Several works related to the understanding of declarative process models have been conducted (e.g., [33, 7]). However, respective research should also incorporate temporal constraints to make the results of such studies applicable to the current proposal.

6 Summary and Outlook

This paper analyzed a real-world scenario from healthcare being subject to complex temporal constraints. Thereafter, a review of proposals that may support the time patterns of the scenario was conducted. Unlike existing proposals, TConDec-R allows specifying all constraints related to process time patterns in a comprehensive way. Additionally, we demonstrated that all requirements of the healthcare scenario can be modeled with TConDec-R. Finally, a tool was presented for managing TConDec-R models.

In future work, we will study additional real-world process scenarios in other domains to demonstrate broad applicability of the approach. Furthermore, we will investigate the use and validation of algorithms to improve the support of TConDec-R in several respects, e.g., to provide personal schedules or generate predictions. Related to that, an empirical evaluation for analyzing the efficiency of the proposed approach will be performed. Moreover, the proposed language is intended to be evaluated in terms of usability and scalability. In addition, we will extend the proposed approach by enhancing it with the data perspective as well. Finally, we plan to improve the presented tool in order to (1) enable features like optimization [11], recommendations [3], and predictions [12], and (2) make it stable enough to share it as an open source project.

¹⁰ It is available at <http://azarias.lsi.us.es/TCR/ModelChecker>

Acknowledgments: This research has been supported by the Pololas project (TIN2016-76956-C3-2-R) and by the SoftPLM Network (TIN2015-71938-REDT) of the Spanish Ministerio de Economía y Competitividad.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.
2. I. Barba, A. Lanz, B. Weber, M. Reichert, and C. del Valle. Optimized time management for declarative workflows. In *BPMS*, volume 113 of *LNBIP*, pages 195–210. Springer, 2012.
3. I. Barba, B. Weber, C. del Valle, and A. Jimenez-Ramirez. User recommendations for the optimized execution of business processes. *Data & Knowledge Engineering*, 86:61 – 84, 2013.
4. A. Burattin, F.M. Maggi, and A. Sperduti. Conformance checking based on multi-perspective declarative process models. *Expert Systems with Applications*, 65:194–211, 2016.
5. M. de Leoni, F.M. Maggi, and W.M.P. van der Aalst. Aligning event logs and declarative process models for conformance checking. In *Proc. BPM*, volume 7481 of *LNCS*, pages 82–97. Springer, 2012.
6. M. Dumas, W.M.P. van der Aalst, and A.H. ter Hofstede, editors. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley-Interscience, 2005.
7. C. Haisjackl, I. Barba, S. Zugal, P. Soffer, I. Hadar, M. Reichert, J. Pinggera, and B. Weber. Understanding Declare models: strategies, pitfalls, empirical results. *Software and System Modeling*, 15(2):325–352, 2016.
8. E. Heuck, T. Hildebrandt, R. Kiærulff Lerche, M. Marquard, H. Normann, R. Iven Strømsted, and B. Weber. Digitalising the general data protection regulation with dynamic condition response graphs. In *Proc. BPM*, pages 124–134. 2017.
9. T. Hildebrandt, R.R. Mukkamala, T. Slaats, and F. Zanitti. Contracts for cross-organizational workflows as timed dynamic condition response graphs. *The Journal of Logic and Algebraic Programming*, 82(5):164–185, 2013.
10. Y. Jiang, N. Xiao, Y. Zhang, and L. Zhang. A novel flexible activity refinement approach for improving workflow process flexibility. *Computers in Industry*, 80:1–15, 2016.
11. A. Jimenez-Ramirez, I. Barba, C. del Valle, and B. Weber. Generating multi-objective optimized business process enactment plans. In *Proc. CAiSE*, volume 7908 of *LNCS*, pages 99–115. Springer, 2013.
12. A. Jimenez-Ramirez, I. Barba, J. Fernandez-Olivares, C. del Valle, and B. Weber. Time prediction on multi-perspective declarative business processes. *Knowledge and Information Systems*, pages 1 – 31, in press.
13. A. Lanz, M. Reichert, and B. Weber. Process time patterns: A formal foundation. *Information Systems*, 57:38–68, 2016.
14. A. Lanz, B. Weber, and M. Reichert. Time patterns for process-aware information systems. *Requirements Engineering*, 19(2):113–141, 2014.
15. F.M. Maggi. Discovering metric temporal business constraints from event logs. In *Proc. BIR*, volume 194 of *LNBIP*, pages 261–275. Springer, 2014.

