Towards a new Tool for Managing Declarative Temporal Business Process Models

Andres Jimenez-Ramirez, Irene Barba, and Carmelo Del Valle

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Spain
{ajramirez,irenebr,carmelo}@us.es

Abstract. Business processes which require a high flexibility are commonly specified in a declarative (e.g., constraint-based) way. In general, offering operational support (e.g., generating possible execution traces) to declarative business process models entails more complexity when compared to imperative modeling alternatives. Such support becomes even more complex in many real scenarios where the management of complex temporal relations between the process activities is crucial (i.e., the temporal perspective should be managed). Despite the needs for enabling process flexibility and dealing with temporal constraints, most existing tools are unable to manage both. In a previous work, we then proposed TConDec-R, which is a constraint-based process modeling language which allows for the specification of temporal constraints. In this paper we introduce the basis and a prototype of a constraint-based tool with a client/server architecture for providing operational support to TConDec-R process models.

Keywords: constraint satisfaction problems, constraint programming, business process modeling support, process flexibility

1 Introduction

There is an increasing interest in aligning information systems in a process-oriented way [24]. Thereby, a business process (BP) consists of a set of activities that jointly realize a business goal and whose execution needs to be coordinated in an organizational as well as technical environment [24]. In this context, Declarative business process languages represent a promising modeling alternative in scenarios in which a high level of flexibility is demanded. Therefore, declarative approaches are becoming increasingly popular as they are able to cope with some of the limitations imperative notations are facing [23, 18, 16, 9, 4].

Although declarative modeling languages have been extensively discussed in literature, to the best of our knowledge, they have not been used to model complex scenarios that comprise constraints going beyond control-flow. This might be due to the fact that offering operational support for such models can be quite challenging since usually many constraints need to be obeyed [13, 4], multiple instances of a process get concurrently executed within a particular timeframe and shared resources need to be allocated [25, 12]. Furthermore, the time perspective has not received sufficient attention yet. In today's fast paced world, for any enterprise it is crucial to know the temporal properties of its business processes [5, 6, 1, 14].

To fill this gap, in a previous work [2] we proposed the TConDec-R language, a declarative process modeling language that allows for the specification of temporal constraints related to commonly recurring time patterns [14].

In this paper, we propose a constraint-based tool for supporting the use of TConDec-R language. More specifically, this tool allows:

1. Modeling declarative business processes through the TConDec-R language.
2. Checking the correctness of TConDec-R models.
3. Generating execution traces for such models.
4. Checking the conformance of given traces regarding a specific model.

The remainder of the paper is organized as follows. Section 2 presents the background on the TConDec-R language. Section 3 describes the architecture of the proposed constraint-based tool. Finally, Section 4 concludes the paper.

2 TConDec-R Language

This section provides backgrounds on constraint-based process models, which are needed for understanding this work.

As basis of the TConDec-R language [2], we use Declare [21] for specifying activities and their behavioral (i.e., control-flow) constraints. We consider this declarative modeling language as appropriate as it enables the specification of a wide range of process models in a flexible way. Such 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 fulfilled in this context. Declare constraints can be categorized as existence constraints, relation constraints, and negation constraints [21].

In addition, TConDec-R extends Declare to allow the specification of 10 process time patterns (TPs) that we systematically identified in [15, 14] by analyzing a large collection of process models from various domains.

Table 1 shows an example of the 4 most common of the 10 TPs divided into two categories according to pattern semantics.[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).

To properly cover the resource perspective, existing works (e.g., [2, 3, 12, 17, 20, 18, 19]) extended constraint-based specifications by additionally considering resource constraints for each enactment of a process activity. Few works [2, 18, 7,

[1] The full set of time patterns are grouped in 4 categories. The reader is referred to [15] for details.

Table 1. Selected process time patterns and examples.

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

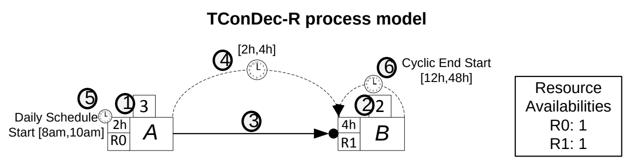

Fig. 1. A simple TConDec-R process model.

