

Generating optimized configurable business process models in scenarios subject to uncertainty

Andrés Jiménez-Ramírez ^{a,*}, Barbara Weber ^b, Irene Barba ^a, Carmelo Del Valle ^a

^a University of Seville, Dpto. Lenguajes y Sistemas Informáticos, Spain

^b University of Innsbruck, Department of Computer Science, Austria

Keywords:

Configurable business process models
Constraint-based business process models
Flexibility
Robustness
Planning and scheduling
Constraint programming

A B S T R A C T

Context: The quality of business process models (i.e., software artifacts that capture the relations between the organizational units of a business) is essential for enhancing the management of business processes. However, such modeling is typically carried out manually. This is already challenging and time consuming when (1) input uncertainty exists, (2) activities are related, and (3) resource allocation has to be considered. When including optimization requirements regarding flexibility and robustness it becomes even more complicated potentially resulting into non-optimized models, errors, and lack of flexibility.

Objective: To facilitate the human work and to improve the resulting models in scenarios subject to uncertainty, we propose a software-supported approach for automatically creating configurable business process models from declarative specifications considering all the aforementioned requirements.

Method: First, the scenario is modeled through a declarative language which allows the analysts to specify its variability and uncertainty. Thereafter, a set of optimized enactment plans (each one representing a potential execution alternative) are generated from such a model considering the input uncertainty. Finally, to deal with this uncertainty during run-time, a flexible configurable business process model is created from these plans.

Results: To validate the proposed approach, we conduct a case study based on a real business which is subject to uncertainty. Results indicate that our approach improves the actual performance of the business and that the generated models support most of the uncertainty inherent to the business.

Conclusions: The proposed approach automatically selects the best part of the variability of a declarative specification. Unlike existing approaches, our approach considers input uncertainty, the optimization of multiple objective functions, as well as the resource and the control-flow perspectives. However, our approach also presents a few limitations: (1) it is focused on the control-flow and the data perspective is only partially addressed and (2) model attributes need to be estimated.

1. Introduction

A Business Process (BP) can be defined as a set of activities which are performed in coordination in an organization to achieve a business goal [93]. These activities can be manual activities, other BPs, or even pieces of software. Nowadays, in order to support BPs, BP Management (BPM) embraces methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, and other sources

of information [90]. Such management generally follows a strict methodology to ensure the quality of the information systems which are created. Typically, the traditional BPM life cycle [93] includes four phases, i.e., process design & analysis (i.e., a design of the BP is created following the requirements), system configuration (i.e., the software defined in the BP design is implemented), process enactment (i.e., the software is executed following the BP design) and evaluation (i.e., monitoring information or logs are analyzed to look for design improvements) [93].

The quality of a BP design has a great influence on all the phases of the BPM life cycle and it is essential for BP improvement, which has been ranked as the number one priority for top management by the 2010 Gartner survey [36].

* Corresponding author. Tel.: +34 606772495.

E-mail addresses: ajramirez@us.es (A. Jiménez-Ramírez), barbara.weber@uibk.ac.at (B. Weber), irenebr@us.es (I. Barba), carmelo@us.es (C. Del Valle).

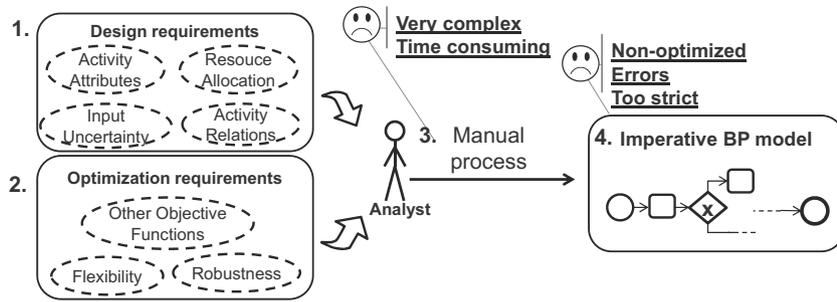


Fig. 1. Motivation overview.

1.1. Problem statement

In the process design & analysis phase the BP models are typically specified by hand using imperative languages like EPC or BPMN [12]. This way, a precise activity sequence which establishes how a given set of activities has to be performed is defined. Such a sequence typically includes temporal relations between activities or even dependencies with input data. Typically, such activities are related to a set of attributes (e.g., duration and cost) which need to be estimated. Furthermore, in many real scenarios such estimates might be subject to input uncertainty (e.g., non-punctual clients imply uncertainty in their arrival times) [84]. In addition, the BP performance can be greatly influenced by how the different resources are assigned to each activity that has to be performed [45,81]. Therefore, regarding such scenarios and motivated by the case study described in the paper, when designing a BP model, analysts have to face certain design requirements (cf. Fig. 1 (1)), such as:

1. Dealing with activity attributes and their estimated values.
2. Managing the input uncertainty which exists in many real scenarios [84] in which providing a range of possible values for a BP property is most reliable than providing an exact value which may be difficult to know. For example, the arrival time of clients can be considered uncertain due to unpunctual clients.
3. Dealing with relations between the activities, i.e., control-flow as well as temporal and data constraints of the BP.¹
4. Considering resource allocation.

Since uncertain scenarios are considered, managing such input uncertainty becomes necessary. For this, flexibility and robustness are proposed since we consider this is the best way to properly address the considered uncertainty. The situation is further complicated if the aforementioned design requirements have to be addressed along with optimizing some (potentially) conflicting objective functions (e.g., time and cost). Such optimization requirements (cf. Fig. 1 (2)) can be summarized as follows:

1. Flexibility, i.e., the capability to adapt to input uncertainty [34,66,78]. For this, designed models should consider different execution alternatives to support such uncertain scenarios [93].
2. Robustness, i.e., the capability to withstand the uncertainty to some extent [18,20,27]. For this, BP models should be designed to avoid making unnecessary adaptations which typically are costly.
3. Other objective functions are commonly considered since the BP design usually involves a trade-off between different

quality dimensions which may be in conflict or be opposed [67].

This task of creating a BP design can form a very complex problem and be very time consuming (cf. Fig. 1 (3)). For this, methods and tools for supporting analysts during the BP design are becoming more and more important. Moreover, the resulting models may be non-optimized, potentially contain errors, and might be too strict [29,58,94] (cf. Fig. 1 (4)).

In such context, there exist some proposals for generating BP models or that could be extended in such direction (cf. [29,40,47,53,60,63,64,75,94]). These proposals are based on generating a single execution plan which fulfills all the BP constraints starting with a constraint-based specification. This plan could be, in turn, used for the generation of an imperative model. As a major drawback of existing proposals, considering only one single execution plan unnecessarily restricts the flexibility of the resulting imperative model. Thus, to the best of our knowledge, the existing proposals are not sufficient to address all the previously mentioned requirements, e.g., dealing with the flexibility needs of existing BPs [66].

1.2. Contribution

In order to facilitate the human work which is involved in the process design & analysis phase and to improve the resulting imperative BP models we propose a method for automatically creating configurable BP models (i.e., a modeling artifact that captures a family of process models in an integrated manner) [22] from declarative specifications [29] (cf. Fig. 2). The proposed approach considers all the aforementioned requirements which have to be considered when creating a suitable BP model, i.e., activity attributes, resource allocation, input uncertainty, relation between activities, optimization of several objective functions, as well as flexibility and robustness issues, and is detailed in the following.

Declarative models are typically easier to specify and less time-consuming than imperative models in scenarios where high variability is required [29]. Therefore, we propose to use a declarative specification as starting point of the proposed approach. For this, the Declare language² [63] is used as basis, 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. We extend Declare in order to widen its design flexibility by considering stochastic values for modeling the uncertainty of the scenario (as required in the considered problems, cf. Section 1.1), resulting in the SDeclare language. To be more precise, with the proposed extension, some properties of a BP (such as activity attributes, data and temporal constraints, and resource availability) can be expressed

¹ Note that the considered scenarios are focused on the control-flow and the resource perspectives of the BPs and the data perspective is only partially considered.

² Declare is one of the most referenced and used declarative BP languages in the context of BPM.

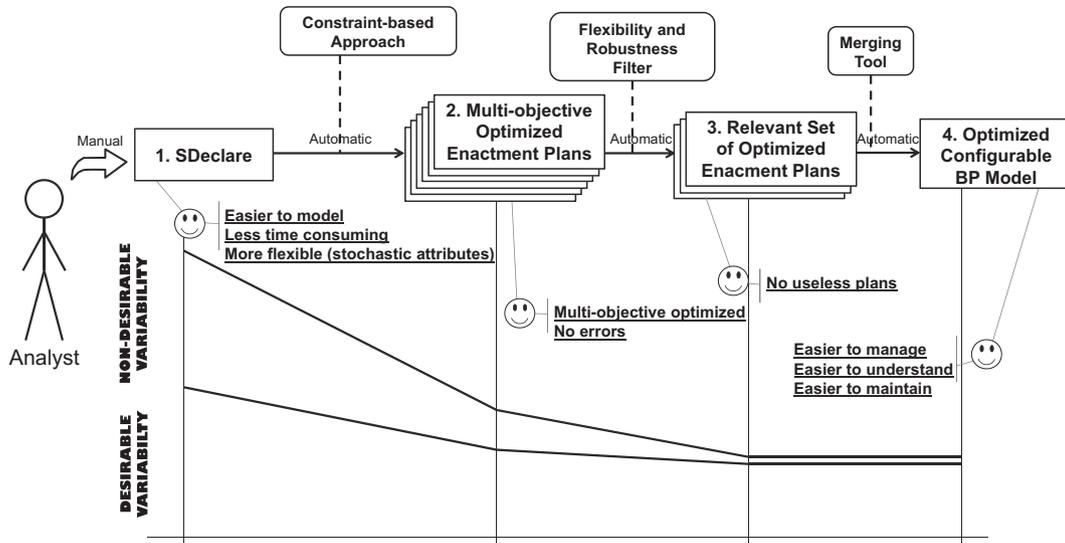


Fig. 2. Contribution overview.

through probabilistic mass functions instead of with fixed values. For example, our approach allows one to specify the uncertainty about the duration of an activity by using a flat discrete range (e.g., [15–20], meaning that such an activity may last 15, 16, 17, 18, 19 or 20 units of time with the same probability). This can be used, for example, for specifying that the arrival time of clients is uncertain due to unpunctual clients, or that the availability of some resources is subject to uncertainty. The SDeclare language is then used for the declarative specification of the BP models (cf. Fig. 2 (1)).

Since a declarative model captures highly variable scenarios, it may include many execution alternatives that are not desirable for the business regarding the optimization of a set of objective functions. For this, the current approach is based on extracting a desirable part of the variability of a declarative model through the generation of multi-objective optimized enactment plans (cf. Fig. 2 (2)), while discarding bad execution alternatives. This process is done automatically using a constraint-based approach which obtains the best execution alternatives of a declarative model according to a set of given objective functions, i.e., any quantitative property that can be measured related to an enactment plan like completion time, cost, etc. For this, activities to be executed have to be selected and ordered (planning problem [33]) considering both control-flow constraints as well as resource and data constraints imposed by the declarative specification (scheduling problem [14]).

Since the generated set of multi-objective optimized enactment plans may contain similar alternatives or non-robust alternatives, such set must be filtered, as proposed in our approach. That filtering step is performed regarding how the set of plans manages the input uncertainty in such a way that flexibility and robustness are optimized. Specifically, those alternatives which are too strict (i.e., not robust [18,20,27]) or which only withstand an extent of the uncertainty which is already withstood by another alternative (i.e., do not contribute to the flexibility of the final solution [34,78,66]), are discarded.³ Therefore, the variability of the source declarative model is reduced to a set of relevant plans (cf. Fig. 2

(3)) where most of the non-desirable alternatives are removed. In this way, the proposed approach manages both flexibility and robustness at design-time,⁴ as motivated in Section 1.1.

Typically the enactment plans which are kept after such filtering process share many commonalities since they are created from the same declarative specification and optimize the same objective functions. Therefore, a configurable BP model [22,35,39,49,71,88] can be created by merging all these plans (cf. Fig. 2 (4)). The goal of creating configurable BP models in the proposed approach is twofold: (1) supporting the analysts in the management of the set of optimized plans, and (2) helping the analysts understand what the different plans share, what their differences are, and why and how these differences occur [71].

In previous related work (cf. [44]) we presented an approach for generating optimized enactment plans from constraint-based specifications. However, this paper significantly extends this previous work by: (1) extending the considered constraint-based language (i.e., SDeclare) by allowing the specification of stochastic attributes, (2) efficiently improving the constraint-based algorithm to obtain optimized enactment plans regarding several objective functions, (3) generating configurable BP models from those optimized enactment plans regarding flexibility and robustness concerns, and (4) applying the proposed approach to a real scenario, and using such scenario to validate the proposed approach in practical settings through the analysis of different performance measures.

The remainder of this paper is structured as follows: Section 3 introduces backgrounds on related areas, Section 4 shows how a configurable BP model is extracted from a SDeclare model, Section 5 explains a real example, Section 6 deals with the evaluation, Section 7 presents a critical discussion, Section 2 summarizes related work, and Section 8 includes some conclusions and future work.

2. Related work

The Declare language [63] has been extended in several works [60,61,94]. In fact, SDeclare is based on the time extension defined

³ Note that robustness could be considered as an additional objective function to be optimized when generating the optimized enactment plans. However, since both flexibility and robustness help to deal with the considered uncertainty, we separate them from the remaining objective functions to be optimized with the goal of considering both together in the same step. In fact, when solving the considered problems, while the other objective functions are considered when generating the set of optimized enactment plans, flexibility and robustness are considered in a second step to filter such set.

⁴ Note that flexibility can be managed by: (1) design, i.e., at design-time some control-flow patterns which allows one to consider different alternatives (e.g., OR structures) are included in the model or (2) flexible PAISs, i.e., at run-time several activities of a flexible BP model (e.g., a declarative model) are enabled to be executed [66]. Since this approach is focused on the process design & analysis phase, flexibility at run-time is out of the scope of this paper.

in [60] where it is possible to define time lags over the different Declare constraints. The same time-aware extension is considered in [94] where, additionally, a deep reasoning based on a finite automaton is performed to warn the users to avoid wrong states. Furthermore, a data-aware extension has been recently proposed in [61]. Such extension is considered in the current approach. Nevertheless, unlike the current approach, Montali et al. [61] is based on Event Calculus and it is focused on monitoring and operational support. As a major contribution of SDeclare regarding existing proposals [60,61,94], it allows specifying the input uncertainty of real scenarios by using stochastic values.

