OptBPPlanner: Automatic Generation of Optimized Business Process Enactment Plans

Andrés Jiménez, Irene Barba, Carmelo del Valle, and Barbara Weber

Abstract Unlike imperative models, the specification of business process (BP) properties in a declarative way allows the user to specify *what* has to be done instead of having to specify *how* it has to be done, thereby facilitating the human work involved, avoiding failures, and obtaining a better optimization. Frequently, there are several enactment plans related to a specific declarative model, each one presenting specific values for different objective functions, e.g., overall completion time. As a major contribution of this work, we propose a method for the automatic generation of optimized BP enactment plans from declarative specifications. The proposed method is based on a constraint-based approach for planning and scheduling the BP activities. These optimized plans can then be used for different purposes like simulation, time prediction, recommendations, and generation of optimized BP models. Moreover, a tool-supported method, called OptBPPlanner, has been implemented to demonstrate the feasibility of our approach. Furthermore, the proposed method is validated through a range of test models of varying complexity.

1 Introduction

Nowadays, there exists an increasing interest in aligning information systems in a process-oriented way as well as in the effective management of business processes (BPs, i.e., sets of activities which are performed in coordination in an organization to achieve a business goal) [1]. BP management (BPM) supports BPs using

A. Jiménez • I. Barba • C. del Valle
Dpto.LenguajesySistemasInformáticos , University of Seville, Seville, Spain
e-mail: ajramirez@us.es; irenebr@us.es; carmelo@us.es

B. Weber
DepartmentofComputerScience , University of Innsbruck, Innsbruck, Austria
e-mail:barbara.weber@uibk.ac.at

methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, and other sources of information [2]. Typically, the traditional BPM life cycle [1] includes several phases, i.e., process design and analysis, system configuration, process enactment, and evaluation [1].

The quality of a BP design is essential for BP improvement, which has been ranked as the number one priority for top management by the 2010 Gartner survey [3]. When using an imperative approach, however, there are lots of manual work involved since the modelling expert has to describe exactly *how* it should be done. The usage of declarative processes, in turn, allows the user to specify *what* has to be done instead of *how*, thereby facilitating the human work involved and avoiding failures. There are frequently different ways to execute a declarative model in such a way that all constraints are fulfilled. The different execution alternatives, however, can vary significantly in how well different performance objective functions such as the overall completion time can be optimized. Thus, from the declarative process specification, optimized BP enactment plans can be automatically generated, which can greatly improve the overall BPM life cycle [1]. Specifically, these plans can be used, among others, for (1) simulation [4]; (2) time prediction [5], both improving the BP design and analysis phase; (3) recommendations [6], assisting users during process enactment; and (4) the generation of optimized BP models [7, 8], facilitating the human work which is involved in the BP design and analysis phase (cf. Sect. 5.1), which are innovative and interesting topics to be addressed nowadays. The main novelties of our approach regarding existing BP proposals are that our proposal (1) considers optimization and (2) deals with resource allocation.

In this work, we propose a tool-supported method, named OptBPPlanner,[1] for the automatic generation of optimized BP enactment plans from declarative process specifications to optimize the overall completion time. In this way, our proposal facilitates the work of the modelling expert since she only needs to specify the *what* and let our approach to decide the *how*. For this, activities to be executed need to be selected and ordered (planning problem [9]) considering both control flow and resource constraints (scheduling problem [10]) imposed by the declarative specification. For planning and scheduling (P&S) the activities such that the process objective function is optimized, a constraint-based approach is proposed. Moreover, the proposed approach has been validated through an empirical evaluation considering different test models of varying complexity (cf. Sect. 4).

This paper is organized as follows: Sect. 2 introduces background; Sect. 3 includes an overview of the proposed method including the associated tool support; Sect. 4 shows some experimental results; Sect. 5 presents a critical discussion of the advantages, drawbacks, and some applications of our proposal; Sect. 6 summarizes related work; and finally, Sect. 7 includes some conclusions and future work.

[1] A web-based application for the generation of optimized BP enactment plans from ConDec-R specifications can be accessed at http://regula.lsi.us.es/OptBPPlanner.

2 Background

We use the declarative language ConDec [11] as basis to specify constraint-based BP models (cf. Def. 1), since it allows the specification of BP activities together with the constraints which must be satisfied for correct BP enactment and for the goal to be achieved.