16, 9, 4] enhanced constraint-based specifications with temporal constraints. In this context, TConDec-R language considers both the resource and the temporal perspectives. [2]

Definition 1. *(**TConDec-R activity**). A **TConDec-R activity** act* $=$ $(a, dur, role)$ *refers to a process activity a with its estimated duration dur and the role of the required resource.*

Definition 2. *(**TConDec-R process model**). A **TConDec-R process model** TCRM= (Acts,* C_T*, Res) corresponds to an extended constraint-based process model, where Acts corresponds to a set of TConDec-R activities,* 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), and Res represents the resource availability.*

A TConDec-R model is said to be correct if it represents a feasible problem without conflicts (i.e., there are some traces that satisfy the model). TConDec-R constraints are specified according to the graphical notation proposed for Declare constraints [21] and using the graphical notation proposed in [14] for visualizing the temporal constraints.

Example 1. Figure 1 shows a simple example of a TConDec-R process model where $Acts = \{(A, 2h, R0), (B, 4h, R1)\}$, Res corresponds to $\{(R0, 1), (R1, 1)\}$, and C_T comprises (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,

[2] This paper focus on the temporal perspective of TConDec-R. Details of other features like the resource perspective can be found in [2].

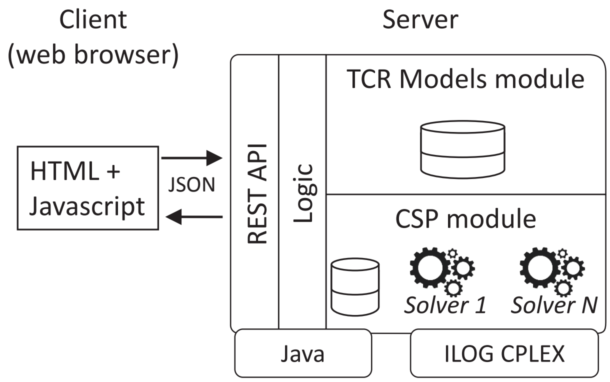

Fig. 2. Proposed architecture.

(3) $Precedence(A, B)$, expressing that activity B may only be executed if A is executed before, (4) $TimeLagEndStart(A, B, 2h, 4h)$, expressing that for each execution of A, there must be at least one execution of B such that there is a time lag of at least 2 hours and at most 4 hours, (5) $DailyScheduleStart(A, 8am, 10am)$, expressing each execution of A must be started between 8am and 10am, and (6) $CyclicEndStart(B, 12h, 48h)$, expressing that between the end and start of two succeeding executions of B, there must be a time lag of at least 12h and at most 48h.

When executing a constraint-based process model, information about the executed activities is recorded in an execution trace.

Definition 3. *(**Trace**). Let $TCRM = (Acts, C_T, Res)$ be a TConDec-R process model. Then, a **trace** $\sigma = (ID, < e_1, e_2, ...e_n >)$ consists of an identifier ID and a sequence of start and completion events respectively. Thereby, an event e relates to a specific execution (e.g., the i-th execution) of a TConDec-R activity $(a, dur, role) \in Acts$ (such execution is denoted by a_i) and has one of the following two forms: (1) $e = start(a_i, Rj_k, T)$, i.e., the i-th enactment of activity a using the k-th resource with role j was started at time T, or (2) $e = comp(a_i, T)$, i.e., the i-th enactment of activity a was completed at time T.*

In general, a constraint-based process model (e.g., a TConDec-R process model) can be modeled as a Constraint Satisfaction Problem (CSP), as detailed in [2]. The latter is a key concept in Constraint Programming (CP) [22], which is a powerful paradigm for correctly modeling and solving a wide variety of problems (e.g., combinatorial problems).

As detailed in previous work [2], once a constraint-based process model is modeled as a CSP, several operational support tasks are enabled like generating valid traces [3], process optimization [10] or providing predictions [11].

3 Contraint-based Software Tool

The tool presented in this paper is implemented as a client-server application (cf. Fig. 2). Both ends can be deployed separately and connected through the