We are not aware of any other approaches for generating configurable BP models—or set of enactment plans—from declarative specifications. However, there exist some further proposals which could be extended in such direction [40,47,53,60,63,74]. Specifically, Pesic [63] proposes the generation of a non-deterministic finite state automaton from constraint-based specifications which represents exactly all traces that satisfy the constraints. However, the big disadvantage following such an approach would be that the process of generating the automaton from the declarative specifications is exponential with respect to the size of the formula [62], and, unlike the proposed approach, no heuristic is used. Additionally, CLIMB [60] could be used to generate quality traces from declarative specifications, and calculate its values for different objective functions. Unlike the proposed approach, Montali [60] does neither consider optimality nor resource availabilities. Therefore, these would only cover the planning part of the current proposal, but not the scheduling aspects. In a related way, the work [47] plans and schedules tasks considering resources and the optimization of one objective function through an integer constraint-based specification. Although [47] presents a similar constraint-based approach, it misses dealing with multi-objective optimization, and does not support high level constraints. Moreover, in [53], a constraint formalization is proposed to generate variations of an ad-hoc BPMN sub-processes. In a similar way, Rychkova et al. [74] proposes the specification of processes based on a first-order logic language and translates them to an imperative language. In turn, related to BP, Hummer et al. [40] provides a model-driven approach which produces an imperative process specification from a declarative specification. Unlike our approach, Hummer et al. [40], Lu et al. [53] and Rychkova et al. [74] do not consider the optimization of any objective function. Furthermore, none of this approaches [40,47,53,60,63,74] considers the uncertainty of the scenario through stochastic attributes.

As mentioned, several approaches exist for dealing with flexibility issues [3,18], even in the context of BP [66,78]. However, in this paper as a novel contribution, we provide quantitative definitions for both robustness and flexibility which allow us to measure how the uncertainty of a real scenario is supported by an enactment plan and by a configurable BP model respectively.

In literature, different approaches deal with the variability of BPs [39,48,69,71,77,80]. In PESOA [77] and C-EPC [69,71] configurable BP models are used as basis. Variation points are either defined by a set of annotations of activities [77] or by including configurable BP nodes [69,71]. In turn, the work [69] improves the C-EPC language by incorporating both resource and data notation in the C-EPC models. Unlike C-EPC and PESOA, Provop [39] and RULE [48] consider a base process model which is configured using a set of predefined change operations [39] or applying some business rules [48]. Furthermore, there are other approaches which mix declarative specifications with configurable BP models, e.g., ConfDeclare [80] presents a declarative language in which activities can be hidden and constraints can be omitted. Regarding how configurable BP models are created, essentially, there are two ways: manually and automatically. On the one hand, the manual creation of configurable BP models can be carried out from

scratch by manually specifying the variation points [22,35,39,48,71,77]. The main problems of the manual creation is that it is typically a very time consuming task and requires deep skills on the modeling language. On the other hand, an automatic method has been proposed to generate a configurable BP model, in C-EPC language, from a set of BP models by analyzing the similarities of the source BP models and including variation points where they differ [68,70]. The main problem of the automatic creation of configurable BP models is that it requires a family of BP models independently specified. Since we propose to automatically create configurable BP models, the current approach builds upon these techniques.

3. Background

Using a constraint-based BP model as starting point, this approach applies constraint-based planning and scheduling techniques considering multiples objectives functions in order to generate a configurable BP model. When generating the configurable BP models, flexibility and robustness concerns are considered.

3.1. Constraint-based BP models

Different paradigms for process modeling exist, e.g., imperative and declarative. Irrespective of the chosen paradigm, desired behavior must be supported by the process model, while forbidden behavior must be prohibited [60,63]. While executable process models specify exactly how things have to be done,⁵ declarative models only focus on what should be done. In literature, several rule-based and constraint-based languages for declarative BP modeling are proposed (e.g., [23,54,63,64,92]). In our proposal we use the constraint-based language Declare (also known as ConDec) [63,64] for the BP control-flow specification. We consider Declare to be a suitable language, since it allows the specification of BP activities together with the goal to be achieved (cf. Definition 1).

Definition 1. The **goal** of a BP is specified through the constraints which must be satisfied during the BP enactment.

Moreover, Declare allows to specify a wide set of BP models in a simple and flexible way. In addition, Declare has been widely referenced in the past years in the context of BPs [17,51,55,60]. Declare is based on constraint-based BP models (cf. Definition 2).

Definition 2. A **constraint-based BP model** $CM = (A, C_{BP})$ consists of a set of activities A , and a set of constraints C_{BP} prohibiting undesired execution behavior. Each activity $a \in A$ can be executed arbitrarily often if not restricted by any constraints.

Such definition is provided to formalize the concepts which already exist in the literature related to constraint-based BP model.

Constraints can be added to a Declare model to specify forbidden behavior, restricting the desired behavior. For this, Declare proposes an open set of templates which can be divided into 4 groups:

1. **Existence** templates: unary relationships 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 impose the presence of a certain activity when some other activity is performed, e.g., Precedence(A,B) specifies that to execute activity B, activity A needs to be executed before.

⁵ With the term executable models we refer to imperative models which are rather strict and which represent only one enactment plan (or at most only few decision points are included).

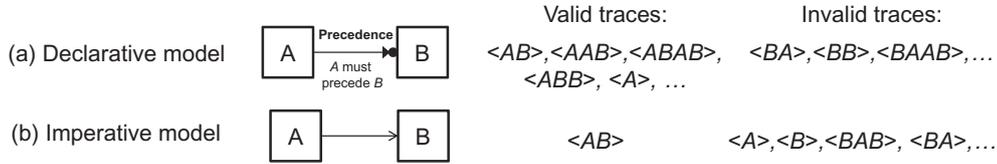


Fig. 3. Increased flexibility of declarative models versus executable models.

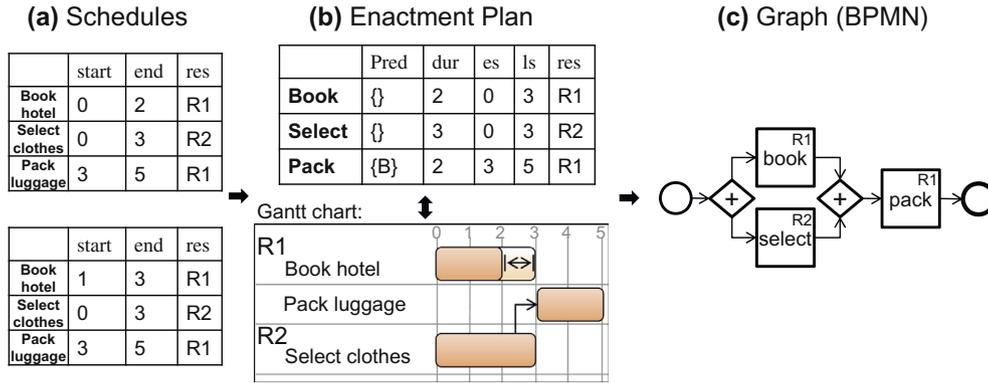


Fig. 4. Relations between schedules, enactment plans, Gantt charts and graphs.

3. **Negation** templates: negative relationships used to forbid the execution of activities in specific situations, e.g., $\text{NotCoexistence}(A,B)$ specifies that if B is executed, then A cannot be executed, and vice versa.
4. **Choice** templates: n-ary relationships expressing the need of executing activities belonging to a set of possible choices, e.g., $\text{ExactlyChoice}(N,\{A,B,C\})$ specifies that exactly N activities of the set $\{A,B,C\}$ must be executed.

Due to their flexible nature, there are different ways to execute a constraint-based BP model in such a way that all constraints are fulfilled, i.e., the process goal is reached (cf. Definition 1 and Example 1).

Example 1. Fig. 3(a) shows a constraint-based BP model where traces⁶ $\langle AAB \rangle$, $\langle AB \rangle$, $\langle ABAB \rangle$, $\langle ABB \rangle$, $\langle A \rangle$ are some of the valid ways of executing such model, while traces $\langle BA \rangle$, $\langle BB \rangle$, $\langle BAAB \rangle$ are invalid since A must precede B. In contrast, Fig. 3(b) shows an executable model where there is only one valid execution trace, $\langle AB \rangle$.

The different valid execution alternatives related to a specific constraint-based BP model, however, can greatly vary in respect to their quality, i.e., how well different performance objective functions (cf. Definition 3) can be achieved.

Definition 3. The **objective function** of a BP is the function to be optimized during the BP enactment, e.g., minimization of overall completion time.

Many real scenarios require the optimization of multiple objective functions. Thus, we suggest the automatic generation of a multi-objective optimized configurable BP model from a constraint-based BP model by applying constraint programming for Planning and Scheduling (P&S) the BP activities.

3.2. Planning, scheduling and constraint programming

The area of scheduling includes problems in which it is necessary to determine a schedule for a set of activities related by temporal and resource constraints (in our context the control-flow constraints). A schedule states (1) the start and end times of the activities to be executed and (2) the resource which is assigned to perform each activity. Since different activities may require the same resources, they may compete for limited resources (i.e., resource constraints). In scheduling problems several objective functions are usually considered to be optimized, in most cases related to temporal measures, or considering the optimal use of resources.

In a wider perspective, in Artificial Intelligence (AI) planning [33], 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 general, the objective in planning consists of, given a set of available activities, generating a schedule by selecting and ordering a set of activities in a way that the resulting plan reaches a given goal. Furthermore, in planning problems, usually the optimization of certain objective functions is considered.

In such context temporal analysis is typically applied over the resulting schedules to figure out the temporal slack of the activities [14,37], i.e., to calculate which activities can delay or advance their execution without affecting the completion time of the schedule. In this respect, different techniques such as CPM, PERT or Gantt charts [31] can be used to perform this analysis in order to calculate the enactment plan (cf. Definition 4) related to a specific schedule. The same enactment plan can be related to different schedules, as shown in Fig. 4.

Definition 4. An **enactment plan** $P = (pid, Acts)$ is identified by pid and is composed of a set of activities $act \in Acts$ which are executed without preemption. Each activity act is a tuple $\langle actid, Pred, dur, es, le, res \rangle$ where: $actid$ is a unique identifier in the enactment plan, $Pred$ is the list of its precedence activities (i.e., those activities which must be executed before act), dur is the estimated duration of act , es is the earliest start time (i.e., the soonest that the activity

⁶ For the sake of clarity, traces represent sequences of activities, i.e., no parallelism is considered in the examples. Moreover, only completed events for activity executions are included in the trace representation.

act can start), *le* is the latest end time (i.e., the latest that the activity *act* can finish), and *res* is the resource which performs the activity *act* in the plan.⁷

Such definition is provided to formalize the concepts which already exist in the literature related to that term.

The activities of an enactment plan which are not preceded by any other activity are called initial activities. In a similar way, the activities which do not precede any other activity are called final activities. Therefore, regarding the precedence relations between the activities (stated by the *Pred* attribute) and the parallelism that exists between activities which are executed by different resources (stated by the *res* attribute) the schedule can be represented as a graph (cf. Definition 5). We use the graph definition introduced in [70]. In such graph, both a start node which precedes all of the initial activities and a final node which is preceded by all the final activities are additionally included.

Definition 5. A **Graph** $G = (gid, N, Edges)$ is identified by *gid* and consists of a set of pairs of nodes $n \in N$, i.e., *Edges*. Each edge denotes a direct edge between two nodes in the graph. A node $n \in N$ is a tuple $\langle nid, l, t \rangle$ where *nid* is a unique identifier of a node in the graph, *l* is its label, and *t* is its type.

Such definition of graph allows to represent a schedule (i.e., an enactment plan) in many different languages, e.g., BPMN or EPC (cf. Example 2). Note that a one-to-one relation is considered between a graph and an enactment plan. As an example, the types of nodes (i.e., *t*) in BPMN language [12] are ‘activity’, ‘event’, or ‘gateway’. A node of type ‘gateway’ allows labels (i.e., *l*) ‘AND’, ‘OR’, ‘XOR’, etc., while ‘event’ nodes allow ‘start’ and ‘end’ labels.

Example 2. Fig. 4 (a) shows two schedules related to how to prepare a holiday where the activities *book a hotel*, *select the clothes* and *prepare the luggage* are considered. Since both schedules include the same activities, which are executed by the same resources and also in the same order, they result in the same enactment plan. As can be seen in the Gantt diagram related to the enactment plan (cf. Fig. 4 (b)), the activity *Book hotel* presents 1 temporal unit of slack. In addition, Fig. 4 (c) shows the related graph using BPMN.⁸ This graph consists of the following 7 nodes (cf. Definition 5): $\langle 1, start, event \rangle$, $\langle 2, AND, gateway \rangle$, $\langle 3, book, activity \rangle$, $\langle 4, select, activity \rangle$, $\langle 5, AND, gateway \rangle$, $\langle 6, pack, activity \rangle$ and $\langle 7, end, event \rangle$; which are paired (cf. Definition 5) as follows: (1,2), (2,3), (2,4), (3,5), (4,5), (5,6), and (6,7).

In such context, constraint programming (CP) [72] supplies a suitable framework for modeling and solving problems involving P&S aspects [76]. In order to solve a problem through CP, it needs to be modeled as a constraint satisfaction problem (CSP) (cf. Definition 6).

Definition 6. A **CSP** $P = (V, D, C_{CSP})$ is composed of a set of variables V , a set of domains D which is composed of the domain of values for each variable $var_i \in V$, 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.

⁷ Note that, since activities are executed without preemption and the same resource cannot be used to perform more than one activity in parallel, there are implicit precedence relations between the activities which are executed by the same resource since our approach does not allow a resource doing multiple activities in parallel.

⁸ For simplicity, role information is shown inside the activity boxes in the BP graph. In [70], a general solution for managing role information and other non-control-flow elements is shown.

The given definition is provided to formalize the concepts which already exist in the literature related to CSP.

A solution to a CSP (cf. Definition 7) consists of assigning values to CSP variables.

Definition 7. A **solution** $S = \langle (var_1, val_1), (var_2, val_2), \dots, (var_n, val_n) \rangle$ for a CSP $P = (V, D, C_{CSP})$ is an assignment of a value $val_i \in dom_i$ to each variable $var_i \in V$.

A solution is **feasible** when the assignments variable-value satisfy all the constraints, i.e., a goal state is reached. In a similar way, a CSP is feasible if at least one feasible solution for this CSP exists. From now on, S^{var} refers to the value assigned to variable *var* in a solution *S*.

Similar to CSPs, constraint optimization problems (COPs, cf. Definition 8) require solutions that optimize an objective function.

Definition 8. A **COP** $P_o = (V, D, C_{CSP}, o)$ related to a CSP $P = (V, D, C_{CSP})$ is a CSP which also includes an objective function *o* to be optimized.

A feasible solution *S* for a COP is **optimal** when no other feasible solution exists with a better value for the objective function *o*.

Constraint programming allows to separate the models from the algorithms, so that once a problem is modeled in a declarative way as a CSP, a generic or specialized constraint-based solver can be used to obtain the required solution. Furthermore, constraint based models can be extended in a natural way, maintaining the solving methods. Several mechanisms are available for solving CSPs and COPs [72], which can be classified as search algorithms (i.e., for exploring the solution space to find a solution or to prove that none exists) or consistency algorithms (i.e., filtering rules for removing inconsistent values from the domain of the variables). In turn, search algorithms can be classified as complete search algorithms (i.e., performing a complete exploration of a search space which is based on all possible combinations of assignments of values to the CSP variables) and incomplete search algorithms (i.e., performing an incomplete exploration of the search space so that, in general, to get a feasible or an optimal solution is not guaranteed). In this work we apply P&S to generate different possible enactment plans from the same constraint-based BP model through an incomplete search algorithm.