16. F.M. Maggi, M. Dumas, L. García-Bañuelos, and M. Montali. Discovering data-aware declarative process models from event logs. In *Proc. BPM*, volume 8094 of *LNCS*, pages 81–96. Springer, 2013.
17. F.M. Maggi and M. Westergaard. Using timed automata for *a Priori* warnings and planning for timed declarative process models. *International Journal of Cooperative Information Systems*, 23(1):1440003, 2014.
18. R. S. Mans, N. C. Russell, W.M.P. van der Aalst, P.J.M. Bakker, and A.J. Moleman. Simulation to analyze the impact of a schedule-aware workflow management system. *Simulation*, 86(8-9):519–541, 2010.
19. S. Mertens, F. Gailly, and G. Poels. Enhancing declarative process models with dmn decision logic. In *Proc. BPMDS*, volume 214 of *LNBP*, pages 151–165. Springer, 2015.
20. M. Montali, F.M. Maggi, F. Chesani, P. Mello, and W.M.P. van der Aalst. Monitoring business constraints with the event calculus. *ACM Transactions on Intelligent Systems and Technology*, 5(1):17, 2013.
21. N. Mulyar, M. Pesic, W.M.P. van der Aalst, and M. Peleg. Declarative and procedural approaches for modelling clinical guidelines: Addressing flexibility issues. In *Proc. BPM*, volume 4928 of *LNCS*, pages 335–346. Springer, 2008.
22. Ovarian cancer (CG122). <http://www.nice.org.uk/CG122>, 2011. [Online; accessed 11-January-2017].
23. M. Pesic. *Constraint-Based Workflow Management Systems: Shifting Control to Users*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2008.
24. M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems*. Springer, 2012.
25. M. Rovani, F.M. Maggi, M. de Leoni, and W.M.P. van der Aalst. Declarative process mining in healthcare. *Expert Systems with Applications*, (23):9236–9251, 2015.
26. S. Schönig, C. Di Ciccio, F.M. Maggi, and J. Mendling. Discovery of multi-perspective declarative process models. In *Proc. ICSSOC*, volume 9936 of *LNCS*, pages 87–103. Springer, 2016.
27. B. Schultheiß, J. Meyer, R. Mangold, T. Zemmler, and M. Reichert. Designing the processes for ovarian cancer surgery. Technical Report DBIS-6, University of Ulm, 1996.
28. W.M.P. van der Aalst, M. Pesic, and M.H. Schonenberg. Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*, 23(2):99–113, 2009.
29. K. van Hee, H. Schonenberg, A. Serebrenik, N. Sidorova, and J.M. van der Werf. Adaptive workflows for healthcare information systems. In *Proc. BPM*, volume 4928 of *LNCS*, pages 359–370. Springer, 2008.
30. M. Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.
31. M. Westergaard and F.M. Maggi. Looking into the future: Using timed automata to provide a priori advice about timed declarative process models. In *Proc. CoopIS*, volume 7565 of *LNCS*, pages 250–267. Springer, 2013.
32. M. Zeising, S. Schönig, and S. Jablonski. Towards a common platform for the support of routine and agile business processes. In *Proc. CollaborateCom*, pages 94–103, 2014.
33. S. Zugal, P. Soffer, C. Haisjackl, J. Pinggera, M. Reichert, and B. Weber. Investigating expressiveness and understandability of hierarchy in declarative business process models. *Software and System Modeling*, 14(3):1081–1103, 2015.