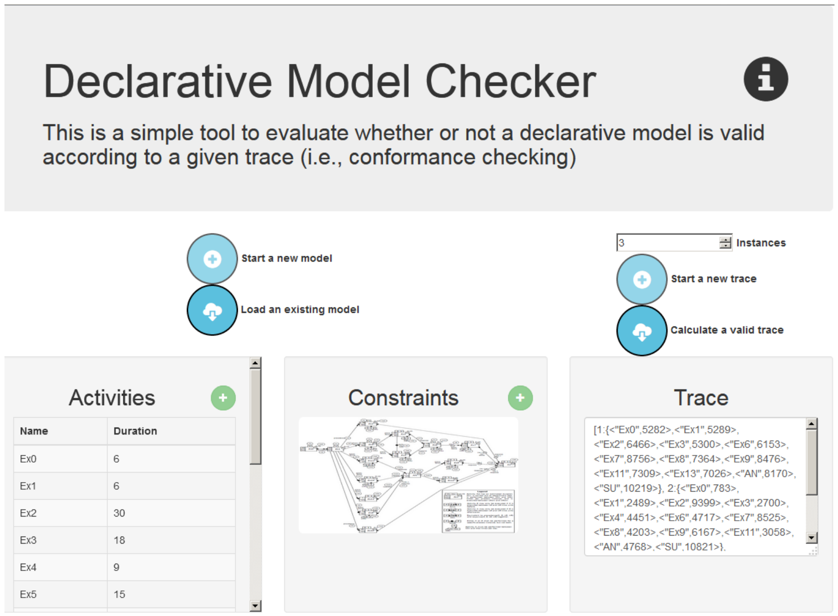

Fig. 3. HTML-based user interface of the functional prototype

REST API layer of the server. Moreover, the client side (cf. Sect. 3.1) deploys a light-weight web interface for modeling declarative business processes using the TConDec-R language. In turn, the server side (cf. Sect. 3.2) is in charge of (1) transforming the problem that is specified through the tool into a CSP, (2) solving the CSP using a solver and (3) interpreting the solution which is obtained by the solver.

3.1 Client Side

The client comprises an HTML-based user interface (cf. Fig. 3). It can be considered as a light-weight client since it is only in charge of making the functionality offered by the server accessible.

The user is enabled to create TConDec-R specifications by including activities and constraints which are textually shown in the interface. In addition, previously created models can be loaded from the server and edited in the client through the HTML-based interface. Once models are defined and the resources which may be available in runtime are stated, different actions can be performed.

For checking the correctness of the model previously specified, the server is requested for looking if the model can be instantiated or not. As explained latter, the server deploys a set of CP mechanisms for addressing this task in the CSP module. In case that the CSP module finds a solution, it means that the model is correct. In case that it explores the complete search space and there is not a solution, it means that the model is incorrect. Finally, in case that the CSP module is not able to find a solution in the given time,[3] it means that the model is too complex and the solver needs more time to elevate a conclusion beyond that.

Similar to the previous point, in order to generate traces for a given model the server will look for an instance regarding such model. In case that the model is correct, the user receives the trace which is calculated by the CSP module.

Finally, for checking the conformance of the model with a trace (or partial trace), the server will look for an instance regarding such model where all the evens of the given trace are reproducible on the instance. In case that the CSP finds a solution, it means that the conformance is checked.

3.2 Server Side

The server comprises two parts. First, the REST API which exposes the functionality in a way that it can be consumed by any client independently of the language in which it is implemented. In summary, such interface offers a series of endpoints that can be accessed via HTTP requests which trigger the different supported functionalities.

Second, the logic which eventually implements the desired functionality. For this, two different main modules are implemented. On the one hand, for managing the models there is an independent module which is in charge of that part (cf. TCR Models module in Fig. 2). Models can be created, retrieved, updated or eliminated from the system. Such module is written in Java language and stores the information in a local database.

On the other hand, the CSP module is in charge of the complex tasks, i.e., checking the correctness of models, generating traces and checking the conformance of traces. With the aim to make the architecture independent of a CSP language, this module implements inner connectors to the CSP solvers. Therefore, the CSP module first transforms the desired complex task into a CSP. Secondly, the module orchestrates the necessary executions of the solver. Finally, the solver solutions are gathered to compose the solution of the complex task which is returned by the module. Part of this module is written in Java while other parts are written in different solver languages. Currently, ILOG CPlex [8] is used as a solver.

4 Conclusion

In the current work, we build upon a declarative business process modeling language which allows specifying sophisticated temporal constraints, i.e., the TConDec-R language [2]. Although there exists related work on declarative BP modeling [23, 18, 16, 4], only few approaches pay attention to the temporal perspective from a wider point of view. Unlike TConDec-R, existing works do not

[3] Since the considered problems present a NP complexity, a time limit is established when solving the CSPs.

consider other requirements such as the support of constraints that may refer to a calendar or schedule, and time-based constraints.