Since actual problems typically involve multiple conflicting objective functions, multi-objective constraint optimization problems (MO-COPs, cf. Definition 9) are considered in the current work. The reader is referred to [26] for a review of the literature on MO-COPs.

Definition 9. A **MO-COP** $MP_o = (V, D, C_{CSP}, OFs)$ related to a CSP $P = (V, D, C_{CSP})$ is a CSP which also includes a set of objective functions *OFs* to be optimized (maximized or minimized).

Such definition is provided to formalize the concepts which already exist in the literature related to that MO-COP.

In multi-objective optimization problems, usually no unique optimal solution exists, but a set of Pareto optimal solutions (cf. Definition 10) can be found.

Definition 10. Let *Sols* be the set of all the solutions of a MO-COP MP_o which includes *n* objective functions, i.e., $OFs = OF_1, \dots, OF_n$. Then, a solution $sol_1 \in Sols$ is **Pareto optimal** if $\nexists sol_2 \in Sols$ such that $\forall OF \in OFs : sol_2^{OF}$ is better or equal than sol_1^{OF} , i.e., for obtaining a feasible solution which improves one objective functions, at least another objective function needs to be deteriorated.

Since many MO-COPs present NP complexity [32], Pareto optimized solutions are considered (cf. Definition 11).

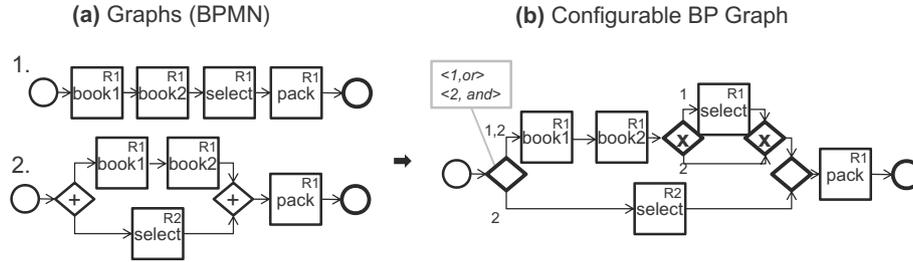


Fig. 5. Two enactment plans (a) are merged into a single configurable BP model (b).

Definition 11. Let $Sols$ be the set of all the solutions of a MO-COP MP_o and let $Sols_z \subseteq Sols$ be the subset of the solutions already explored at certain time. Then, a solution $sol_1 \in Sols_z$ is **Pareto optimized** if it is Pareto optimal regarding only the subset $Sols_z$, i.e., if $\nexists sol_2 \in Sols_z$ such that $\forall OF \in OFs : sol_2^{OF}$ is better or equal than sol_1^{OF} .

To solve multi-objective optimization problems there are, basically, three approaches:

1. Defining a new objective function by combining the original objective functions, e.g., through weighted-sum function [16,30,95]. However, these approaches does not necessarily guarantee that the final solution will be neither acceptable [4,19,46,59,85] nor Pareto optimized [19].
2. Working with stochastic algorithms like genetic, simulated annealing or ant colonies algorithms to obtain a set of Pareto optimized solutions. For example, previous works applied the simulated annealing technique [83,87] and evolutionary multi-objective optimization algorithms [21,82] for solving multi-objective optimization problems.
3. Optimizing one of the objective functions while constraining the other ones (e.g., ε -constraint method [38]). These methods are based on optimizing only one of the objective functions while all the others objective functions are used to state additional constraints. Here, the main challenge is to select the proper bounds for the objectives which are not optimized. Each approach typically solves this issue with its own method. In general, each single-objective problem is solved several times by varying the value of the bound. The complete set of Pareto optimal solutions can be figured out if the bounds are adequately varied [52].

In this work, the ε -constraint method [38] is applied since it appeared well suited for our purposes and typically provides good results. In addition, in a previous approach [44], such a technique was applied and it achieved good results. Furthermore, the base algorithm to be used in the ε -constraint method were developed and analyzed in previous approaches [7,9] and results showed its effectiveness for improving performance. However, other promising multi-objective optimization techniques (e.g., stochastic algorithms) could be also applied.

3.3. Configurable BP models

Typically, different enactment plans (cf. Definition 4), also called *variants* in the context of BPs, can be performed in scenarios which entail high variability. In most cases these plans share many commonalities. Hence, these variations can be combined in a configurable BP model leading to a compact representation [22,49,70,71]. Generally, configurable BP models allow analysts to understand what these variations share, what their differences are, and why and how these differences occur [71].

Configurable BP models are typically created by hand (1) from scratch, (2) from an existing BP model by including possible adaptations [35], or (3) by merging some BP models related to the same or similar goals which already exist [70]. In the last case, the source BP models need to be compared and merged, which might result in a tedious, time-consuming and error-prone process if it is performed by hand [70]. To overcome these problems, there exist approaches focused on automatically merging different BP models in a configurable BP model [68,70].

Configurable BP models can be represented by configurable BP graphs, which are defined (cf. Definition 12) based on [70].

Definition 12. A **Configurable BP Graph** $CG = (G, E2I, N2LI)$ consists of: (1) a graph, $G = (gid, N, Edges)$ (cf. Definition 5), (2) a function $E2I$ that maps each edge $e \in Pairs$ to a set of process graph identifiers (i.e., $E2I$ identifies which branches of CG belong to each source graph which is merged in CG), (3) a function, $N2LI$ that maps each node $n \in N$ to a set of pairs $\langle gpid, l \rangle$ where $gpid$ is a graph identifier and l is the label of node n in graph $gpid$ (i.e., $N2LI$ identifies which nodes, with the corresponding label, belong to each graph which is merged in CG).

In the current approach, configurable BP models are created by using the process merger tool presented in [70] after being adapted to work with BPMN. This tool is based on a merging algorithm which analyzes the similarities of the input models (i.e., the graphs) and creates a new model (i.e., the configurable BP graph) which includes configuration nodes for those points where the input models are different. Therefore, each branch and node of the configurable BP model can be related either to one or more graphs. To store these relations, each branch/node of the configurable BP graph includes identifiers related to the corresponding plan (i.e., $E2I$ function). In addition, nodes also store the associated label related to each identifier (i.e., $N2LI$ function). Since a configurable BP model includes different graphs, we consider that a configurable BP model includes different enactment plans (cf. Example 3).

Example 3. Fig. 5 shows 2 graphs which are merged into a configurable BP model.⁹ The first gateway in Fig. 5(b) is a configurable node which corresponds to an ‘OR’ gateway in the process 1 and an ‘AND’ gateway in the process 2.

3.4. Dealing with uncertainty

Uncertainty is typically present in most real scenarios. Therefore, mechanisms to deal with uncertainty are required for real systems. In such context, flexibility and robustness concerns have received increasing attention in last years [3,18,20,34,37,86], also in the context of BPs [66,78].

Although flexibility and robustness are typically used to evaluate techniques or tools which cope with the natural uncertainty of

⁹ As there is not ambiguity, some labels are not shown (i.e., they are the same as in the branch).

real scenarios, literature related to adaptability-like topics acknowledges that there is not a simple and general definition for these terms. Some approaches even define different terms with similar definitions [3,34], which leads to misunderstandings. Between dozens of existing definitions of flexibility and robustness, in this work a representative set of them is selected in order to highlight their commonalities and differences.

On the one hand, flexibility was defined in the 80s as “the ability of an organization to adapt to [...] changes that [...] impact on the organization’s performance” [1]. It was supported by [89] in the 90s and more recently by [78], which applies this term to BPs. In accordance with [1,34,66,78,89], we define the term flexibility as the capability to *adapt* a plan to external events in order to achieve a goal (i.e., to change the original plan to a new plan which generally has a different performance but achieves the same goal). Note the active feature of the verb *adapt* - the flexibility is an active ability (cf. Example 4).

Example 4. A person who is going to the cinema wearing summer clothes when it is sunny but the forecast is uncertain increases the flexibility taking a foldable raincoat in a handbag. This way, taking a raincoat makes adaptation to the weather possible, and hence, the flexibility is increased (note that changing the clothes is necessary only if it rains).

On the other hand, robustness was defined in the 70s as “the ability to respond successfully to [...] environmental changes” [27],¹⁰ which was recently supported by [20]. Similar definitions can be found in P&S [41], in complex systems [15] and in project management [37]. In a related way, we define the term robustness [15,18,20,27,37,41,42] as the capability of a process to *withstand* external events in order to prevent undesirable impacts (i.e., changing the plan is not required, and hence, the same performance after the occurrence of the events is reached). In contrast to flexibility, robustness is considered a passive ability (cf. Example 5).

Example 5. The same person of Example 4 increases the robustness wearing mid-season clothes instead of summer clothes since the mid-season clothes can withstand good and bad weather, and hence, the robustness is increased (note that, unlike in Example 4, in this case changing the clothes is not required).

4. Generating an optimized configurable BP model from a declarative specification

This section presents a method for generating an optimized configurable BP model using a declarative specification as starting point. For the declarative specification of BPs, in the current approach an extension of the constraint-based language called Declare is proposed (cf. Fig. 2 (1)). The generation of configurable BP models from declarative specifications is divided in three steps:

1. A set of multi-objective optimized enactment plans is generated from the declarative specification (cf. Fig. 2 (2)). For this, a constraint-based approach is considered. In this step the high variability of the declarative specification is reduced by discarding bad plans according to the objective functions to be optimized.
2. The generated set of enactment plans is then filtered regarding different quality measures, i.e., robustness and flexibility (cf. Fig. 2 (3)). This step is based on selecting those plans which behave better against the uncertainty of the environment. Thus, the variability is slightly reduced.

3. To create a unified representation of all the enactment plans, a configurable BP model is created out of the filtered set of optimized enactment plans (cf. Fig. 2 (4)).

4.1. SDeclare language

As mentioned in Section 1.2, to specify the process in a declarative way, Declare [63] is used as basis. Motivated by requirements described in literature [60,63] as well as the necessities of the case study we have conducted we extend in this work Declare to SDeclare by considering: (1) resource reasoning, (2) temporal and data constraints and, (3) stochastic estimates.

Definition 13. A SDeclare process model $SDM = (SActs, Data, C_{BP}, AvRes, OFs)$ related to a constraint-based BP model $CM = (A, C_{BP})$ (cf. Definition 2) is composed of: a set of SDeclare activities (*S-Activities* in the following) $SActs = (a, Res, Atts)$ (cf. Definition 14) related to each $a \in A$, problem data information *Data* (which is the information which influences the process execution), a set of SDeclare constraints C_{BP} (which relates the activities included in *SActs* and *Data*), a set of available resources *AvRes*, and a set of objective functions to be optimized *OFs* (cf. Definition 3). The set *AvRes* is composed of tuples $\langle role, \#role \rangle$ meaning that there are $\#role$ available resources to *role*.

Definition 14. A S-Activity $SAct = (a, Res, Atts)$ represents a BP activity *a* (cf. Definition 2) which can be performed by any resource included in *Res*, and which has a set of attributes associated *Atts* (e.g., duration and profit). The set *Atts* is composed of tuples $\langle att, value \rangle$.

4.1.1. Resource reasoning

To support the direct reasoning with resources (which is not possible in Declare) we extended Declare by including: (1) alternative resources for executing each S-Activity (cf. *Res* in Definition 14), and (2) the set of available resources (cf. *AvRes* in Definition 13). In this way, SDeclare directly supports the most common workflow resource pattern, i.e., the role-based distribution [73], which also supports our case study. This pattern models the ability to specify at design-time one or more roles which will be assigned to the instances of an activity at run-time. Note that, besides the role-based distribution pattern, SDeclare is open to support further resource patterns [73] by including the related constraints in the proposed CSP model (cf. Definition 13). However, as mentioned, in this work we focus on the role-based distribution pattern, which is the one required for modeling the considered case study.

The information related to resource availabilities can be unknown until starting the BP enactment. Since this information is independent of the S-Activities, it can be changed without affecting the specification of the activities, and vice versa. This is not a problem for our proposal since static information (i.e., the control-flow and resource constraints) is complemented with more changing information (i.e., the estimates), and finally the most dynamic information (i.e., information about resource availabilities) is included. In this way, with our approach, the configurable BP model can be automatically generated just before starting the BP enactment by considering the actual values of the resource availabilities and estimates.

4.1.2. Temporal and data constraints

To support increased expressiveness of Declare templates, we extend it by considering temporal and data constraints. In this way SDeclare allows to specify temporal constraints in a similar

¹⁰ Actually, instead of robustness, the work [27] uses the terms “passive” or “internal” flexibility.

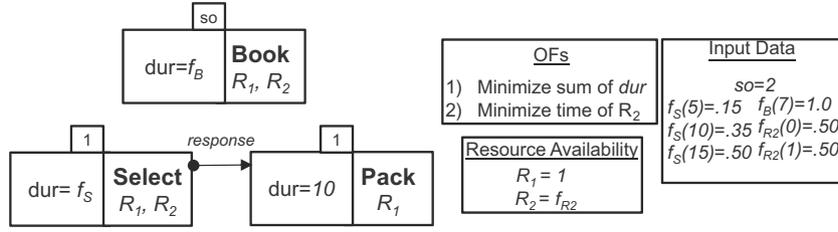


Fig. 6. Example of SDeclare process model.

way as [60,94], i.e., all the Declare constraints have been extended to support temporal modifiers, e.g., the SDeclare constraint *Precedence*($A, B, [5,10]$) states that for starting the execution of activity B , activity A needs to be finished between 5 and 10 time units before. Furthermore, we extend Declare by including data constraints in a similar way as [60].

Therefore, input data can be related to (1) activity attributes, e.g., in Fig. 6, the duration of the activity B is specified by the input data, (2) resource availability, i.e., the number of available resources of a role, and (3) SDeclare constraints, e.g., in Fig. 6, the selection of the choice constraint depends on the input data *availableActs*.

4.1.3. Stochastic estimates

As mentioned, to allow the specification of certain input uncertainty in the declarative BP models which are designed, we extended Declare by including the **stochastic attributes** for certain parts of the model (i.e., S-Activity attributes, data and temporal constraints, and resource availability). Estimates can be obtained by interviewing business experts or by analyzing past process executions. Moreover, both approaches can be combined to get more reliable estimates.

Since estimating values can be quite challenging [84], SDeclare allows specifying any discrete value of the model in a stochastic way by using probability mass functions (PMFs in the following) which are functions that give the probability of a variable taking a certain value. These PMFs can be associated to any input data (cf. *Data* in Definition 13) of the SDeclare model. Thus, the *Data* property of a SDeclare model consist of tuples $\langle dName, dValue/dPMF \rangle$, meaning that the data $dName$ can be associated to a fixed value ($dValue$) or to a PMF ($dPMF$) (cf. Example 6).

