

CombiS-BP Editor: Combining Declarative and Imperative Languages in BP Modelling

Demonstration paper

Luisa Parody, María Teresa Gómez-López, Rafael M. Gasca, Angel Jesus Varela-Vaca
Computer Languages and Systems Department,
University of Seville, Seville, Spain
{lparody, maytegonomez, gasca, ajvarela}@us.es

Abstract—Business Process models allow business experts to describe the activities that must be performed to achieve a defined goal. Several imperative standard languages have been created to describe both control-flow and data-flow perspectives. Unfortunately, the imperative specification may be very difficult, complex and even inefficient to any kind of models, particularly, when these models depend on input data of each specific process instance. On the other hand, although business experts are able to define a declarative specification, they may be unaware of translating it into an imperative process model by using the existing standard languages. As a solution, we present CombiS-BP Editor, a tool that enables combined modelling of the two aforementioned specifications: (i) allows an imperative specification when experts know the execution order of the activities in the model and; (ii) enables a declarative specification when the experts know *what* has to be done instead of *how* it has to be done.

I. INTRODUCTION

In management theory of the last years, a process-oriented perspective has been considered the shell on organizational (re)structuring. Nowadays, organizations still experience difficulties in applying this process-oriented perspective to the design and maintenance of their information systems. Many of the problems deal with the imperative representation of business processes, since they contain unsuitable information for computer systems to provide flexible automated business process support. Specifically, a business process, henceforth referred as BP, consists of a set of activities that are performed in coordination within an organizational and technical environment. These activities jointly perform a business goal [1].

The use of BP modelling is becoming increasingly more used. Typically, business processes are specified in an imperative manner, (e.g., by indicating that activity A is executed just after activity B ends). This imperative specification allows business experts to describe relationships between activities and to transform the process into an executable model. Therefore, an imperative description defines exactly how things have to be performed. Nevertheless, a declarative representation takes into account the business concerns that govern the BP. A BP may be exposed to different environments and subjected to many conditions in which not always a sequence of activities can be described at design time. This is the reason why several authors have proposed languages to define BP as declarative models ([2], [3], [4] and [5]). These declarative languages enable to describe temporal order of the activities, even depending on the values of the variables in the data flow. The problem of these proposals is that they do not allow to describe the activities order with the aim to optimise a function where the data flow variables are involved.

An example to understand the necessity of an imperative and declarative languages joined, to specify a BP, is the booking of a trip. In the organization of a travel, the activities to book hotel and flight tickets can be executed sequentially in different order, even in parallel. The problem arises when the activities order should be set in order to achieve the goal of minimizing the cost of the trip. For instance, by analysing different departure and return dates. It entails the activities should share their input data values, needing to be executed several times for each possible date to guarantee the minimal cost.

II. PROPOSED METHOD

Therefore, CombiS-BP Editor is proposed with the aim of combining imperative and declarative languages. This tool facilitates the use of both languages, delegating the creation of the model to a solver [6]. Particularly, CombiS-BP Editor makes possible to include BPMN 2.0 components in an imperative way by means of using a declarative element, called CombA Sub-Process, presented in [7]. CombA Sub-Process aims to define, in a declarative way, a combination of several activities to find out the specific values of the handle data that optimize an objective function where data flow is involved.

To the best of our knowledge, there are no available tools that enable the modelling of a BP where the order relation among the activities depends on the data values of each instance. In the search of existing tools that model declarative constraints in BP and workflows, various relevant tools were found ([8], [9], [10] and [11]). All of them use the value of the data to determine which task is executed next, deal with multiple instances and extend the standard functionality. Although these tools define a set of constraints that relate the activities, the aim of these constraints is to establish an execution order, that in our approach is unknown until the value of the data are analysed at runtime.

A. CombiS-BP Editor

Heretofore, with the existing BPMN 2.0 modelling tools, business experts had to decide the exact order (sequential, parallel, etc.) to model the part where the activities require combination and they could not know how to combine them. CombiS-BP Editor combines the imperative part of the model where experts know *how* things should be done, with the declarative part of the model where experts only know the activities that need to be executed and the optimization function. The data flow instance is taken into account in this declarative specification.

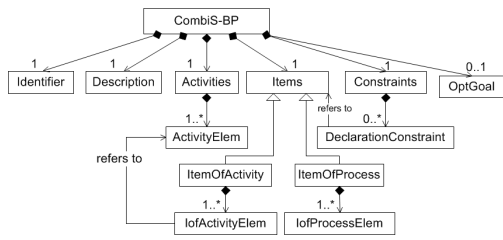


Fig. 1. Vocabulary in CombiS-BP Editor.

In order to include CombA Sub-Process and in order to describe the declarative part of the process, the grammar used by CombiS-BP Editor is shown in Fig. 1. CombA Sub-Process contains an *Identifier*, its functionality is described through a *Description* and its goal to be optimized is represented as *OptGoal*. The set of activities to be combined are represented as *Activities*. Each *ActivityElem* from *Activities* represents the features of each activity. Finally, the relationships between the activities are represented as *Constraints*, where some *DeclarationConstraints* are defined through *Items* (belonged to Process or to Activities).