Taking the TConDec-R language [2] as basis, this paper is focused on a constraint-based tool to support the TConDec-R models.[4] Such a tool allows (1) modeling scenarios through the TConDec-R language, (2) checking if the scenarios are correctly modeled, i.e., they can be instantiated, (3) generating valid execution traces according to such models, and (4) checking the conformance of execution traces.

Although the developed tool can be merely considered as a prototype, we strongly believe that the proposed approach can be successfully applied in many sophisticated scenarios for enabling flexible process support. This is faced by integrating the high-level abstraction of BP in the CP context and contributes on improving the maturity of the declarative technology.

As future work, we will investigate the use and validation of constraint-based algorithms to improve the support to TConDec-R in several respects, e.g., to provide personal schedules or generate time predictions. In addition, we will further extend the proposed approach by considering the data perspective of business process as well as the temporal and the control-flow. Finally, it is planned to improve the presented tool in order to (1) enable additional features like process optimization, recommendations or predictions, and (2) make it stable enough to share it as an open source project.

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

References

1. C. Arevalo, M.J. Escalona, I. Ramos, and M. DomÃŋnguez-MuÃśoz. A metamodel to integrate business processes time perspective in bpmn 2.0. *Information and Software Technology*, 77:17–33, 2016.
2. I. Barba, A. Lanz, B. Weber, M. Reichert, and C. del Valle. Optimized time management for declarative workflows. In *Enterprise, Business-Process and Information Systems Modeling*, volume 113 of *LNBIP*, pages 195–210. Springer Berlin Heidelberg, 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(0):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. P. Dadam, M. Reichert, and K. Kuhn. Clinical workflows - the killer application for process-oriented information systems? In *BIS 2000*, pages 36–59. Springer, 2000.
6. Johann Eder, Euthimios Panagos, and Michael Rabinovich. Time constraints in workflow systems. In *Advanced Information Systems Engineering*, volume 1626 of *LNCS*, pages 286–300. 1999.

[4] It is available at http://azarias.lsi.us.es/TCR/ModelChecker

7. 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.
8. IBM. CPLEX CP Optimizer. urlhttp://www-01.ibm.com/software/commerce/optimization/cplex-cp-optimizer, 2016. [Online; accessed 11-July-2016].
9. 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.
10. A. Jimenez-Ramirez, I. Barba, C. del Valle, and B. Weber. Generating multi-objective optimized business process enactment plans. In *Advanced Information Systems Engineering*, volume 7908 of *LNCS*, pages 99–115. Springer Berlin Heidelberg, 2013.
11. 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.
12. A. Jimenez-Ramirez, I. Barba, Weber, B., and C. Del Valle. Generating optimized configurable business process models in scenarios subject to uncertainty. *Information Software Technology*, 57:571–594, 2015.
13. D. Knuplesch, M. Reichert, and A. Kumar. A framework for visually monitoring business process compliance. *Information Systems*, 64:381–409, 2017.
14. A. Lanz, M. Reichert, and B. Weber. Process time patterns: A formal foundation. *Information Systems*, 57:38–68, 2016.
15. A. Lanz, B. Weber, and M. Reichert. Time patterns for process-aware information systems. *Requirements Engineering*, 2012.
16. 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), 2014.
17. M. Montali. *Specification and Verification of Declarative Open Interaction Models: a Logic-Based Approach*. PhD thesis, Department of Electronics, Computer Science and Telecommunications Engineering, University of Bologna, 2009.
18. M. Montali, F.M. Maggi, F. Chesani, P. Mello, and W.M.P. van der Aalst. Monitoring business constraints with the event calculus. *ACM TIST*, 5(1):17, 2013.
19. Marco Montali, Federico Chesani, Paola Mello, and Fabrizio M. Maggi. Towards data-aware constraints in declare. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, pages 1391–1396, 2013.
20. Marco Montali, Maja Pesic, Wil M. P. van der Aalst, Federico Chesani, Paola Mello, and Sergio Storari. Declarative specification and verification of service choreographiess. *ACM Trans. Web*, 4(1):3:1–3:62, 2010.
21. M. Pesic. *Constraint-Based Workflow Management Systems: Shifting Control to Users*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2008.
22. F. Rossi, P. van Beek, and T. Walsh, editors. *Handbook of Constraint Programming*. Elsevier, 2006.
23. 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.
24. Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.
25. J. Xu, C. Liu, X. Zhao, S. Yongchareon, and Z. Ding. Resource management for business process scheduling in the presence of availability constraints. *ACM Transactions on Management Information Systems*, 7(3), 2016.