Constraints can be added to a ConDec model to restrict the desired behavior (cf. [11]). ConDec templates, i.e., parameterized graphical representation of constraints, are grouped into:

1. Existence templates: unary relations concerning the number of times one activity is executed, e.g., *exactly*(N,A) specifies that A must be executed exactly N times.
2. Relation templates: positive binary relations used to establish what should be executed, e.g., *precedence*(A,B) specifies that before any execution of activity B at least one execution of activity A must have been done.
3. Negation templates: negative relations used to forbid the execution of activities in specific situations, e.g., *notCoexistence*(A,B) specifies that if B is executed, then A cannot be executed, and vice versa.

Definition 1

A constraint-based process model $S = (A, C_{BP})$ consists of a set of activities A and a set of constraints C_{BP} limiting execution behaviors. For each activity $a \in A$, resource constraints can be specified by associating a role with that activity. The activities of a constraint-based process model can be executed arbitrarily often if not restricted by any constraints.

On the other hand, the area of scheduling [10] includes problems in which it is necessary to determine an enactment plan for a set of activities related by temporal constraints. Moreover, the execution of all activities requires the use of limited capacity resources. In general, the goal consists of finding a feasible plan which satisfies both temporal and resource constraints, optimizing certain objective functions (e.g., minimization of the overall completion time). In a wider perspective, in planning [9], the activities to be executed are not established a priori, hence it is necessary to select them from a set of alternatives and to establish an ordering.

In a related way, constraint programming (CP, i.e., a software technology for modelling and solving problems by using constraints to relate variables) supplies a suitable framework for dealing with P&S problems [12]. To solve a problem through CP, it needs to be modelled as a constraint satisfaction problem (CSP, cf. Def. 2).

Definition 2

A **CSP** $P = (V, D, C_{CSP})$ is composed of a set of variables V, a domain of values D for each variable in V, and a set of constraints C_{CSP} between variables, so that each constraint represents a relation between a subset of variables and specifies the allowed combinations of values for these variables.

A *solution* to a CSP consists of assigning values to CSP variables, being feasible when the assignments satisfy all the constraints. In CP, global constraints can be defined to improve the modelling of the problems. Similar to CSPs, constraint optimization problems (COPs, cf. Def. 3) require solutions that optimize certain objective functions.

Definition 3

A **COP** $Po = (V, D, C_{CSP}, OF)$ is a CSP which also includes an objective function, OF, to be optimized.

Several mechanisms are available for solving CSPs and COPs, e.g., complete search algorithm, i.e., performing a complete exploration which is based on all possible combinations of assignments of values to the CSP variables. Regardless of the used search method, the global constraints can be implemented through filtering rules (i.e., rules responsible for removing values which do not belong to any solution) to efficiently handle the constraints in the search for solutions.

3 Method for Generating Optimized Enactment Plans

In our approach, two steps can be differentiated: (1) creating a declarative process specification (cf. Fig. 1a) and (2) generating optimized BP enactment plans (cf. Fig. 1b, c).

Creating declarative process specifications. In a first step, a declarative specification covering the control flow, the resource perspective, and the estimates of the BP to be supported is created. As stated, we use the constraint-based language ConDec (cf. Sect. 2) as basis. To plan and schedule the process activities, our proposal extends the ConDec specification by considering (1) estimations for the duration and the role of the required resource of the BP activities and (2) resource availabilities, resulting in a ConDec-R process model (cf. Def. 4). These estimates can be obtained by interviewing business experts or by analyzing past process executions.

Definition 4

A **ConDec-R process model** $CR = (Acts, C_{BP}, Res)$ related to a constraint-based process model $S = (A, C_{BP})$ (cf. Def. 1) is composed of a set of extended BP activities Acts, which contains tuples (a, role, dur) which includes for each BP activity $a \in A$ the role of the required resource (i.e., role) and the estimated duration (i.e., dur); a set of ConDec constraints C_{BP}; and a set of available resources Res which is composed of tuples (role, #role) which includes for each role (i.e., role) the number #role of available resources.

To develop the OptBPPlanner tool, we have extended Declare [13], which is a workflow management system that can be used to specify ConDec models.