Example 6. In Fig. 6, a SDeclare process model related to how to prepare a holiday is depicted. To prepare a holiday three activities need to be performed, select the cloths, pack them, and book the flights between each stopover. The objective is to minimize the planning time and, additionally, to minimize the use of a second resource. Therefore, $SActs = \{Book, Select, Pack\}$, $OFs = \{minimize\ sum\ of\ dur, minimize\ time\ of\ R_2\}$, $AvRes = \{\langle R_1, 1 \rangle, \langle R_2, f_{R2} \rangle\}$ (i.e., the number of available resources of R_2 is defined by the PMF f_{R2}), $Data = \langle so, 2 \rangle$ and $C_{BP} = \{Response(Select, Pack), Exactly(1, Select), Exactly(1, Pack), Exactly(so, Book)\}$ (i.e., the number of repetitions of *Book* is defined by the input data *so*). In addition, $\{Book, Select, Pack\}$ are three S-Activities (cf. Definition 14) where $Book = \langle Book, \{R_1, R_2\}, \{\langle dur, f_B \rangle\} \rangle$, $Select = \langle Select, \{R_1, R_2\}, \{\langle dur, f_S \rangle\} \rangle$ (i.e., its durations are defined by the PMFs f_B and f_S), and $Pack = \langle Pack, \{R_1\}, \{\langle dur, 10 \rangle\} \rangle$. Note that there is uncertainty related to the duration of *Select*, i.e., it may last 5, 10 or 15 units of time and 15 is the most probable value.

Using PMFs, the estimates reflect the business reality better [2]. There are extensive studies focused on patterns of PMFs that represent the uncertainty best [2,5,28,56] that are not discussed here since it is out of the focus of this paper.

4.2. From the SDeclare model to a set of optimized enactment plans

In this section, the generation of a set of optimized enactment plans from a SDeclare model (cf. Definition 13) is explained. This includes: (1) the sampling of the stochastic properties of the SDeclare model to obtain a set of non-stochastic models, (2) the modeling of the non-stochastic models as MO-COPs, (3) the use of global constraints implemented through filtering rules to improve the modeling of the MO-COPs and to efficiently handle the constraints in the search for solutions, and (4) the search algorithm for solving the MO-COPs which obtains the set of optimized enactment plans.

4.2.1. Sampling the SDeclare model

As stated in Section 4.1.3, stochastic properties can be included in the SDeclare model (e.g., S-Activity attributes, data properties or resource availabilities). These properties represent the input uncertainty that is considered in the scenario and it is used to evaluate the flexibility of the configurable BP model and the robustness of the optimized enactment plans included in the model, as explained later.

For managing the uncertainty of the SDeclare model when generating the related optimized enactment plans, the different stochastic properties are sampled (cf. Definition 15) by considering their associated PMFs.

Definition 15. Let $SDM = (SActs, Data, C_{BP}, AvRes, OFs)$ be a SDeclare model with n stochastic properties $prop_1, \dots, prop_n$. Then: a **sample** is a set of n tuples $\langle prop_i, val_i \rangle, i = 1 \dots n$ which indicates the fixed value val_i that the property $prop_i$ takes in such sample. The value val_i is randomly selected considering the PMF related to $prop_i$ (i.e., f_{prop_i}).

Each sample is used to create a non-stochastic model (cf. Definition 16) from a SDeclare model by assigning a fixed value to each stochastic property (cf. Example 7). In the proposed approach, multiple samples are generated in order to obtain a representative set of non-stochastic models. Each non-stochastic model is, in turn, transformed to a MO-COP.

Definition 16. A **non-stochastic SDeclare model** is a SDeclare model in which all properties are defined by fixed values.

Example 7. Regarding the SDeclare model of Fig. 6, a possible sample could be: $\{\langle f_S, 10 \rangle, \langle f_B, 7 \rangle, \langle f_{R2}, 0 \rangle\}$. Applying the sample to the SDeclare model, the non-stochastic SDeclare model of Fig. 7 is obtained.

4.2.2. Transforming the non-stochastic models into MO-COPs

For applying constraint programming for the generation of optimized enactment plans, each non-stochastic model (i.e., a SDeclare model plus a sample) is translated to a MO-COP (cf. Definition 9). Regarding the proposed MO-COP, S-Activities (repeated activities in the MO-COP, cf. Definition 17), which can be executed arbitrarily

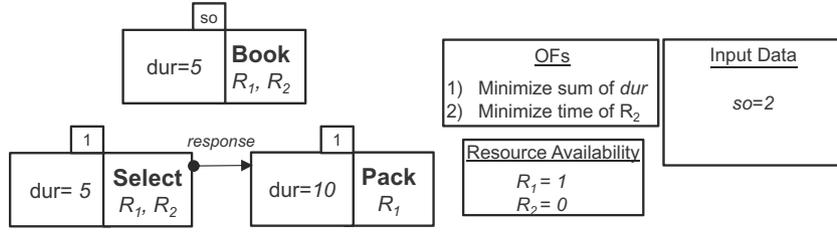


Fig. 7. A non-stochastic SDeclare model resulting from applying the sample of Example 7 over the SDeclare model of Fig. 6.

```

TemporalPrecedence(A,B,[min, max]) is added OR bounds of nt(B) changed OR bounds of
st(act(B,1)) changed OR bounds of et(act(A,1)) changed ->
  If LB(nt(B)) > 0 then
    nt(A) <- nt(A) - {0}
  If LB(et(act(A,1))) + min > LB(st(act(B,1))) then
    LB(st(act(B,1))) <- LB(et(act(A,1))) + min
  If UB(et(act(A,1))) - max > UB(st(act(B,1))) then
    UB(et(act(A,1))) <- UB(st(act(B,1))) - max

```

Fig. 8. Filtering rule for temporal precedence template in SDeclare.

often if not restricted by any constraint, are modeled as sequences of optional scheduling activities (cf. Definition 18). This is required since each execution of a repeated 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 17. A **repeated activity** $ra = (a, Res, Atts, nt)$ is a S-Activity $SAct = (a, Res, Atts)$ (cf. Definition 14) which can be executed several times. This is managed by defining a CSP variable which specifies the number of times the S-Activity is executed (i.e., nt).

For each repeated activity, nt_{MAX} ¹¹ scheduling activities exist, which are added to the CSP specification.

Definition 18. A **scheduling activity** $sa = (st, et, res, sel)$ related to a repeated activity $ra = (a, Res, Atts, nt)$ represents a specific execution of ra , where st and et are CSP variables indicating the start and the end times of sa , respectively, $res \in Res$ is a CSP variable representing the resource used for its execution, and sel is a CSP variable indicating whether or not sa is selected to be executed (i.e., equal to 0 in the case that it is not executed and equal to 1 otherwise).

In this way, the SDeclare model $SDM = (SActs, Data, C_{BP}, AvRes, OFs)$ is initially modified by the samples which are considered in such a way that one non-stochastic model is generated for each sample. Each non-stochastic model of this set is, in turn, transformed to a MO-COP $P_o = (V, D, C_{CSP}, OFs)$ (cf. Definition 9) where:

1. $V = \{nt(a), a \in SActs\} \cup \{st(a_i), et(a_i), res(a_i), sel(a_i), i \in [1.. nt_{MAX}(a)], a \in SActs\} \cup OFs$, i.e., V includes the CSP variables related to the repeated and scheduling activities together with one CSP variable for each objective function.
2. D is composed of the domains of each CSP variable, where $UB(var)$ and $LB(var)$ represent the upper and lower bounds of the domain of var , respectively.
3. C_{CSP} is composed of the global constraints (implemented by the filtering rules) related to C_{BP} together with the constraints which are inherent to the proposed model, which are listed as follows:
 - (a) $\forall a \in SActs \forall i : 1 \leq i < nt(a) : et(a_i) \leq st(a_{i+1})$ (i.e., a specific execution of a repeated activity precedes the next execution of the same activity).

- (b) $\forall a \in SActs \forall i : 1 \leq i \leq UB(nt(a)) : sel(a_i) == nt(a) > i$ (i.e., the nt variable of the repeated activity is directly related to the sel variables of the associated scheduling activities).
- (c) $\forall a \in SActs \forall i : 1 \leq i \leq nt(a) : st(a_i) + duration(a) == et(a_i)$ (i.e., the start and the end times of each scheduling activity are related by the estimated duration of the associated S-Activity).
- (d) $\forall OF_i \in OFs, optimize(OF_i)$.¹²

Resource constraints are not explicitly stated since most constraint-based systems provide a high-level constraint modeling specific to scheduling which includes an efficient management of shared resources.

4.2.3. Increasing performance with filtering rules

Many constraint-based approaches for modeling and solving P&S problems have been proposed [72]. Moreover, several proposals exist for filtering rules related to specialized scheduling constraints (e.g., [11,50]) Therefore, the considered problems could in principle be managed by adapting existing constraint-based approaches. However, some SDeclare templates entail complex reasoning about several combined innovative aspects, such as the alternating executions of activities together with the varying number of times which these activities are executed. Therefore, we implemented our own specific global constraints through innovative filtering rules to facilitate the specification of the problems. Moreover, the related filtering rules enable the efficiency in the search for solutions to increase since during the search process these filtering rules remove inconsistent values from the domains of the variables [6]. In this way, the constraints stated in the SDeclare specification (cf. Definition 13) are included in the MO-COP model through the related global constraints. In the MO-COP, initial estimates are made for upper and lower bounds of variable domains and these values are refined during the search process by the developed filtering rules (cf. Example 8).

In this work, filtering rules related to the constraints which were not considered in previous works¹³ (i.e., [6,43,44]) have been developed, i.e., related to choice and data.

¹² The optimization can be either maximization or minimization.

¹³ A detailed description of the developed basic SDeclare filtering rules can be found at <http://regula.lsi.us.es/MOPlanner/FilteringRules.pdf>.

¹¹ nt_{MAX} represents the maximum value of the initial domain of nt .

Example 8. The filtering rule related $TemporalPrecedence(A, B, [min, max])$ template is shown in Fig. 8, where the filtering rule that describes the pruning of domains appears after symbol \rightarrow . This constraint means that between min and max units of time before the first execution of B , at least one execution of A must be executed.

4.2.4. Solving the MO-COP

Once the problem is modeled as a MO-COP (cf. Definition 9), several constraint-based mechanisms can be used to obtain the solutions to the MO-COP, i.e., multi-objective optimized enactment plans (cf. Definition 4). Since the generation of optimal plans presents NP-complexity [32], it is not possible to ensure the optimality of the generated plans for all the cases. However, the developed constraint-based approach allows solving the considered problems in an efficient way as empirically demonstrate later in the case study.

Algorithm 1. Generation of *EnactmentPlans* from a MO – COP

input: MO-COP P_0
output: Set<EnactmentPlan> *plans*

- 1 Map<ObjectiveFunction, Range> *ranges* = calculateRegions(P_0);
- 2 Set<Region> *regions* = divide(P_0 , *ranges*);
- 3 Set<EnactmentPlan> *plans_with_dominated* = solveRegions(P_0 , *regions*);
- 4 *plans* = removeParetoDominated(*plans_with_dominated*);

The proposed constraint-based approach includes a multi-objective optimization search algorithm which is based on the ε -constraint method [38] (cf. Section 3.2). This search algorithm solves a number of single-objective COPs optimizing one of the objective functions and constraining the remaining objective functions. Specifically, given a MO-COP $P_o = (V, D, C_{CSP}, OFs)$ with N objective functions (i.e., $OFs = \{OF_1, \dots, OF_N\}$), the proposed algorithm (cf. Algorithm 1) follows four steps:

1. For each objective function $OF_i \in OFs$, the related range (i.e., tentative maximum and minimum values that can be obtained for OF_i) is calculated (line 1 of Algorithm 1) by using the algorithm calculateRegions (cf. Algorithm 2). At the beginning of Algorithm 2 an empty set of enactment plans is created for storing the solutions which are being generated (cf. line 1 of Algorithm 2). Moreover, a solver which is in charge of finding solutions (i.e., enactment plans¹⁴) for single-objective COPs is created (i.e., *solver* at line 2 of Algorithm 2). For each OF_i , a COP $P_i = (V, D, C_{CSP}, OF_i)$ which includes the same variables, domains and constraints than P_o but which only optimizes OF_i is generated (lines 4-6 of Algorithm 2). Then, an incomplete complete search algorithm is used to find one optimized solution Sol_i for such problem within a given time limit (cf. line 7 of Algorithm 2). The solution is then stored in the set *sols* (cf. line 8 of Algorithm 2). All the solutions which are store in *sols* are then used to calculate a range of tentative maximum and minimum values for each objective function OF_i (cf. lines 9-13 of Algorithm 2). This is performed by calculating the maximum and minimum values which are achieved for each OF_i in all the solutions stored in *sols* (cf. Example 9).

Example 9. For a MO-COP with three objective functions and $sols = \{(OF_1 = 10, OF_2 = 5, OF_3 = 4), (OF_1 = 9, OF_2 = 6, OF_3 = 1), (OF_1 = 2, OF_2 = 4, OF_3 = 8)\}$, the maximum (minimum) value for each OF_i , denoted as OF_i^M (OF_i^m), is: $OF_1^M = 10, OF_2^M = 6$ and $OF_3^M = 8$ ($OF_1^m = 2, OF_2^m = 4$ and $OF_3^m = 1$). Then, $range(OF_1) = [2, 10]$, $range(OF_2) = [4, 6]$ and $range(OF_3) = [1, 8]$.

Algorithm 2. calculateRegions method: Calculate the range of values for each *ObjectiveFunction*.

input : MO-COP P_0
output: Map<ObjectiveFunction, Range> *ranges*

- 1 Set<EnactmentPlan> *sols* $\leftarrow \emptyset$;
- 2 COPSolver *solver* = createSolver();
- 3 **foreach** OF_i in $P_0.OFs$ **do**
- 4 COP P_i = createCOP();
- 5 $P_i.(V, D, C_{CSP}) = P_0.(V, D, C_{CSP})$;
- 6 $P_i.OF = OF_i$;
- 7 EnactmentPlan $Sol_i = solver.solve(P_i, \mathbf{Best}, TIME_LIMIT)$;
- 8 *sols.add(Sol_i)*;
- 9 **foreach** OF_i in $p0.OFs$ **do**
- 10 Range $r = createRange()$;
- 11 $r.max = maximumValue(sols, OF_i)$;
- 12 $r.min = minimumValue(sols, OF_i)$;
- 13 *ranges.put(OF_i, r)*;

2. With the goal of obtaining an uniformly distributed set of solutions for P_o , the solution space (i.e., a N -dimensional space) is divided into smaller N -dimensional regions (cf. line 2 of Alg. 1) by using the divide algorithm.¹⁵ A region of a solution space with N objective functions consists of N sub-ranges, each one related to one objective function. In the divide algorithm, each range which is calculated for each objective function in the step 1, $range(OF_i)$, is divided into a given number DIV of non-overlapped sub-ranges, i.e., $range_j(OF_i) \forall j = 1 \dots DIV$. Each sub-range $range_j(OF_i)$ of a range $range(OF_i)$ has the same size than the other sub-ranges related to the same objective function OF_i , with the exception of the first and the last sub-ranges, i.e., $\forall 2 \leq j \leq DIV - 1 : range_j(OF_i) = [OF_i^m + (j - 1) \times |range(OF_i)| / DIV, OF_i^m + j \times |range(OF_i)| / DIV]$, where $|range(OF_i)|$ refers to the size of $range(OF_i)$, i.e., $OF_i^M - OF_i^m$. Regarding the first and the last sub-ranges, since the solution space is not totally explored in step 1 (since the search algorithm stops when a time limit is reached) we cannot ensure that no solutions exist out of the calculated ranges. Therefore the minimum value of the first sub-range and the maximum value of the last sub-range are not fixed with the goal of avoiding missing some potential solutions. In this way, $range_1(OF_i) = [-\infty, OF_i^m + |range(OF_i)| / DIV]$, and $range_{DIV}(OF_i) = [OF_i^m + (DIV - 1) \times |range(OF_i)| / DIV, +\infty]$. Then, the sub-ranges related to each objective function are combined with the sub-ranges related to all the other objective functions with the goal of obtaining different regions, R_v , where $v \in \mathbb{N}^N$ is a vector which contains the indices of the sub-ranges which belong to R_v , i.e., $range_j(OF_i) \in R_v$ if and only if $v[i] = j, \forall 1 \leq i \leq N, \forall 1 \leq j \leq DIV$ (cf. Example 10).

