Providing Support for the Optimized Management of Declarative Processes

Irene Barba[1], Andreas Lanz[2] Andrés Jiménez-Ramírez[1], Barbara Weber[3], Manfred Reichert[2], and Carmelo Del Valle[1]

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

[2] Institute of Databases and Information Systems, Ulm University, Germany
{Andreas.Lanz,Manfred.Reichert}@uni-ulm.de

[3] Department of Computer Science, University of Innsbruck, Austria,
Technical University of Denmark, Denmark
bweb@dtu.dk

Abstract. Declarative process models are becoming increasingly popular due to the high flexibility they offer to process participants. Based on a declarative process model, there exist numerous possible enactment plans, each one with specific values for relevant objective functions (e.g., overall completion time). How to actually execute such a model is quite challenging due to several reasons: (1) proper objective functions must be considered to find optimized enactment plans, (2) users often do not have an understanding of the overall process, (3) the presence of a variety of temporal constraints to be met during process enactment, and (4) the need to coordinate multiple instances of a process concurrently executed (which compete for shared resources). This is further complicated by the fact that the enactment of new process instances may continuously start over time and many organizations do not exactly know their future demands. In such context, to properly support users in enacting declarative process models, this paper suggests generating optimized enactment plans from declarative process models. The generated enactment plans may be used for different purposes, e.g., to provide personal schedules to users. Moreover, they may be dynamically adapted if required. To evaluate the applicability of our approach in practical settings we apply it to a real process scenario from the healthcare domain.

Keywords: Process flexibility, declarative process model, temporal constraints, constraint programming, scheduling, healthcare processes

1 Introduction

For several years, there has been an increasing interest in aligning information systems in a process-oriented way [2, 15]. Usually, business processes (BPs) have numerous constraints to be obeyed during process enactment. To provide operational support, BPs are mostly specified in an imperative way, i.e., defining a precise schema that establishes how a given set of activities has to be performed. However, imperative process models are often too rigid to meet the flexibility requirements of users. As an alternative, therefore, declarative process models are increasingly used [13, 9] since they provide an

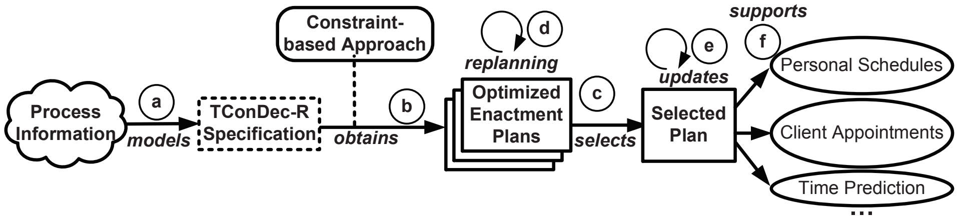

Fig. 1. Overview of our approach.

increased flexibility when executing the process, allowing users to specify *what* has to be done instead of *how* [8].

On one hand, a declarative process model offers a high flexibility to end users by allowing for the required degree of freedom. On the other, executing a declarative model usually entails larger efforts for users compared to imperative models [10]. In general, numerous enactment plans related to the same declarative model exist, each one presenting specific values for relevant objective functions (e.g., overall completion time). Consequently, the decision on how to execute a declarative model is quite complex. It becomes even more challenging when dealing with business processes that (1) contain complex temporal as well as cross-instance constraints, (2) require an efficient management of shared resources, and (3) need to consider the optimization of specific objective functions in an uncertain and changing environment. In such a context, usually, users neither know the complete BP nor the impact of their actions. Both might result in sub-optimal enactment plans. Therefore, advanced user support is required when executing declarative process models. Such support can be provided by automatically generating optimized enactment plans from the respective declarative process model. Moreover, during enactment it should be possible to flexibly adapt these plans if required taking relevant run-time information into account.

Optimized enactment plans foster sophisticated user support through: (1) personal schedules [4], i.e., assisting process participants by recommending which activities they shall perform when, (2) reduced turnaround and waiting times, i.e., organizing appointments while taking the start times of the activities the users are involved in into account, and (3) time predictions, i.e., predicting enactment times for future activities. Note that (1) allows for a better planning of work, while (3) enables decision support for process participants [14].

2 Contribution

This paper suggests the generation of optimized enactment plans for a set of process instances derived from a declarative model. Figure 1 provides an overview of our approach. First, the declarative specification of the BP is defined (Step a in Fig. 1). For this, we consider the declarative language DECLARE[1] [13] as basis. In order to address the described problems, DECLARE is extended by (1) including capabilities for reasoning about resources as well as the estimated duration of process activities, and

[1] DECLARE is one of the most referenced and used declarative process modeling languages.

(2) supporting common time patterns reported in literature [6, 5] (e.g., cross-instance constraints related to time). This results in the TConDec-R language.