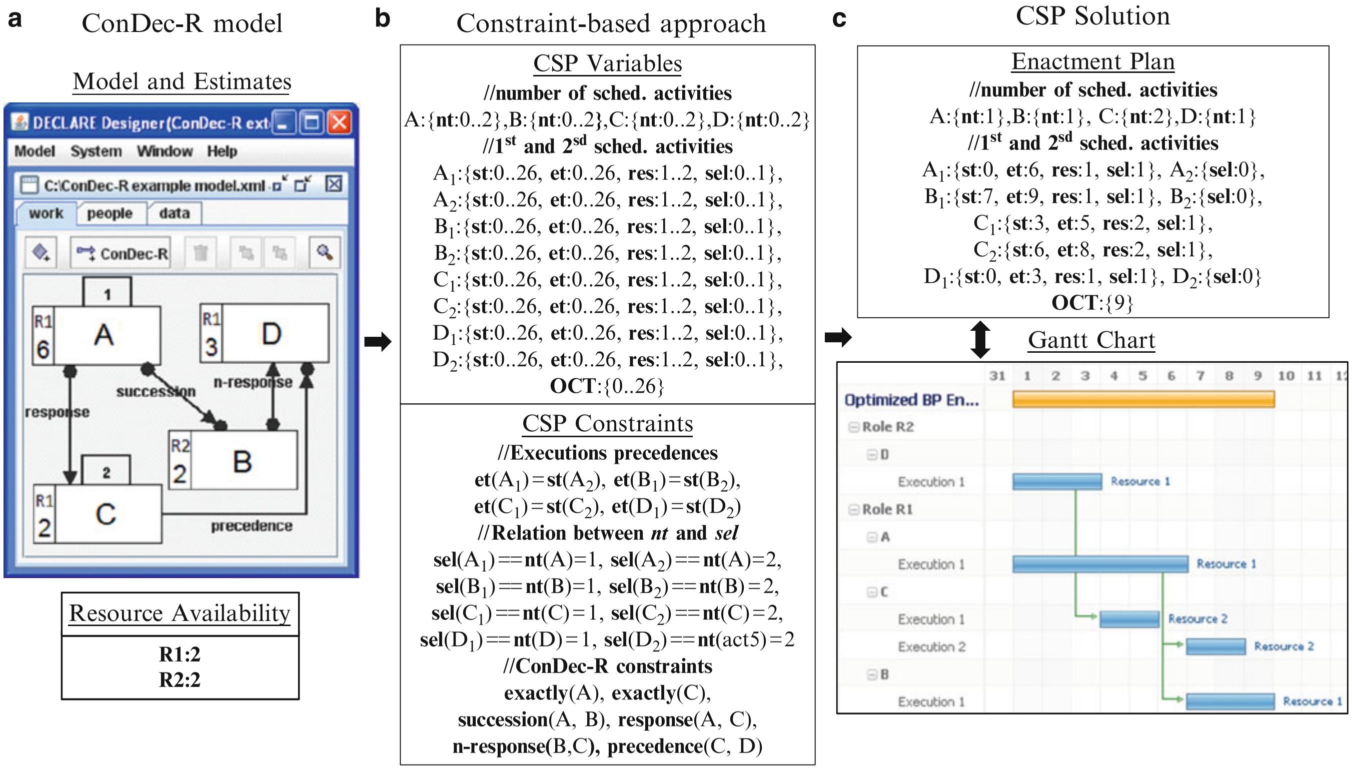

Fig. 1 Overview of our approach

The proposed extension allows the specification of ConDec-R models by including (1) duration and required resource for each BP activity and (2) number of available resources. Figure 1a shows a simple ConDec-R model (created using the aforementioned extension) (cf. Def. 4) where: Acts = {(A,R1,6), (B,R1,2), (C,R2,3), (D,R1,2)}; C_{BP} = {*exactly*(1,A), *exactly*(2,B), *succession*(A,B), *response*(A,B), *negate-response*(B,C), *precedence*(C,D)}; and *Res* = {(R1,2), (R2,2)}.

Generating of optimized BP enactment plans. In our proposal, optimized BP enactment plans are generated by applying a constraint-based approach for P&S the BP activities, taking the ConDec-R specification into account. In the proposed constraint-based approach, BP activities are modelled as repeated activities (cf. Def. 5), which are sequences of optional scheduling activities (cf. Def. 6). This is required since each execution of a BP activity is considered as one single activity which needs to be allocated to a specific resource and temporarily placed in the enactment plan, i.e., stating values for its start and end times.

Definition 5

A repeated activity ra = (r, dur, nt) is a BP activity which can be executed several times. It is composed by r, which represents the role of the required resource, the estimated duration dur, and a CSP variable nt which specifies the number of times the BP activity is executed.