Example 10. Fig. 9 depicts a solution space which is divided in nine regions (i.e., $R_{1,1}, R_{1,2}, \dots, R_{3,3}$) for a MO-COP with two objective functions (i.e., OF_1 and OF_2) whose ranges are divided in tree sub-ranges (i.e., $range_1(OF_1), range_2(OF_1), \dots, range_3(OF_2)$).

¹⁴ In the proposed approach the schedules (i.e., the raw solutions of the considered COPs) are directly transformed to enactment plans. Therefore, for the sake of simplicity, the solutions of such COPs are considered enactment plans.

¹⁵ Due to its triviality, unlike the other algorithms, the divide algorithm is not formally shown in algorithm shape.

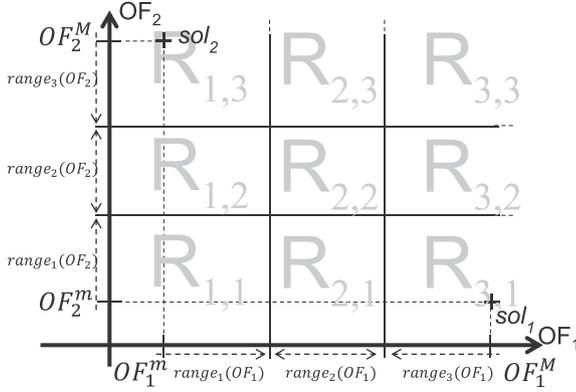


Fig. 9. Solution space with two objective functions which is divided into nine regions.

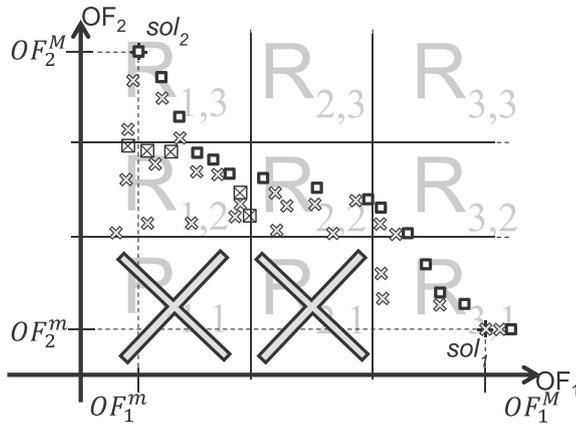


Fig. 10. Set of solutions for a 2-objectives MO-COP where the Pareto optimized solutions are depicted by squares, solutions which are Pareto dominated by solutions from the same region are depicted by crosses, and solutions which are Pareto dominated by solutions from other regions are depicted by crosses inside a box. Dominated regions are indicated with a big cross.

3. In order to look for a uniformly distributed set of solutions, each region is independently managed (line 3 of Alg. 1) by using the algorithm `solveRegions` (cf. Algorithm 3). Initially, an empty set of dominated regions is created (cf. line 1 of Algorithm 3). This set is created with the goal of storing all the regions which are dominated by others. Since only the Pareto optimized solutions are considered, the order of solving the regions influences the efficiency since some calculus can be saved by applying a proper ordering. Thus, this algorithm solves the aforementioned problems $P_{v,i}$ (i.e., a problem which optimizes the objective function i in the region R_v) starting with those problems which belong to the region which dominates more regions (line 2 of Algorithm 3). A region R_v dominates $R_{v'}$ if and only if $\forall 1 \leq k \leq N : v[k] > v'[k]$ (being N the number of dimensions), cf. Example 11.¹⁶ If a solution is found in a region, all the COPs related to the regions which are dominated by the former do not need to be solved (line 3 of Algorithm 3) since all their solutions are Pareto dominated by any solution which belongs to the former region.

Example 11. In Fig. 9, region $R_{3,3}$ dominates $R_{2,2}$, $R_{2,1}$, $R_{1,2}$ and $R_{1,1}$, region $R_{2,3}$ dominates $R_{1,2}$ and $R_{1,1}$, and so on.

¹⁶ For the sake of clarity, we assume that each objective function is maximized. The problem of minimization is analogous.

Therefore, for each region R_v which is not dominated, an empty set of solutions is created (cf. line 4 of Algorithm 3) to store all the enactment plans related to such region. In addition, N COPs $P_{v,i} = (V, D, C_v, OF_i)$ are generated (cf. lines 5 and 6 of Algorithm 3), where $C_v = C_{CSP} \cup \{OF_l \in range_{v[l]}(OF_l), \forall 1 \leq l \leq N\}$ (i.e., which only optimizes OF_i and where all the objectives functions are constrained to be in the related sub-range $range_{v[l]}(OF_l)$; cf. lines 7-10 of Algorithm 3). Unlike step 1, in order to generate a wide set of solutions, we use an *anytime optimization algorithm* [96] which is an incomplete search algorithm which updates the best solution which is found during the search. Then not only the best solution (cf. line 11 of Algorithm 3) but some intermediate solutions are returned.

Algorithm 3. `solveRegions` method: Generate *EnactmentPlans* by solving the COPs related to each region.

```

input : MO-COP  $P_0$ 
        Set<Region> regions
output: Set<EnactmentPlans> plans

1 Set<Region> dominated  $\leftarrow \emptyset$ ;
2 foreach region in regions by region.countDominatingRegions(regions) desc do
3   if !dominated.contains(region) then
4     Set<EnactmentPlan> sols  $\leftarrow \emptyset$ ;
5     foreach  $OF_i$  in  $P_0.OFs$  do
6       COP  $P_{v,i} = createCOP()$ ;
7        $P_{v,i}.(V,D,C_{CSP}) = P_0.(V,D,C_{CSP})$ ;
8        $P_{v,i}.OF = OF_i$ ;
9       foreach  $OF_j$  in  $P_0.OFs$  do
10         $P_{v,i}.C_{CSP}.add("OF_j \in region.get(OF_j).getRange()");$ 
11        sols = solver.solve( $P_{v,i}$ , Anytime, TIME_LIMIT);
12      if !sols.isEmpty() then
13        dominated.addAll(calculateDominatedRegions(regions, region));
14        plans.addAll(removeParetoDominatedSolutions(sols));

```

If at least one solution is found within a region R_v (cf. line 12 of Algorithm 3), the regions which are dominated by R_v are included in the set *dominated* to avoid the search for solutions in that dominated region (cf. line 13 of Algorithm 3). In addition, the solutions which are obtained within R_v are filtered by removing the solutions which are Pareto dominated (cf. line 14 of Algorithm 3 and Example 12).¹⁷

Example 12. For a MO-COP with two objective functions and a solution space divided in nine regions, Fig. 10 shows the different solutions obtained within each region, where no solutions are found in regions $R_{2,3}$ and $R_{3,3}$, and where $R_{1,1}$ and $R_{2,1}$ are eliminated (i.e., the COPs related to them are not solved) since some solutions are found in region $R_{3,2}$ which dominates $R_{1,1}$ and $R_{2,1}$. In Fig. 10, each cross represents a solution which is Pareto dominated by another solution in the same region.

4. After all the COPs are solved (i.e., a diversified set of solutions is obtained), solutions which are dominated by solutions from a different region are removed (cf. line 4 of Algorithm 1). Then a distributed set of Pareto optimized solutions is obtained (cf. Example 13).¹⁸

¹⁷ In a general case, the complexity of the Pareto dominance algorithm is $O(n^2)$ where n is the number of solutions [referencia]. Then, the fact of having all the solutions divided in non-overlapped regions (i.e., the solutions are clustered) reduces the complexity since $O(n^2) < O((n/m)^2) \times m, \forall m > 1$ where m is the number of regions.

¹⁸ The complete set of Pareto optimal solution is not the goal of the proposed algorithm, but a representative and distributed set of Pareto optimized solutions.

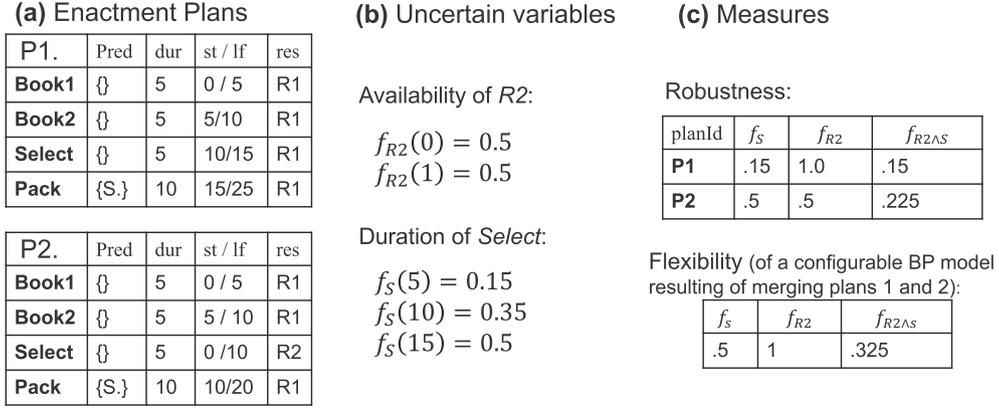


Fig. 11. For two different enactment plans (a) generated from the SDeclare model of Fig. 6, and considering two uncertain variables (b), the robustness of each plan and the flexibility of the related configurable BP model are calculated (c).

Example 13. In Fig. 10 all solutions which are dominated by any solution which belongs to a different region are depicted by a cross inside a box, and all the Pareto optimized solutions are depicted by a square.

4.3. Quantifying the flexibility and the robustness

In this section definitions for both flexibility and robustness are proposed in order to measure how the generated models deal with the uncertainty. Such definitions are based on related literature, which is introduced in Section 3.4. In the BP field, flexibility and robustness can be treated as quantitative attributes related to a specific process model. In the context of this paper, robustness and flexibility are quantified over enactment plans (cf. Definition 4) and configurable BP models (cf. Definition 12) respectively.

As mentioned in Section 3.3, a configurable BP model includes different enactment plans since there exists a one-to-one relation between graphs and enactment plans. Each enactment plan which is included in such models has its own level of robustness against a specific variable which presents uncertainty (cf. Definition 19). This uncertainty is defined through the related PMF which is included in the SDeclare model (cf. Definition 13).

Definition 19. Let P_i be an enactment plan (cf. Definition 4); let v be a variable related to some attribute of P_i and which is defined in the domain $D(v)$ with a PMF $f_v : D(v) \rightarrow [0 \dots 1]$, $\sum_{x \in D(v)} f_v(x) = 1$; and let $W(P_i, v)$ be the set of values of v which P_i withstands, i.e., P_i tolerates scenarios where v takes any value in $W(P_i, v)$ without changing its performance (e.g., without changing its objective function values, cf. Definition 3). Then: the **robustness of P_i against v** , $Rob(P_i, v)$, is the probability of the variable v taking a value that P_i withstands. When v is a discrete variable, then $Rob(P_i, v) = \sum_{x \in W(P_i, v)} f_v(x)$. When v is a continuous variable, $W(P_i, v)$ is considered as the non-overlapped ranges of values of v (i.e., $[[r1_{inf}, r1_{sup}], [r2_{inf}, r2_{sup}] \dots]$) which P_i withstands; then $Rob(P_i, v) = \sum_{x \in W(P_i, v)} (\int_{rx_{inf}}^{rx_{sup}} f_v(x) dx)$.

In this way, the *robustness against a variable* is applied over single alternatives (i.e., single enactment plans) of a configurable BP model. By contrast the term flexibility (cf. Definition 20) is applied over configurable BP models (cf. Example 14).

Definition 20. Let P be a configurable BP model which contains different enactment plans $P_i \in P$; and let v be a variable related to some attribute of P and which is defined in the domain $D(v)$ with a PMF $f_v : D(v) \rightarrow [0 \dots 1]$, $\sum_{x \in D(v)} f_v(x) = 1$. Then: the **flexibility of P against v** , $Flex(P, v)$, is the probability of the variable v taking a value that P withstands by adapting its workflow to any of its

alternatives P_i . When v is a discrete variable, then $Flex(P, v) = \sum_{x \in \bigcup_{P_i \in P} W(P_i, v)} f_v(x)$. When v is a continuous variable, $W(P_i, v)$ is considered as the non-overlapped ranges of values of v (i.e., $[[r1_{inf}, r1_{sup}], [r2_{inf}, r2_{sup}] \dots]$) which P_i withstands; then $Flex(P, v) = \sum_{x \in \bigcup_{P_i \in P} W(P_i, v)} (\int_{rx_{inf}}^{rx_{sup}} f_v(x) dx)$.

When the variable v follows a flat distribution, the robustness and the flexibility can be expressed as $Rob(P_i, v) = |W(P_i, v)| / |D(v)|$ and $Flex(P, v) = |\bigcup_{P_i \in P} W(P_i, v)| / |D(v)|$ respectively.

Example 14. Consider the two enactment plans depicted in Fig. 11 (a), and the two probability mass functions shown in Fig. 11 (b) (i.e., f_{R2} which is related to the number of available resources with role R2 and f_S which is related to the duration of the activity Select, i.e., S). Then, some measures can be calculated (cf. Fig. 11 (c)). The robustness of the enactment plan P_1 against S, $Rob(P_1, S)$, is equal to 0.15 since P_1 only withstands that activity Select takes 5 units of time. However, the robustness of P_1 against R2, $Rob(P_1, R2)$, is equal to 1 since P_1 is valid for any availability of R2 (note that R2 is not used in P_1). In a similar way, the robustness against these 2 variables is calculated for plan P2. Considering the last column of the table Robustness, it can be concluded that the enactment plan P2 manages the uncertainty better than P1. In a related way, once a configurable BP model is created by merging these two enactment plans, then the flexibility of such model can be calculated as stated in Definition 20. Therefore, $Flex(P, R2)$ is equal to 1, since P includes plans which can withstand any value of R2, and $Flex(P, S)$ is equal to 0.5, since the value 15 for S is not withstood by any plan of P . Considering both variables together, $Flex(P, R2 \wedge S) = 0.325$ which means that the 32.5% of the input uncertainty is properly managed by the configurable BP model.

4.4. From a set of optimized enactment plans to a configurable BP model

In this section a method for the automatic generation of a configurable BP model which subsumes the plans which are previously calculated according to flexibility and robustness issues is detailed. Such method includes: (1) the selection of the relevant plans, and (2) the merging of the relevant plans into a configurable BP model.