From a TConDec-R specification, in turn, optimized enactment plans can be automatically generated (Step b in Fig. 1). In this context, the activities to be executed are selected and ordered (i.e., *planning problem* [3]), considering control-flow constraints as well as temporal and resource constraints imposed by the declarative specification (i.e., *scheduling problem* [1]). For planning and scheduling the activities in a way such that a given objective function becomes optimized (i.e., for generating a set of optimized enactment plans), a constraint-based approach can be proposed. Finally, from the set of optimized enactment plans (i.e., schedules), the plan that fits best to the scenario requirements is selected for enactment (Step c in Fig. 1). The selected plan is then used for improving process support in several respects (Step f in Fig. 1).

Additionally, the proposed approach supports the dynamic adaptation of optimized enactment plans during run-time through replanning, and hence enables an increased run-time flexibility (Step d in Fig. 1). Each time the set of optimized enactment plans is updated, a new plan for providing process support is selected. In general, replanning might become necessary either if the actual enactment of the process instances deviates from the optimized enactment plan generated earlier (e.g., estimates might turn out to be inaccurate or resource availability might unexpectedly change) or in scenarios in which the enactment of new process instances continuously starts and the demand of the services to be provided varies over time. Respective scenarios are known as lot-sizing ones with uncertain demand [12].

To evaluate the applicability of our approach in practical settings we consider to apply it to a real process scenario from the healthcare domain. The considered scenario deals with the scheduling of surgeries and their preparations in the context of ovarian carcinoma [11, 7]. The application of the proposed approach to the medical scenario offers several advantages: (1) avoiding constraint violations, (2) managing shared resources in an effective way, (3) improving and automating (inter-process) coordination between different hospitals, (4) reducing waiting times of patients as the schedule becomes known beforehand, and (5) minimizing the length of patient stays in hospitals.

Note that the approach is not restricted to healthcare environments, i.e., it can be also applied in other domains for which process flexibility and temporal constraints play a crucial role (e.g., automotive engineering and flight planning [6]).

The proposed approach combines original contributions regarding (1) the formal specification and operational support of complex temporal constraints during process enactment, (2) cross-instance coordination, (3) resource allocation (i.e., scheduling) before and during process enactment, and (4) optimization of objective functions.

3 Conclusion

This paper proposed the generation of optimized enactment plans (e.g., minimizing overall completion time) from declarative temporal process models. This generation is quite challenging since it considers complex temporal constraints, cross-instance coordination, resource allocation before and during process enactment, and optimization of a given objective function. It is further complicated by the fact that the enactment of new process instances may continuously start over time and many organizations do not exactly know their future demands.

The generated plans can be used for different purposes, e.g., providing users with a personal schedule, suggesting appointments to process stakeholders, or predicting enactment times for process activities. Moreover, these plans are adapted during the enactment of the process if necessary.

As future work, we will test our approach in the context of ovarian carcinoma.

References

1. P. Brucker and S. Knust. *Complex Scheduling (GOR-Publications)*. Springer, 2006.
2. M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley-Interscience, Hoboken, NJ, 2005.
3. M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, Amsterdam, 2004.
4. A. Lanz, J. Kolb, and M. Reichert. Enabling personalized process schedules with time-aware process views. In *CAiSE'13 Workshops*, LNBIP, pages 205–216. Springer, 2013.
5. A. Lanz, M. Reichert, and B. Weber. Process time patterns: A formal foundation. *Information Systems*, 57:38–68, 2016.
6. A. Lanz, B. Weber, and M. Reichert. Time patterns for process-aware information systems. *Requirements Engineering*, 19(2):113–141, 2014.
7. Ovarian cancer (CG122). `http://www.nice.org.uk/CG122`, 2011. [Online; accessed 7-April-2016].
8. M. Pesic. *Constraint-Based Workflow Management Systems: Shifting Control to Users*. PhD thesis, Technische Universiteit Eindhoven, The Netherlands, 2008.
9. P. Pichler, B. Weber, S. Zugal, J. Pinggera, J. Mendling, and H.A. Reijers. Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. In *Proc. BPM Workshops*, pages 383–394, 2011.
10. M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems*. Springer, 2012.
11. B. Schultheiß, J. Meyer, R. Mangold, T. Zemmler, and M. Reichert. Designing the processes for ovarian cancer surgery (in german). Technical Report DBIS-6, University of Ulm, 1996.
12. L. Tiacci and S. Saetta. Demand forecasting, lot sizing and scheduling on a rolling horizon basis. *International Journal of Production Economics*, 140(2):803–814, 2012.
13. 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.
14. W.M.P. van der Aalst, M.H. Schonenberg, and M. Song. Time prediction based on process mining. *Information Systems*, 36(2):450–475, 2011.
15. M. Weske. *Business Process Management: Concepts, Languages, Architectures (Second Edition)*. Springer, 2012.