For each repeated activity, nt_{MAX}[2] scheduling activities exist, which are added to the CSP problem specification, apart from including a variable nt.

Definition 6

A scheduling activity sa = (st, et, res, sel) represents a specific execution of a repeated activity, where st and et are CSP variables indicating the start and the end times of the activity execution, respectively, res is a CSP variable representing the resource used for the execution, and sel is a CSP variable indicating whether or not the scheduling activity is selected to be executed.

Moreover, to improve the modelling of the problems and to efficiently handle the constraints in the search for solutions, our constraint-based proposal includes a global constraint implemented through a filtering rule (cf. Sect. 2) for each ConDec-R template (cf. Fig. 1b). For a detailed description of these filtering rules, see [14]. In this way, the ConDec-R process model CR = (Acts, C_{BP}, Res) (cf. Def. 4, Fig. 1a) is translated into a COP Po = (V, D, C_{CSP}, OF) (cf. Def. 2, Fig. 1b) where:

1. $V = \{\mathrm{nt}(a) | a \in \mathrm{Acts}\} \cup \{\mathrm{st}(a_i), \mathrm{et}(a_i), \mathrm{sel}(a_i), \mathrm{res}(a_i) | i \in [1 \ldots \mathrm{nt}(a)], a \in \mathrm{Acts}\} \cup \mathrm{OC}$ T. OCT is a CSP variable which represents the overall completion time,[3] i.e., $\mathrm{OCT} = \max_{a \in \mathrm{Acts}}(\mathrm{et}(a_{\mathrm{nt}(a)}))$

[2] nt_{MAX} represents the maximum value of the initial domain of nt (cf. Fig. 1b).

[3] The overall completion time is the time needed to complete all process instances which were planned for a certain period.

2. D is composed of the domains of each CSP variable. The domain [0…2] is used for nt since 2 is the maximum cardinality for the BP activities (established by existence relations in the constraint-based model). The domain [0…26] is used for et and st since 26 would be the completion time if all the scheduling activities were serially executed taking the maximum cardinality for the BP activities into account.
3. C_{CSP} is composed of the global constraints related to C_{BP} together with the constraints which are inherent to the proposed model:

   - $\forall a \in \mathrm{Acts}, \forall i : 1 \leq i \leq \mathrm{nt}(a) : \mathrm{et}(a_i) \leq \mathrm{st}(a_{i+1})$ (i.e., a specific execution of a repeated activity precedes the next execution of the same activity).
   - $\forall a \in \mathrm{Acts}, \forall i : 1 \leq i \leq \mathrm{nt}(a) : \mathrm{sel}(a_i) = \mathrm{nt}(a_{i+1}) \geq i$ (i.e., the nt variable of the repeated activity is directly related to the sel variable of this associated scheduling activity).

4. OF = OCT.

For the current approach, to solve the constraint-based problems, the COMET system [15] is used, since it is able to generate high-quality solutions for highly constrained problems in an efficient way. This system provides a scheduling module that offers high-level constraint modelling and search abstraction, both specific to scheduling. The COP related to the ConDec-R specification is considered as a scheduling problem (cf. Fig. 1b) to take advantage of the efficient COMET mechanisms and high-level modelling. The optimized BP enactment plan is then created from the CSP solution (cf. Fig. 1c) and is composed of (1) the number of times each BP activity is executed, (2) the start and the completion times for each activity execution, and (3) the resource which is used for each activity execution. The generated enactment plans can be graphically represented by a Gantt chart [16] (cf. Fig. 1c)[4]. This chart illustrates the activity schedules and allows users to understand the solution at a glance. Moreover, the relations between executions of activities are depicted in the Gantt chart due to the ConDec-R constraints of the model (e.g., the relation between the first execution of D and the first execution of the C is due to the constraint *precedence*(C,D)).

Since the generation of optimal plans for these types of problems presents NP complexity [17], it is not possible to ensure the optimality of the generated plans for all cases. The developed constraint-based approach, however, allows solving the considered problems in an efficient way, as demonstrated in Sect. 4.

4 Empirical Evaluation

To evaluate the suitability of our proposal, a controlled experiment is conducted.

Purpose: The purpose of the empirical evaluation is to analyze our proposal in the generation of optimized BP enactment plans from ConDec-R models.

[4] The generated Gantt chart of Fig. 1c groups activities by roles, e.g., the *Execution1* of D is performed by the *Resource 1* of the *Role R2*. The rest of activities are performed by *Role R1*.