4.4.1. Selecting the relevant plans

In order to select the relevant plans (cf. Fig. 2 (3)) from the set of optimized enactment plans (denoted by PS from now on) a two-steps algorithm is proposed:

Table 1

Properties which are calculated for the set of enactment plans of Fig. 11, where S and R_2 are defined by the PMFs f_S and f_{R_2} respectively.

PMFs				Enactment plans			Measures				
S	f_S	R_2	f_{R_2}	$Plan_{id}$	$sum\ dur$	$timeR_2$	$W(P_i, S)$	$W(P_i, R_2)$	$Rob(P_i, S)$	$Rob(P_i, R_2)$	$\overline{Rob}(P_i)$
5	0.15	0	0.5	P1	25	0	[5]	[0, 1]	0.15	1.0	0.575
10	0.35	1	0.5	P2	20	5	[5, 10]	[1]	0.50	0.5	0.5
15	0.5										

1. Considering that the uncertainty of the scenario is specified over the stochastic variables (i.e., v_p) associated to some properties of the SDeclare model, a set of properties are calculated for each enactment plan $P_i \in PS$ (cf. Example 15):

- Withstood values for each uncertain variable: For each v_p , the range of withstood values are calculated (i.e., $W(P_i, v_p)$). Note that calculating the withstood ranges of the S-Activity attributes or of the availability of resources might be trivial. However, when the uncertainty is specified over data properties which affect a constraint, calculating the withstood ranges may require more elaborated calculus.
- Robustness for each uncertain variable: The robustness against each uncertain variable v_p (i.e., $Rob(P_i, v_p)$) is calculated as stated in Definition 19.
- Average robustness: Finally, the general robustness of a plan (i.e., $Rob(P_i)$) is calculated as the mean of the individual robustness of each stochastic variable.

Example 15. Table 1 shows a set of 2 optimized enactment plans (cf. column $Plan_{id}$) generated from the SDeclare model shown in Fig. 6 with two uncertain variables f_S and R_2 (cf. column PMFs) and its associated objective function values (cf. columns $sum\ dur$ and $timeR_2$). Moreover, regarding the domains that each plan withstands against v_1 and v_2 (cf. columns $W(P_i, v_1)$ and $W(P_i, v_2)$ respectively), the value of the robustness against these variables can be calculated as stated in Definition 19. The values of the robustness are depicted on columns $Rob(P_i, v_1)$ and $Rob(P_i, v_2)$ respectively. Furthermore, the value of the general robustness (cf. column $Rob(P_i)$) is calculated as the mean of $Rob(P_i, v_1)$ and $Rob(P_i, v_2)$.

2. In this step, the relevant plans are selected. For this, three different policies can be considered:

- All plans are kept: No plan is removed. In this case, the non-desirable variability is not reduced (cf. Fig. 2 (3)).
- The plans which present the highest robustness are kept: The enactment plans are ranked by its average robustness. Then, a percentage of plans which present the lowest robustness are removed (cf. Example 16). The goal of this policy consists of creating a configurable BP model composed by the most robust plans to the detriment of the flexibility, i.e., this policy is not intended to cover the input uncertainty as much as possible.

Example 16. In Table 1 the plan P2 presents an average robustness which is significantly lower than the robustness of the other plans, and then it would be removed if this policy is followed.

- The plans which provide for the highest flexibility are kept: The minimum set of plans which covers the maximum input uncertainty is selected (i.e., the minimum set which maximizes the union of the withstood domains, cf. Example 17). The goal of this policy consists of creating a configurable BP model which provides for the highest flexibility, i.e., which embraces plans which cover as much uncertainty as possible typically to the detriment of the robustness of these plans.

Example 17. When following this policy, in Table 1 the plans P1 and P2 would be selected since they are not totally overlapped.

In this way, this second step removes some enactment plans regardless of their objective function values. Therefore, good plans (i.e., optimized plans) which were calculated in Algorithm 1 are removed here, and then, only those plans which are relevant (i.e., the plans which are selected according to a policy) remain. This way, only the plans which are both good and relevant are kept.

The proposed approach could be easily adapted to consider the robustness as an additional objective function when generating the set of optimized enactment plans. However, in that scenario non-optimal solutions would be included since a new dimension would be considered in Algorithm 1, i.e., the robustness.

4.4.2. Merging the relevant plans into a configurable BP model

As stated in Section 3.3, an adaptation of the Process Merger tool [70] is used to create the configurable BP model out of the selected plans. The enactment plans to be merged are identified by a label, i.e., pid attribute (cf. Definition 4, column $Plan_{id}$ in Table 1). The generated configurable BP model has special nodes called configurable nodes which represent the variation points of the model (cf. Fig. 5 (b)). In addition, each arc of the configurable BP model has a reference to the labels of the plans to which the arc belongs. The variant to be executed is selected from the configurable BP model before the run-time phase regarding (1) the actual values of the uncertain variables of the scenario, (2) the robustness of the plans which withstand such actual values, and (3) the values of the objective functions. Just in case the flexibility which is obtained becomes insufficient (i.e., none of the enactment plans which are in the configurable BP model withstands the actual values of the uncertain variables), replanning becomes necessary and new optimized BP enactment plans will be generated by considering the actual values of the uncertain variables instead of the PMFs and then a new configurable BP model will be created.

5. A real example: a beauty salon of Seville

This section introduces a real example from a beauty salon that is used to validate the current proposal in the considered case study.

Motivation: The considered business has grown considerably in the last years. It has expanded from a small salon with three employees to more than six and included additional facilities to be able to offer additional services. In addition, the uncertainty regarding different aspects of the business has become an important problem, e.g., the arrival time of the clients or the availability of some resources during the day (e.g., due to a resource who feels sick at the beginning of the day but not enough to leave the salon). These changes, including the quick growth together with the complex constraints which need to be obeyed, resulted in problems related to the management of the salon. In particular, long waiting time for clients and missing schedules for employees are causing problems, affecting customer satisfaction and profit of the business.

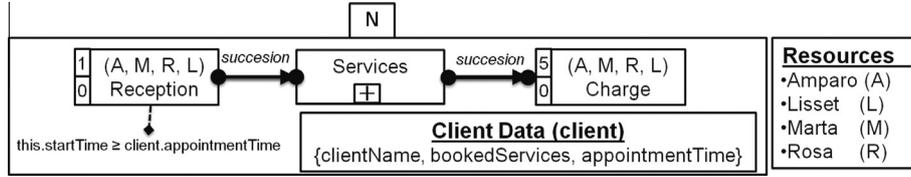


Fig. 12. SDeclare model for the beauty salon problem (top level process).

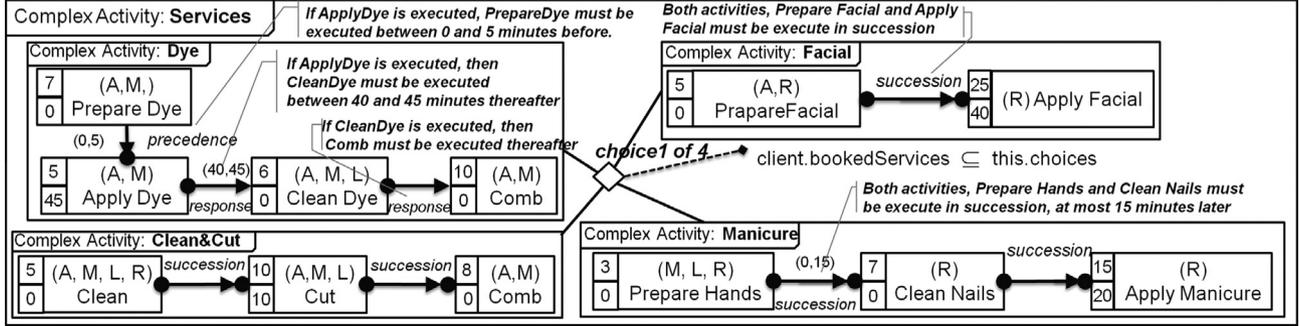


Fig. 13. SDeclare model for some of the services which are offered.

Improving the management: The goal of the business is to improve the current situation through the optimization of some business objective functions. Since our approach generates an optimized configurable BP model, a set of optimized schedules for employees can be suggested each one facing a different possible uncertain scenario, and therefore, the aforementioned problems can be overcome. Moreover, since multi-objective optimization is considered, several important objectives (i.e., minimizing waiting times for clients and maximizing profit) can be optimized. Furthermore, due to the high expressiveness of SDeclare, all the constraints which are given in the scenario together with its uncertainty can be specified.

Scenario details: The beauty salon offers various services¹⁹ like dye, clean&cut, manicure and facial services. It requires its clients to make appointment calls to know how many clients are coming as well as the booked services. There are several full-time employees, e.g., Amparo (A), Rosa (R), Lisset (L) and Marta (M). Each employee has different skills, and hence some activities can be performed by certain employees only. For all activities which are performed in the salon, the manager knows the average estimated duration, the profit which is obtained after their execution, and the employees which can execute that activity. The manager of the salon wants to plan and schedule a working day with several clients taking the following considerations into account:

1. The profit (P) of the resulting working plan has to be maximized (objective function 1).
2. The waiting time (WT) of the clients has to be minimized and distributed uniformly among all the clients (objective function 2):

$$WT = \sqrt{\frac{\sum_{c \in C} (s.endT(c) - c.appT - (\sum_{b \in c.serviced} b.estimate))^2}{C.size}}, \text{ where } C \text{ is the set of clients, } s \text{ is the considered solution, } s.endT(c) \text{ is the time when the client } c \text{ has finished, } c.appT \text{ is the appointment time of } c, c.serviced \text{ is the set of services which are applied to } c \text{ (i.e., included in the plan), and } b.estimate \text{ is the estimated duration for service } b.$$

¹⁹ For the sake of clarity, the depicted scenario is a subset of the actual beauty salon, i.e., the salon offers more services and has more employees.

3. The employees can offer some additional services to the client directly in the salon, and the client can accept or refuse. However, these additional services should only be proposed if this leads to optimized plans.

SDeclare specification: Typically, as illustrated in Fig. 12, a client visit starts with the reception in the beauty salon. After that, the staff applies some services to the client and, finally, the client is charged. Complex activity *Services* is composed of other activities²⁰ (e.g., dye, clean&cut, facial and manicure, cf. Fig. 13), while *Reception* and *Charge* are BP activities (cf. Definition 14). For each BP activity two attributes are considered: (1) estimated activity duration, and (2) profit which is obtained after executing the activity.²¹ Moreover, the set of alternative resources which can perform the BP activity is also included (cf. Example 18).

Example 18. In Fig. 12, activity *Reception* has an estimated duration of 1 minute and a profit of 0, and can be performed by A, R, M or L.

Notice that each instance created from the model of Fig. 12 represents one client visiting the beauty salon. The current problem deals with N clients (represented by the Existence constraint of Fig. 12, stated by the label N) which come to the salon at different times and with different bookings during a working day which are specified as data information.

The data perspective also appears in Fig. 12. The *Client-Data* includes all the information which is related to the client bookings, and consists of: (1) *clientName*, (2) *bookedServices*, which represents the mandatory services that the salon staff has to cover, and (3) *appointmentTime*, which is the time when the client is supposed to arrive at the salon. Through the data perspective, it is possible to model that activity *Reception* cannot start before the client appointment time (cf. Fig. 12). Moreover, a data constraint is used (in conjunction with the choice constraint) to ensure that all the services the client has booked are selected, i.e., the

²⁰ In a similar way to PSL [65], SDeclare allows hierarchical modeling (i.e., complex activities aggregate activities).

²¹ As can be seen in Figs. 12 and 13, the profit of the services is associated to one of the activities of the related services.

Table 2
Case study research questions.

Id	Research question
MQ1	Is Algorithm 1 appropriate for finding a uniformly distributed set of Pareto optimized solutions for SDeclare models of different complexity?
AQ1	Can Algorithm 1 find solutions within the different regions in which the solution space is divided?
AQ2	Does Algorithm 1 behave successfully independently of the complexity of the problems?
MQ2	Is the proposed approach useful for a business expert?
AQ3	Can the proposed approach improve the results which are manually obtained by an expert?
AQ4	Can the proposed approach generate an unified artifact which behaves properly against the input uncertainty?

Table 3
Quantified variables for the embedded design.

Variable	Description
$minWT, minP, MaxWT, MaxP$	Initial ranges for each objective function, which are calculated in the initial searches of Algorithm 1
$\%SF_{rx}$	The percentage of SDeclare models in which, at least one solution is found within region rx ($rx \in \{r1, r2, r3, r4\}$) regarding the total number of SDeclare models which are considered
$NActs_{rx}$	Size of the enactment plans which are generated within region rx , which is measured as the number of activities
WT_{rx}, P_{rx}	Value for each objective function which is obtained within region rx
$SAdd_{rx}$	Number of additional services per client which are included in the solutions which are found within region rx (i.e., those services which were not initially booked but are included in the plan)
$\%PS_{rx}$	Percentage of Pareto optimized solutions obtained within region rx regarding the total number of Pareto optimized solutions which are obtained for the considered problem

generated plans will always include the booked services (cf. [Fig. 13](#)). In addition, in the current scenario uncertainty is mainly present in the appointment time of clients and, less frequently, in the duration of activities (e.g., due to a temporal malfunctioning of the facilities or because people are not always working at the same place; also it can take a variable amount of time to cut the hair of a person depending on her type of hair) and in the availability of resources (e.g., employees falling sick). To accommodate for such situations, this uncertainty is considered in the SDeclare model before starting the process (cf. [Example 19](#)). In this case study flat discrete ranges are used as PMFs for simplicity.

Example 19. Let 14:30 be the appointment time of certain client. The manager knows that such client used to be unpunctual; therefore, she specifies in the SDeclare model her appointment time through the flat PMF $appT = \{14:25, 14:30, 14:35, 14:40\}$.²² In addition, before starting the day (and before generating the configurable BP model), one of the resources has informed the manager that she does not feel well; therefore, the manager specifies her availability through the flat PMF $avR = \{0, 1\}$ in the SDeclare model.

Generating the optimized plans: Given a SDeclare model $SDM = (Acts, Data, C_{BP}, AvRes, OFs)$ for the beauty salon problem, where $Acts, Data, C_{BP}$ and $AvRes$ are shown in [Figs. 12 and 13](#), and OFs are described in scenario details (i.e., maximization of the profit and minimization of the waiting time), the tool generates multi-objective optimized enactment plans. These plans represent personal schedules which will guide the employees to optimally serve the clients while facing the uncertainty, i.e., potentially avoiding unnecessary replannings. For example, the set of plans which is generated regarding the client of [Example 19](#) will probably include optimized plans which can deal with client arrival times between 14:25 and 14:40. In a similar way, the approach can deal with employees falling sick by generating different plans (some of the plans will assign tasks to this employee and some of them will not). These plans are, in turn, represented as a

²² In this example, 5 min is considered as the minimum amount of time which can be measured.