CombiS-BP Editor has been developed as an extension of OPBUS tool ([12] and [13]), which is an eclipse plugin. CombiS-BP Editor integrates a BPMN 2.0 modeller that enables the creation of CombA Sub-Process (see Fig. 2). The user interface is composed of four main parts: edition zone, palette, properties and problem tabs, and a project workspace zone with basic menus. Business experts model the common BP in the edition zone. In this part, declarative and imperative specifications are differentiated through the different elements used in the process. The palette provides the graphical definition of BPMN elements (imperative specification) and CombA (declarative specification), which can be selected and dropped into the edition zone.

The marker associated to CombA Sub-Process graphical definition in the palette is a set of puzzle pieces symbol (such as defined in [7]), and to CombA Task (activities involved in the combination), the graphical definition is a unique puzzle piece symbol. CombA Flow is a solid line that can connect only CombATask with CombA Sub-Process. The declarative definition is completed with the properties part, which provides support for the definition of elements details. In order to fill the CombA Sub-Process grammar shown in Fig.1, some properties and elements are added. Specifically, the property *ItemOfActivity* and *ItemOfProcess* are added, to CombATask and CombA elements respectively, through a Data element from the palette. In the same way, the set of *Constraints* are added to CombA element and specify by the properties provided in the Constraint element (see Fig. 2). Finally, the properties, provided in the properties part, associated to CombA Sub-Process element are *Id*, *Description* and *Objective function*.

III. CONCLUSION

In this paper, an editor to combine imperative and declarative business process description is presented. The declarative proposal enables the definition of an optimization function where data flow variables are related. The grammar of the declarative language is included in a component that can

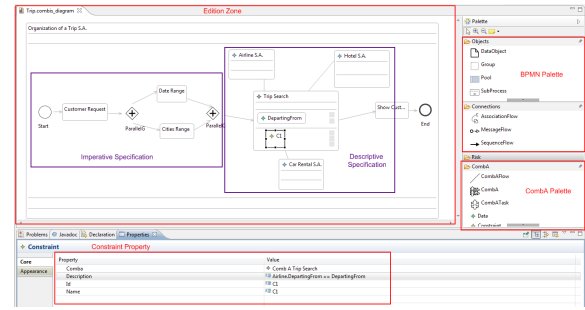


Fig. 2. CombiS-BP Editor.

be combined with BPMN 2.0 components, being transparent for the modeller how the model is created to optimize each instance of the BP. Thanks to this editor, it is possible to define a flexible and complete model, where the experts can specify *what* has to be done and check the consistency in an easy way.

ACKNOWLEDGMENT

This work has been partially funded by the Ministry of Science and Technology of Spain (TIN2009-13714) and the European Regional Development Fund (ERDF/FEDER).

REFERENCES

- [1] M. Weske, *Business Process Management: Concepts, Languages, Architectures*. Springer, 2007.
- [2] S. W. Sadiq, M. E. Orlowska, and W. Sadiq, "Specification and validation of process constraints for flexible workflows," *Information Systems*, vol. 30, no. 5, pp. 349–378, 2005.
- [3] I. Rychkova, G. Regev, and A. Wegmann, "High-level design and analysis of business processes: the advantages of declarative specifications," in *RCIS*, 2008, pp. 99–110.
- [4] M. Pesic and W. M. P. van der Aalst, "A declarative approach for flexible business processes management," in *Business Process Management Workshops*, ser. Lecture Notes in Computer Science, J. Eder and S. Dustdar, Eds., vol. 4103. Springer, 2006, pp. 169–180.
- [5] I. Rychkova, G. Regev, and A. Wegmann, "Using declarative specifications in business process design," *Journal of Computer Science*, vol. 5, no. 3, pp. 45–68, 2008.
- [6] L. Parody, M. T. Gómez-López, R. Martínez Gasca, and A. J. Varela-Vaca, "Improvement of Optimization Agreements in Business Processes involving Web Services," *Communications of the IBIMA*, vol. 2012, 2012.
- [7] L. Parody, M. T. Gómez-López, and R. M. Gasca, "Extending bpmn 2.0 for modelling the combination of activities that involve data constraints," in *BPMN*, 2012, pp. 68–82.
- [8] W. M. P. van der Aalst, L. Aldred, M. Dumas, and A. H. M. ter Hofstede, "Design and implementation of the yawl system," in *CAiSE*, 2004, pp. 142–159.
- [9] S. Goedertier and J. Vanthienen, "Em-bra2ce v0.1: A vocabulary and execution model for declarative business process modeling," in *Department of Decision Sciences and Information Management - KBI*, 2007.
- [10] M. Pesic, H. Schonenberg, and W. M. P. van der Aalst, "Declare demo: A constraint-based workflow management system," in *BPM (Demos)*, 2009.
- [11] O. Kopp, T. Binz, U. Breitenbücher, and F. Leymann, "Bpmn4tosca: A domain-specific language to model management plans for composite applications," in *Business Process Model and Notation*, vol. 125, 2012, pp. 38–52.
- [12] Opbus tools, "http://www.lsi.us.es/quivir/index.php/Main/AJVarelaOPBUS", 2012.
- [13] A. J. Varela-Vaca, R. M. Gasca, and S. Pozo, "Opbus: Risk-aware framework for the conformance of security-quality requirements in business processes," in *SECURITY*, 2011, pp. 370–374.