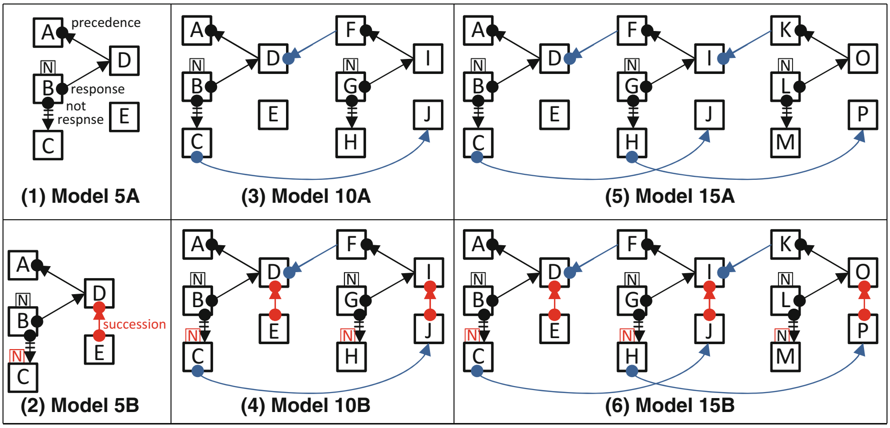

Fig. 2 Generic ConDec models

Objects: Different ConDec models are generated by considering correctness and representativeness. Consequently, we require the test models to be of medium size (i.e., including 5–15 BP activities which can be executed {15, 30, 60} times[5]) and comprise all three types of ConDec templates (cf. Sect. 2, cf. Fig. 2): (1) Model 5A includes 5 BP activities and few constraints, (2) Model 5B extends the Model 5A by including more constraints, (3) Model 10A includes 10 BP activities and few constraints, (4) Model 10B extends the Model 10A by including more constraints, (5) Model 15A includes 15 BP activities and few constraints, and (6) Model 15B extends the Model 15A by including more constraints. Moreover, in the case of *existence* constraints, a value for label *N* is established. In addition, different durations and required resources for each BP activity are considered, since these aspects have a great influence in the complexity of the search of optimal solutions. Specifically, 30 instances for each problem are randomly generated by varying activity durations between 1 and 40 and the role required between R1 and R2. In addition, two available resources for R1 and R2 are considered.

Furthermore, different time limits for the search algorithm are considered to show the applicability in scenarios with different response time requirements.[6]

Independent Variables: For the empirical evaluation, (1) M, i.e., the generic ConDec model, with the values {M5A, M5B, M10A, M10B, M15A, M15B}; (2) TL, i.e., the time limit (seconds) to find an optimal solution, with the values {5, 50, 300}; and (3) *N*, i.e., the value for the label *N* of the *existence* constraints in the models, with the values {15, 30, 60} are considered.

[5] These values are considered to analyze the behavior of our proposal when dealing with problems of different size, i.e., with different number of repetitions of certain activities.

[6] The set of problems which are used are available at http://regula.lsi.us.es/ISD12/EV.zip.

Table 1 Average values related to the experimental executions

Indep. var.			Resp. var. (TL = 5)			Resp. var. (TL = 50)			Resp. var. (TL = 300)		
M	*N*	Nact	OCT	%Opt	Topt	OCT	%Opt	Topt	OCT	%Opt	Topt
M5A	15	17	334.8	100.0	0.28	334.8	100.0	0.28	334.8	100.0	0.28
M5A	30	32	728.1	83.3	1.70	728.1	100.0	2.77	728.1	100.0	2.77
M5A	60	62	1,478.7	93.3	0.69	1,478.7	93.3	20.00	1,478.7	93.3	20.00
M5B	15	33	600.6	100.0	0.08	600.6	100.0	0.08	600.6	100.0	0.08
M5B	30	63	1,477.0	100.0	0.29	1,477	100.0	0.29	1,477.0	100.0	0.29
M5B	60	123	2,626.0	100.0	2.30	2,626	100.0	2.30	2,626.0	100.0	2.30
M10A	15	34	471.4	40.0	0.41	471.4	43.3	2.38	471.4	50.0	14.71
M10A	30	64	894.4	50.0	1.28	894.4	50.0	1.28	894.4	50.0	1.28
M10A	60	124	1,698.7	6.7	3.47	1,698.7	36.7	16.53	1,698.7	43.3	35.34
M10B	15	66	789.3	33.3	0.15	788.8	33.3	0.15	788.8	36.7	8.48
M10B	30	126	1,495.0	40.0	0.70	1,494.2	43.3	2.52	1,494.2	43.3	2.52
M10B	60	246	2,896.5	16.7	3.83	2,895.1	16.7	3.83	2,894.2	16.7	3.83
M15A	15	52	563.6	13.3	0.28	563.6	13.3	0.28	563.2	13.3	0.28
M15A	30	97	1,027.2	16.7	2.08	1,027.2	16.7	2.08	1,027.2	20.0	32.29
M15A	60	187	2,016.0	6.7	4.43	2,016.0	23.3	16.51	2,016.0	23.3	16.51
M15B	15	99	1,071.4	13.3	0.95	1,070.9	13.3	0.95	1,070.6	20.0	45.55
M15B	30	189	1,879.7	40.0	1.70	1,875.8	43.3	4.25	1,875.8	43.3	4.25
M15B	60	369	3,972.0	0	–	3,934.5	20.0	11.19	3,923.7	23.3	36.94