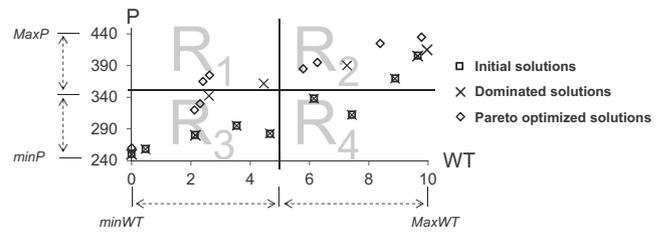


Fig. 14. Solutions which are found for a specific setting of the beauty salon problem.

configurable BP model by following one of the available policies. Such model will support the manager of the beauty salon in managing the working day in an optimized way.

In this way, the proposed approach provides support to the manager of the beauty salon by suggesting: (1) a resource for executing each activity, (2) the start and end time of the activities, and (3) the services which will be offered to each client (i.e., services which were not booked by the client).²³

6. Case study

This section provides an empirical study for: (1) evaluating the generation of multi-objective optimized enactment plans from SDeclare specifications through the proposed constraint-based approach, and (2) assessing whether the approach can deal with real problems which involve uncertainty. In this section, the case study protocol for the software engineering field proposed by [\[13\]](#) is followed to improve the rigor and validity of the study. Such protocol suggests the following sections: background, design, case selection, case study procedure, data collection, analysis and interpretation, and validity evaluation.

6.1. Background

In this section, previous research related to the topic of this paper is identified. Different proposals related to the generation

²³ As an example, two Pareto optimal plans for the beauty salon problem can be found at <http://regula.lsi.us.es/MOPlanner/PlansBeautySalon.pdf>.

Table 4
Quantified variables for the holistic design.

Variable	Description
$minWT, minP, MaxWT, MaxP$	Average of the minimum and maximum values of the objective functions (i.e., waiting time and profit) which are obtained by applying the proposed approach
$\overline{WT}, \overline{P}$	Average values of the objective functions which is obtained applying the proposed approach
$\% \Delta WT, \% \Delta P$	Average values of the percentage of increment of the objective functions of the plans which are obtained through the proposed approach versus the real execution plan
$\% Dominated$	Percentage of problems whose real execution plan is dominated (cf. Definition 10) by the Pareto front generated by our approach
$Flex$	Average flexibility (cf. Definition 20) of the generated configurable BP models against the uncertainty provided by the input SDeclare model
$minRob, MaxRob$	Average values of the minimum and maximum robustness (cf. Definition 19) of the plans which are included in the generated configurable BP models

of optimized BP enactment plans from declarative specifications as well as related to flexibility and robustness concerns are discussed later when analyzing the related work. In particular, our proposal uses the SDeclare language for the declarative specification of a BP, generates a set of optimized BP models out of it, and, finally, merge them into an optimized configurable BP model taking flexibility and robustness concerns into account. In such context, the *purpose of this study* is twofold: (1) the evaluation of the proposed algorithm (i.e., Algorithm 1) for generating multi-objective optimized BP enactment plans regarding the distribution of the generated plans along the solution space²⁴ and (2) the evaluation of the whole approach in terms of its suitability to deal with real problems involving uncertainty.

Taking the purpose of the study into account, two main research questions (MQs) are defined (cf. Table 2). In particular, MQ1 checks the suitability of Alg. 1, i.e., evaluates if the obtained optimized BP enactment plans are uniformly distributed over the solution space and if the algorithm performs well when dealing with complex problems. Since Alg. 1 is an initial step of the current approach, the quality of its results influences the quality of the complete approach. For this, MQ1 is divided in two additional research questions (AQs): (1) AQ1 checks whether Alg. 1 finds solutions within the different regions in which the solution space is divided and (2) AQ2 evaluates if Alg. 1 behaves successfully (i.e., finds a uniformly distributed set of solutions) when solving problems of different complexity. In addition, MQ2 assesses if the current approach can be useful to deal with real problems. For this, MQ2 is divided into two additional questions: (1) AQ3 checks whether the solutions which are obtained by our approach improves the actual solutions which are manually obtained by the domain expert for the real scenario in terms of their objective functions, and (2) AQ4 evaluates the generated configurable BP models in terms of its robustness and flexibility against the input uncertainty, which is important in order to avoid replanning.

6.2. Design

The *object of study* is the method which is proposed for generating optimized configurable BP models from SDeclare specifications. For this, two different designs are carried out in this case study.

1. An *embedded* design concerning Algorithm 1. Particularly, this first design considers one analysis unit: the generation of optimized BP enactment plans through solving MO-COPs. In this design, for addressing MQ1 (i.e., AQ1 and AQ2), a set of different non-stochastic SDeclare models are randomly generated leading to different complexities of the MO-COPs which have to be solved. Note that stochastic variables are not considered in this design.

²⁴ Note that one of the goals of Algorithm 1 consists of obtaining an uniformly distributed set of solutions.

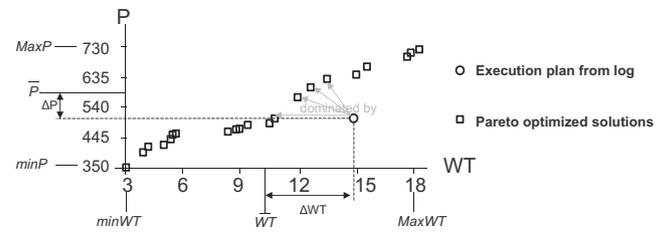


Fig. 15. Solutions which are found for a specific problem related to the beauty salon problem with some stochastic variables.

2. A *holistic* design which considers the overall proposal as a whole. Unlike in the embedded design, for addressing MQ2 (i.e., AQ3 and AQ4), stochastic variables are considered in this design for the appointment time of the clients and for the availability of resources. For dealing with such variables, the sampling step is configured to generate 30 different samples (cf. Definition 15) for each SDeclare model. This way, the non-stochastic SDeclare models (cf. Definition 16) are created in such a way that the stochastic properties of the source SDeclare model (i.e., the appointment times and the resource availabilities) are fixed by considering samples which are randomly generated taking the PMFs about such properties into account. Here, real data which are obtained from a log is considered for generating the input SDeclare models.²⁵

For both designs, Algorithm 1 considers dividing the solution space in 4 regions rx ($rx \in \{r1, r2, r3, r4\}$) and the constraint-based search Algorithm 2 is run until a 5-minutes CPU *TIME_LIMIT* is reached (cf. Algorithms 2 and 3), which is a reasonable amount of time for this business. It is run on a Intel(R) Xeon(R) CPU E5530, 2.40 GHz, 8 GB memory, running Debian 6.0.3. In order to solve the constraint-based problems, the developed algorithms have been integrated into the Comet system [25]. After the application of both the embedded and the holistic designs, the generated information (i.e., optimized BP enactment plans and configurable BP models respectively) is analyzed to answer the research questions (cf. Table 2).

For the embedded design, the data described in Table 3 is quantified (cf. Example 20) for each SDeclare model which is considered.

Example 20. Fig. 14 depicts the set of solutions which are found during the search process for a specific problem. As can be seen, Algorithm 1 divides the solution space in 4 regions. In order to state the limits of each region (i.e., $minWT, MaxWT, minP$ and

²⁵ The logged information comprises: (1) the number of clients of the day (i.e., N), (2) the booked services of each client (i.e., S), (3) the appointment time of each client (i.e., T), and (4) the resource availability of each day (i.e., AvRes). In addition, the manager of the beauty salon logged some stochastic properties related to T and to AvRes through PMFs.

MaxP), the solutions which are found in the first step of Algorithm 1 (cf. Algorithm 2) are used (depicted by squares in Fig. 14).

In this example, as only one problem is considered, $\%SF_{r1} = \%SF_{r2} = \%SF_{r3} = 100\%$ since at least one solution is found within $r1, r2$ and $r3$, and $\%SF_{r4} = 0\%$. As stated in Algorithm 3, since a solution is found in $r1$ and it dominates $r4, r4$ is not explored. Note that the solutions which are depicted within $r4$ were obtained in the first step.

Within the overall solution space, 22 different solutions are found. Of these 22 solutions, 9 are Pareto optimized while 13 are dominated (cf. Definition 11). From the 9 Pareto optimized solutions, 2 solutions belong to $r1$, 4 to $r2$ and 3 to $r3$, which means that $\%PS_{r1} = 22.2\%$, $\%PS_{r2} = 44.4\%$ and $\%PS_{r3} = 33.3\%$. As $r4$ is not solved, i.e., $\%PS_{r4} = 0\%$.

For the holistic design, the data described in Table 4 is quantified (cf. Example 21) for the different non-stochastic SDeclare models which is generated for each considered SDeclare model. In order to measure *Flex*, *MaxRob* and *minRob*, all plans which are generated for each non-stochastic SDeclare model are kept, i.e., the “all plans” policy is applied.

Example 21. Fig. 15 shows the set of solutions which are obtained when solving a specific problem related to the beauty salon scenario. As can be seen, many Pareto optimized solutions (i.e., squares in Fig. 15) are obtained since the problem is solved several times because of the uncertainty variables (i.e., different samples are generated and solved, cf. Section 4.2.1). The maximum and minimum values (i.e., $minWT=3$, $MaxWT=18$, $minP=350$ and $MaxP=730$) as well as their average values (i.e., $\overline{WT}=10.3$ and $\overline{P}=601$) are depicted in each axis.

In addition, the real execution plan is depicted by a circle in Fig. 15. The difference between \overline{WT} (\overline{P}) and $WT = 14.8$ ($P = 515$) in the real execution plan is equal to $\Delta WT = 4.5$ ($\Delta P = 86$). Therefore, applying the proposed approach, the waiting time has been reduced (i.e., $\% \Delta WT = 30.4\%$) in average and the profit has been incremented (i.e., $\% \Delta P = 16.7\%$) in average.

Finally, as depicted, the real enactment plan is dominated by some Pareto optimized solutions (cf., arrows in Fig. 15) which means that solutions which improve both objective functions are found. Therefore, as only one problem is considered in this example, $\%Dominated = 100\%$.

Table 5
Quantified variables for the first experiment (1).

Problem		Unconstrained				Region 1						Region 2					
N	NS	minWT	MaxWT	minP	MaxP	$\%SF_{r1}$	$WT_{r1}(m)$	$P_{r1}(\epsilon)$	$NActs_{r1}$	$\%PS_{r1}$	$SAdd_{r1}$	$\%SF_{r2}$	$WT_{r2}(m)$	$P_{r2}(\epsilon)$	$NActs_{r2}$	$\%PS_{r2}$	$SAdd_{r2}$
10	1	0.1	29.0	273.3	660.0	74.2	14.3	479.0	90.1	10.0	2.5	100.0	19.6	502.4	118.6	38.2	2.7
10	1.5	1.0	31.5	350.5	793.4	63.8	15.2	491.0	98.3	9.2	1.5	100.0	26.2	517.4	128.4	43.5	1.9
10	2	2.1	36.3	492.7	956.3	67.0	17.6	615.2	145.6	11.5	1.8	100.0	26.9	694.8	157.5	44.0	2.2
15	1	3.0	32.4	360.3	782.0	70.5	15.8	515.0	116.9	11.1	2.1	100.0	29.6	521.0	130.0	42.1	2.6
15	1.5	9.7	43.4	538.4	891.5	61.4	22.3	698.4	154.8	9.6	1.4	100.0	34.5	712.4	166.0	39.6	1.8
15	2	8.0	49.6	741.8	1092.1	64.9	24.2	918.4	179.3	9.5	0.9	96.7	39.7	982.4	181.7	42.3	1.4
20	1	15.5	56.3	405.6	859.1	72.6	29.6	625.0	150.5	10.4	1.2	100.0	39.0	621.5	139.5	40.8	1.2
20	1.5	17.8	59.5	715.6	1108.0	59.9	30.9	859.7	172.1	10.5	0.6	96.2	42.9	902.8	180.6	39.1	0.9
20	2	16.5	59.4	950.6	1252.8	45.2	30.1	1105.2	196.0	8.9	1.0	96.2	49.0	1193.6	201.1	47.9	1.1

Table 6
Quantified variables for the first experiment (2).

Problem		Region 3						Region 4					
N	NS	$\%SF_{r3}$	$WT_{r3}(m)$	$P_{r3}(\epsilon)$	$NActs_{r3}$	$\%PS_{r3}$	$SAdd_{r3}$	$\%SF_{r4}$	$WT_{r4}(m)$	$P_{r4}(\epsilon)$	$NActs_{r4}$	$\%PS_{r4}$	$SAdd_{r4}$
10	1	100.0	17.0	340.6	79.5	49.3	1.1	25.8	19.1	413.2	86.6	2.5	1.5
10	1.5	100.0	14.5	365.5	80.1	45.4	0.9	36.2	25.0	451.1	89.0	1.9	1.4
10	2	97.0	15.8	511.3	109.7	42.4	1.4	30.1	24.6	515.2	115.2	2.1	1.4
15	1	100.0	15.1	394.2	83.4	45.3	1.1	29.5	27.7	435.0	89.3	1.5	1.3
15	1.5	96.7	22.1	566.3	129.1	50.1	0.7	37.0	35.4	610.4	138.9	3.6	0.9
15	2	100.0	21.5	735.0	162.0	44.9	0.9	35.1	36.1	779.7	163.6	3.3	1.0
20	1	95.0	26.3	498.7	99.9	46.6	0.8	27.4	31.5	525.4	115.3	2.2	0.9
20	1.5	96.2	27.6	725.9	160.4	46.4	0.7	39.1	40.7	775.9	168.7	4.0	1.1
20	2	100.0	28.4	997.2	191.8	39.1	0.6	52.8	43.4	1004.5	292.5	4.1	0.8

Table 7
Quantified variables for the second experiment.

N	NS	minWT	MaxWT	minP	MaxP	$\overline{WT}(m)$	$\overline{P}(\epsilon)$	$\% \Delta WT$	$\% \Delta P$	$\%Dominated$	\overline{Flex}	minRob	MaxRob
[8, 12]	[1, 1.4]	0.3	26.3	210	1501	7.1	574.2	-26.6	38.3	100	100.0	22.2	44.4
[8, 12]	(1.4, 1.8]	0.1	31.0	305	1561	10.1	681.6	-29.3	32.5	100	100.0	16.7	38.9
[8, 12]	(1.8, 2.2]	3.2	36.5	396	1492	9.9	805.6	-36.8	20.0	100	95.6	22.2	27.8
[13, 17]	[1, 1.4]	3.1	51.6	364	1690	12.2	721.8	-31.2	29.7	100	100.0	25.0	41.7
[13, 17]	(1.4, 1.8]	5.8	59.3	419	1632	18.5	905.0	-27.7	21.8	100	95.6	16.7	33.3
[13, 17]	(1.8, 2.2]	5.1	63.2	538	1681	19.0	984.3	-29.4	16.4	100	95.6	16.7	20.8
[18, 22]	[1, 1.4]	6.0	63.1	515	1877	24.4	851.2	-32.4	21.7	100	88.9	16.7	30.0
[18, 22]	(1.4, 1.8]	7.3	60.7	638	1838	21.5	969.5	-36.8	16.1	100	95.6	13.3	23.3
[18, 22]	(1.8, 2.2]	8.7	69.2	801	1845	30.8	1252.4	-41.3	8.9	100	83.3	13.3	16.7

6.3. Case selection

For this case study, the beauty salon problem is studied. We consider this is a good and suitable case since it fulfills the following selection criteria: (1) it has been created for **an actual business**, (2) the business has grown up and now it has **scheduling problems** (i.e., involves resource allocation, complex constraints and multi-objective optimization), and (3) the problem is **subject to uncertainty**, and such uncertainty can be measured, i.e., the manager can detect the uncertainty of the scenario which can be manually specified in the SDeclare models.