Response Variables: The suitability of our approach is tested regarding (1) the average value of the objective function which is obtained (i.e., OCT), (2) the average percentage of optimal solutions which are found (i.e., %Opt), and (3) the average time for getting the optimal solution, considering the cases in which an optimal solution is found (i.e., Topt).

Experimental Design: 540 instances are generated by considering different values for *M* (6 values), *N* (3 values), and the random generation of durations and required resources (30 problem instances). For each instance a complete search (cf. Sect. 2) is executed to optimize the OCT considering the three different values of TL. The response variables are then calculated by considering the average values of the 30 problem instances.

Experimental Execution: The constraint-based search algorithm is run on an Intel(R) Xeon(R) CPU E5530, 2.4 GHz, 8 GB memory, running Debian 6.0.3. The system COMET [15] is used to solve the developed constraint-based problems.

Experimental Result and Data Analysis: Table 1 shows for each problem (specified by *M*, *N*, and TL), the average values of the response variables (i.e., OCT, %Opt, and Topt) for the 30 problem instances. In addition, the number of scheduling activities (cf. Def. 6) which are executed (i.e., Nact) are shown for each pair $\langle M,N \rangle$.

As expected, %Opt decreases as the number of BP activities and/or scheduling activities increases (i.e., complexity of the problem). As TL increases %Opt increases and OCT decreases but not as significantly since the best values of OCT are achieved in a short time but it was not possible to ensure their optimality. Moreover, Topt increases as the complexity of the problems increases. In general, experimental results show that despite NP complexity of the considered problems, the values for the percentage of optimal solutions found and for the average time for getting optimums are quite good for small- and medium-sized problems (between 17 and 189 scheduling activities[7]). Hence, the approach becomes suitable for run-time applications (e.g., recommendations) and for scenarios where high quality is required.

5 Discussion

One advantage of our proposal is that the optimized BP enactment plans are generated by P&S all BP activities, allowing for a global optimization of the objective function. Moreover, the generation of the optimized plans is carried out through a constraint-based approach, which is suitable for modelling and solving P&S problems [12]. In addition, this approach allows modelling the considered problems in an easy way, since the considered specifications are based on high-level constraints. Furthermore, BPs are specified in a declarative way, which facilitates the human work involved and avoids failures [18]. Moreover, our approach, as extension of other similar works, considers the resource perspective besides the control-flow perspective; hence greater optimization can be obtained.

On the other hand, the proposed approach presents some drawbacks. First, the most important limitations are the assumptions that are made, i.e., the optimized plans are generated by considering estimated values for activity durations and resource availabilities; hence our proposal is only appropriate for processes for which the duration of the activities and resource availabilities can be estimated. However, to consider deviations in the estimates, the optimized plans can be updated—if necessary—through replanning, allowing to react to changes in a quick and flexible way. Secondly, the business analysts must deal with a not standard language for the specification of BPs, therefore a period of training is required. Moreover, the considered declarative specifications deal with both control-flow and resource perspectives, but do not consider the data perspective. It is intended to consider this aspect in future works. In our proposal we focus on the minimization of overall completion time. However, it can be easily extended to consider further objectives, such as cost.

5.1 Applications of Optimized BP Enactment Plans

Simulation: Simulation of BPs can be effectively used for analyzing processes and for improving BP models. BP simulation presents a "fast-forward" view on a

[7] Note that getting the optimum for scheduling problems of 189 activities can entail a great complexity. In fact, there are many scheduling benchmarks of smaller size for which their optimal values are not even known.

current BP, so that the generated simulation models can accurately reflect the real-world process of interest. One interesting application of BP simulation is to identify unbalances between the resources required for executing a particular process and the available resources [19]. Moreover, the effects of alternative resource schedules can be investigated. Our proposal can be used in *what-if* scenarios to evaluate the impact of changing something in the declarative BP. For example, in Fig. 1a, if we reduce the resource availability of B to 1, the generated plan is still valid (i.e., we can do the same considering fewer resources); however, if we reduce R1 to 1, A, B, and C cannot be executed in parallel, then the enactment plan changes. Thus, the results can be studied to analyze and to enhance the current BP model (process design and analysis phase of BPM life cycle [1]).

Time Prediction: There are many scenarios where it is useful to have reliable time predictions [5]. In the current approach, the generated optimized BP enactment plans can be used for this purpose since time information is available based on the estimated durations of the activities. For a given process instance state, the expected completion time for the instance and activities can be calculated by taking the end time of the remaining activities of the optimized plan into account. In this way, the BP enactment plans of the BP model can be used for predicting the completion time of running instances and activities, and hence improving the process design and analysis phase of BPM life cycle [1].

Recommendations: The application of BP enactment plan for generating recommendations is detailed in a previous work [6]. In the current dynamic business world, the economic success of an enterprise increasingly depends on its ability to react to changes in its environment in a quick and flexible way [20]. Therefore, flexible BPM systems are required to allow companies to rapidly adjust their BPs to changes in the environment [21]. In general, increasing flexibility in BPM systems tends to result in decreased user support [22] requiring more experienced users. Typically, given a certain partial trace users can choose from several enabled activities (i.e., activities whose execution does not violate any constraint or only lead to temporary violations [23]) which activity to execute next. This selection, however, can be quite challenging since objective functions of the process should be considered, and users often do not have an understanding of the overall process. Moreover, optimization of objective functions requires that resource capacities are considered. Therefore, recommendation support is needed during BP execution, especially for inexperienced users. As an application of the generated optimized BP enactment plans, recommendations which assists users during process enactment to optimize objective functions of the processes can be generated, hence enhancing the process enactment phase of BPM life cycle [1].

The Generation of Optimized BP Models: BP models are usually defined manually by business analysts through imperative languages. To this end, the analysts must deal with several aspects, such as resource allocation, activity properties, the relations between them, and, in most cases, even the optimization of several objectives. Therefore, the manual specification of BP models can form a very complex problem, can consume a great quantity of time and human resources, may cause some failures, and may lead to non-optimized models. To overcome these problems, taking the information of the optimized BP enactment plans and the constraints of the declarative BP model into account, BP models can be generated, therefore improving process design and analysis phase of BPM life cycle [1].

6 Related Work

There exist some proposals which could be used to generate optimized enactment plans for BPs from constraint-based specifications. Specifically [24] proposes the generation of an automaton from constraint-based specifications based on linear temporal logic (LTL) which represents exactly all traces that satisfy the LTL formulas. When extending this approach by including estimates, the overall completion time of all the traces could then be calculated (e.g., [5]). However, the big disadvantage following such an approach would be that it comes to a state explosion since all the LTL formulas have to be concatenated to build a big automaton [24], and, unlike the proposed approach, no heuristic has been used. In a similar way, CLIMB [25] could be used to generate quality traces from declarative specifications and calculate its completion time. Then, the best traces could be selected. Unlike the proposed approach [25], does neither consider optimality nor resource availabilities. Therefore, this would only cover the planning part of the current proposal, but not the scheduling aspects addressed by our approach.

There are additionally some proposals related to generating imperative BP models [7, 8, 26], giving recommendations [27], simulation, [28] and time prediction [5]. However, unlike in our proposal, the process optimization and the resource allocation are not considered.

7 Conclusions and Future Work

As a major contribution of this work, we propose a method for the automatic generation of optimized BP enactment plans from declarative specifications which assists users during different stages of the BPM life cycle, i.e., BP design and analysis and enactment stages, to optimize objective functions of the processes (i.e., minimization of overall completion time). The proposed method is based on a constraint-based approach for planning and scheduling the BP activities and considers both the control flow and the resource perspective. To demonstrate the feasibility of our approach, a tool, called OptBPPlanner, has been implemented. As for future work, it is intended to extend the proposed approach by considering further objective functions. Moreover, we will explore various constraint-based solving techniques and analyze their suitability for the generation of optimized plans.

Acknowledgments This work has been partially funded by the Spanish Ministerio de Ciencia e Innovación (TIN2009-13714) and the European Regional Development Fund (ERDF/FEDER).

References

1. Weske M (2007) Business process management: concepts, methods, technology. Springer, Berlin
2. van der Aalst W, ter Hofstede A, Weske M (2003) Business process management: a survey. Proc. business process management: International Conference, BPM 2003, pp 1–12
3. Group G (2010) Leading in times of transition: the 2010 CIO agenda (EXP Premier Report No. January 2010), in Report, Gartner, Inc.
4. Rozinat A, Wynn M, van der Aalst W, ter Hofstede A, Fidge C (2009) Workflow simulation for operational decision support. Data Knowl Eng 68(9):834–850
5. van der Aalst W, Schonenberg M, Song M (2011) Time prediction based on process mining. Inf Syst 36(2):450–475
6. Barba I, Weber B, Del Valle C (2011) Supporting the optimized execution of business processes through recommendations. In Proc. BPM Workshops, pp 135–140
7. González-Ferrer A, Fernández-Olivares J, Castillo L (2009) JABBAH: a java application framework for the translation between business process models and HTN. In: Proc. ICKEPS
8. R-Moreno M, Borrajo D, Cesta A, Oddi A (2007) Integrating planning and scheduling in workflow domains. Expert Syst Appl 33(2):389–406
9. Ghallab M, Nau D, Traverso P (2004) Automated planning: theory and practice. Morgan Kaufmann, Amsterdam
10. Brucker P, Knust S (2006) Complex scheduling (GOR-Publications). Springer-Verlag New York, Inc., Secaucus, NJ
11. Pesic M, van der Aalst W (2006) A declarative approach for flexible business processes management. In: Proc. BPM Workshops, pp 169–180
12. Salido M (2010) Introduction to planning, scheduling and constraint satisfaction. J Intell Manuf 21(1):1–4
13. Declare (2011) Declarative approach to workflow management systems. http://www.win.tue.nl/declare/. Online; Accessed 4 May 2012
14. Barba I, Del Valle C (2011) Filtering rules for ConDec templates—pseudocode and complexity. http://regula.lsi.us.es/ISD12/FilteringRulesforConDecTemplates.pdf
15. Dynadec. Comet. http://dynadec.com/support/downloads/. Online; Accessed 4 May 2012
16. Gantt H (1913) Work, wages, and profits. Engineering Magazine Co., New York
17. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman & Co., New York, NY
18. Rychkova I, Regev G, Wegmann A (2008) High-level design and analysis of business processes: the advantages of declarative specifications. In: Proc. RCIS, pp 99–110
19. Reijers H, van der Aalst W (1999) Short-term simulation: bridging the gap between operational control and strategic decision making. In: Proc. IASTED conference on modeling and simulation, pp 417–421
20. Poppendieck M, Poppendieck T (2006) Implementing lean software development: from concept to cash. Addison-Wesley, Boston, MA
21. van der Aalst W, Jablonski S (2000) Dealing with workflow change: identification of issues and solutions. Int J Comput Syst Sci Eng 15(5):267–276
22. van Dongen B, van der Aalst W (2005) A meta model for process mining data. In: Proc. Caise, pp 309–320
23. Pesic M, Schonenberg M, Sidorova N, van der Aalst W (2007) Constraint-based workflow models: change made easy. In: OTM Conferences (1), pp 77–94

24. Pesic M (2008) Constraint-based workflow management systems: shifting control to users. Ph.D. thesis, Eindhoven University of Technology, Eindhoven
25. Montali M (2009) Specification and verification of declarative open interaction models: a logic-based approach. Ph.D. thesis, Department of Electronics, Computer Science and Telecommunications Engineering, University of Bologna
26. Hoffmann J, Weber I, Kraft F (2010) SAP speaks PDDL. In: Proc. AAAI, pp. 1096–1101
27. Schonenberg H, Weber B, van Dongen B, van der Aalst W (2008) Supporting flexible processes through recommendations based on history. In: Proc. BPM, pp 51–66
28. Reijers H (2003) Design and control of workflow processes: business process management for the service industry, vol 2617, LNCS. Springer, Berlin