6.4. Case study procedure

The execution of the study is planned as follows.

1. The business is selected according to the selection criteria.
2. The selected business is modeled as a SDeclare model by the business analyst.
3. We proceed differently depending on the case design we are carrying out:
 - In the case of the embedded design, different SDeclare models are randomly generated by varying the data related to the beauty salon problem. Therefore, each generated model includes the same activities, relations and resources, but differs in the number of clients (N), their booked services (S), and their appointment times (T). Considering the information which is provided by the manager of the salon (i.e., there are normally between 10 and 20 clients per day and a client typically books one or two services) we consider values $\{1, 1.5, 2\}$ for the average number of booked services of the clients (i.e., NS) and the values $\{10, 15, 20\}$ for the number of clients (i.e., N). Based on this information, to average the results over a collection of randomly generated SDeclare models, 30 data instances are randomly generated for each pair $\langle N, NS \rangle$ by varying S and T .²⁶
 - In the case of the holistic design, we have extended the model of Fig. 12 to reflect the reality better, i.e., the real model includes 21 different services and 7 resources. In addition, some services are related to each other to prevent non-sense executions, e.g., to avoid performing the cleaning nails service after the painting nails service. Furthermore, the number of services per client is limited to 4.²⁷

As mentioned, this case design considers real data which is obtained from the log of the beauty salon. For this, the staff of the beauty salon manually logged data for a period of 90 days. In particular, for each day they logged: (1) the number of clients (i.e., N), (2) their booked services (i.e., S), (3) their appointment times (i.e., T), and (4) the resource availability of each day ($AvRes$). In turn, the manager of the beauty salon provided some stochastic variables related to T (i.e., some unpunctual clients) and to $AvRes$ (i.e., staff who probably become unavailable during the day). As stated in Section 5, these stochastic variables were defined with flat PMFs (cf. Section 4.1.3). Therefore, for each day, the same SDeclare model is considered (i.e., they have the same activities, relations and estimates) but each day differs in $N, S, T, AvRes$ and the associated stochastic variables (i.e., problem data).²⁸

These problems are grouped considering N and the average

number of booked services (i.e., $NS = |S|/N$) in order to enable the comparison with the embedded design. For this, N is divided into three ranges $[8, 12]$, $[13, 17]$ and $[18, 22]$, and NS is also divided into three ranges, i.e., $[1, 1.4]$, $(1.4, 1.8]$, $(1.8, 2.2]$.

4. After that, the optimized BP enactment plans and the configurable BP models are obtained for the embedded and the holistic design respectively by applying our proposal.
5. All the relevant information is collected following the collection plan.
6. Finally, the analysis and the interpretation of the collected data is conducted and the validity of the case study procedure is studied.

6.5. Data collection

Different data collection plans are conducted depending on the case study design.

1. In the embedded design, for each pair $\langle N, NS \rangle$, the data related to the quantified variables (cf. Table 3) is collected in three phases while generating the optimized BP enactment plans from the SDeclare models. Such phases are detailed as follows. (1) After the initial searches are performed (i.e., Algorithm 2 is executed, the solution space is divided into four regions, and the values of the variables $minWT, MaxWT, minP$ and $MaxP$ are recorded). (2) After such division, $r1$ is the first region to be solved (note that, as mentioned, $r4$ is dominated by $r1$, cf. Section 4.2.4 and Example 20). Then, the data related to $\%SF_{r1}, WT_{r1}, P_{r1}, NActs_{r1}, \%PS_{r1}$, and $SAdd_{r1}$ is stored. (3) Finally, $r2$ and $r3$ are solved. In the case that no solution is found in $r1, r4$ is also solved. Similarly to the previous phase $\%SF_{rx}, WT_{rx}, P_{rx}, NActs_{rx}, \%PS_{rx}$, and $SAdd_{rx}$ are recorded for such regions. Tables 5 and 6 show the values related to all the quantified variables which are involved in the complete data collection plan.
2. In the holistic design, for each pair $\langle N, NS \rangle$, once the configurable BP model is generated, the data related to the quantified variables (cf. Table 4) is recorded, i.e., $minWT, MaxWT, minP, MaxP, \overline{WT}, \overline{P}, \% \Delta WT, \% \Delta P, \% Dominated, Flex, minRob$, and $MaxRob$. The aforementioned values are shown in Table 7.

6.6. Analysis and interpretation

The data which is collected is analyzed to answer the research questions and to draw conclusions.

In order to address question MQ1, sub-questions AQ1 and AQ2 need to be answered (cf. Tables 5 and 6). Regarding the problems which are solved within each region (i.e., columns $\%SF_{rx}$), $\%SF_{r1}$ is lower than both $\%SF_{r2}$ and $\%SF_{r3}$ since $r1$ is the most constrained region. In turn, the low value for $\%SF_{r4}$ can be explained by the fact that solutions are not searched in $r4$ in the case that at least one solution is found in $r1$. In addition, as the complexity of the problem increases (i.e., N and NS increase), $\%SF_{r1}$ decreases and therefore, $\%SF_{r4}$ increases. However, $\%SF_{r2}$ and $\%SF_{r3}$ keep similar values and close to 100%, which means that at least one solution can be found in $r2$ and $r3$ regardless of the complexity of the problems. Therefore, AQ1 can be answered as true as solutions can be found in all regions.

Furthermore, some differences can be observed when analyzing the objective function values (i.e., columns WT and P). As expected, as the complexity of the problems increases, the value of P increases since more services are included in the generated enactment plans. Moreover, with increasing complexity of the problem the value of WT increases as well since the clients are subject to more delays. Note that these columns are also directly dependent

²⁶ The set of problems which are used for the empirical evaluation is available at <http://regula.lsi.us.es/MOPlanner/ObjectsBeautySalon.zip>.

²⁷ The full model of the beauty salon can be found at <http://regula.lsi.us.es/MOPlanner/BeautySalonModel.zip>.

²⁸ The set of data logged which is used for this experiment is available at <http://regula.lsi.us.es/MOPlanner/ObjectsBeautySalonReal.zip>.

on the number of activities of the plan (cf. columns N_{Acts}), i.e., when N_{Acts} increases, P and WT increase too. In general, $r1$ and $r4$ include the most balanced solutions according to the values of both objective functions, while the solutions with the best values for P and WT belong to $r2$ and $r3$ respectively. This distribution is kept independently of the complexity of the problem. Moreover, for all problems Pareto optimized solutions were obtained (cf. columns $\%PS_{rx}$), which means that a representative Pareto front can be depicted. However, $r4$ contributes less to the Pareto front since most of the solutions which were found within $r4$ are dominated by solutions found within $r2$ or $r3$. In turn, $\%PS_{r1}$ also presents low values since $r1$ is the hardest region to be solved (i.e., the most constrained). Furthermore, in all regions $\%PS_{rx}$ seems to be independent of the complexity of the problem. Therefore, although the time spent by Alg. 1 is only based on the $TIME_LIMIT$ constant, it behaves as intended against all the different complexities (i.e., $\%SF_{r2}$, $\%SF_{r3}$ and columns $\%PS_{rx}$ are independent on the complexity of the problems, while WT_{rx} and P_{rx} values are directly dependent on the complexity of the problems), and hence, $AQ2$ can be answered as true.

Finally, for each problem, a relation among the number of additional services per client (cf. columns S_{Add}), P and WT exists. Specifically, as the number of services which are included increases, the profit also increases to the detriment of the waiting time. However, as the problems become more complex, S_{Add} decreases in all the regions since it is more complicated to include more services since more clients and services have to be considered. Considering these values and that $AQ1$ and $AQ2$ are answered as true, we can conclude that $MQ1$ is also true, i.e., the proposed algorithm is suitable for generating a distributed set of Pareto optimized solutions starting from a $S_{Declare}$ model.

In order to address question $MQ2$, sub-questions $AQ3$ and $AQ4$ need to be answered (cf. Table 7). As can be seen, the column \overline{WT} shows that the solutions provided by our approach are shifted to the lowest part of the range $[\min WT, \max WT]$. This means that more solutions were found in the region related to that part than in the other regions. In turn, the values of \overline{P} are more balanced though still being within the lowest part of the range $[\min P, \max P]$. To overcome this issue, the solution space can be divided in more regions in order to get more balanced solutions. Moreover, $\% \Delta WT$ increases as the complexity of the problems increases, which highlights the benefits of using the proposed approach in real cases. The opposite happens with $\% \Delta P$ since the more complex the problem is, the fewer free time slots are available to offer more services. Nonetheless, in all the cases the values $\% \Delta WT$ and $\% \Delta P$ show that our approach improves the mean of both objective functions compared to plans which were manually created (i.e., WT decreases and P increases). Moreover, the solutions which are provided by our approach dominate the associated real plan in all the cases (cf. column $\%Dominated$). That means that, regarding these objective functions, Alg. 1 provides at least one solution which improves both profit and waiting time when compared with the real solutions. Therefore, $AQ3$ is answered as true.

Regarding the flexibility of the generated configurable BP models (cf. column \overline{Flex}), in most cases it achieves 100%, which means that the uncertainty which was specified by the manager is totally covered by the generated models. In fact, the value of \overline{Flex} is over 83.3% even in the most complex problem, which represents a very high degree of flexibility. Regarding the robustness, as the complexity of the problems increases, both upper and lower limits of the robustness decrease (cf. columns $\min Rob$ and $\max Rob$). This is due to the fact that the more activities exist in the plan, the less slack appears. However, these columns present rather good values since a value of 13.3% of robustness means that the related plan will avoid replanning in this 13.3% of cases. Then, $AQ4$ and consequently $MQ2$ are answered as true.

Comparing both embedded and holistic designs, we can say that using both randomly generated data and real data from a log a similar behavior is observed in the bounds of the objective function values (cf. columns $\min WT$, $\max WT$, $\min P$ and $\max P$ in Tables 5 and 7). However, the upper bounds in the holistic design (cf. columns $\max WT$ and $\max P$ in Table 7) tend to be slightly different since the $S_{Declare}$ model has been extended in this experiment (i.e., more services, more resources and uncertainty are included). The size of the ranges of the waiting time (i.e., differences between columns $\min WT$ and $\max WT$) increases as the average number of services per client increases (i.e., column NS). Such dependency is due to the fact that when NS is high, including additional services involves more complex schedules and therefore, it increases the waiting time more than when NS is low. However, the opposite happens with the size of the ranges of the profit. This is the expected behavior since the simpler the problems are (i.e., the lower values of NS), the more chances to include services exist. Therefore, the approach behaves similarly against both sources of problems.

6.7. Validity evaluation

This section evaluates if the results are valid and not biased. Three types of validity are addressed in this section: construct, internal and external.

Firstly, with relation to the construct validity, it has to be addressed in how far the measures which have been used are appropriate to address the research questions which have been planned. Three different threats are identified related to the acquisition of the data. The first threat is related to how the problems have been randomly generated in the embedded design. In this design, unsolvable problems were not considered in order to evaluate the algorithm better. This is checked considering a simple rule: the generated appointment time of a client plus the time which her booked services consume cannot overpass the closing time of the beauty salon. Due to the parallelism which may exist because of the temporal constraints (i.e., a client can be served by different employees at the same time), this rule leaves out some problems which might be solvable. To mitigate this threat, a more elaborated algorithm can be performed to avoid eliminating problems which may be solvable. Secondly, the complexity of the problems which are generated in the embedded design is controlled only by varying the number of clients and her booked services. Although we consider that the beauty salon is a suitable business due to its complexity, different ways of controlling this complexity can be applied to mitigate this threat, e.g., by changing the type of constraints. The third threat concerns the duration of the logged data in the holistic design (90 days). To the best of our knowledge there is no metric which states how long data must be logged to obtain a meaningful log. To mitigate this threat, longer durations can be considered to get more data and therefore to increase the probability of finding situations where the algorithm does not perform well. Finally, in both designs some values have been fixed in the algorithms, i.e., the number of regions into which the solution space is divided (fixed to 4) and the number of samples which are generated in the holistic design (fixed to 30). Increasing these values would increase the accuracy of the proposal.

Regarding the internal validity, the main threat is that the obtained results related to flexibility and robustness concerns could be biased since their interpretation can be subjective since it depends on the business which is analyzed. To mitigate it, other business experts can be consulted in order to state what is a successful value for flexibility and robustness.

Finally, the external validity considers in how far the obtained results could be generalized to any business. This generalization is threatened by the fact that the beauty salon was the unique

scenario which was studied. Other scenarios can be considered to replicate this study in order to mitigate this threat.

7. Discussion and limitations

The current approach allows modeling the considered problems in an easy way, since the considered declarative specifications (i.e., an extension of the Declare language [63]) are based on high-level constraints. Moreover, with the proposed extension, the expressiveness of the process designs is enhanced compared to [6,43,44] (e.g., stochastic values for modeling the uncertainty of the scenario can be included), and hence more realistic problems can be managed, e.g., the Beauty Salon detailed in Section 5. Such scenario has been selected since: (1) it faces a complex problem which goes far beyond a toy problem since it presents several challenges that can also be found in other domains (i.e., it highly depends on the manager's skill to take decisions, uncertainty is inherent to the business and the declarative language ameliorate it specification), (2) unlike more common scenarios, this kind of business has not been widely supported by previous research and thus, we considered it as an innovative application, and (3) in that scenario we had access to the data and therefore, the analyst was able to collect data for a long period of time for the evaluation. Therefore, our approach is intended to reduce the human work in scenarios with high variability in various ways:

- Since declarative BP model specifications allow their users to specify what has to be done instead of how [63] and the tacit nature of human knowledge is often an obstacle to eliciting accurate process models [29], declarative specifications facilitate the human work which is involved in the process design and analysis phase compared to imperative specifications. Specifically, using a declarative specification, the user only has to define the constraints of her models without being aware of how they are fulfilled. Therefore, several ways of executing such declarative model exist. In turn, imperative specifications entail more complexity since all the possible execution alternatives need to be specified. Such complexity is even higher when a high flexibility is required, in the presence of input uncertainty, or when the resources need to be allocated in a suitable way considering the optimization of certain objective functions.
- Typically, executing a declarative model (which presents high variability) usually entails bigger effort for the involved users compared to executing an imperative model [66,79] since deciding how to exactly execute the process is difficult for the user and this can lead to bad executions (i.e., very bad values for some objective functions). For this, our approach extracts a desirable part of the variability of a declarative model through the generation of multi-objective optimized enactment plans while discarding bad execution alternatives (according to the optimization of some objective functions as well as high flexibility and robustness). This way, the proposed approach facilitates the human work which is involved in the process enactment phase.
- Such optimized plans are then merged into a configurable BP model which supports the analysts in the management of these plans and helps the analysts in understanding what the different plans share, what their differences are, and why and how these differences occur [71].

Since these kind of problems are NP-complete, getting optimal solutions cannot be ensured in general (this is the reason why we use the term optimized plans instead of optimal plans). This way, the quality of the solution which is calculated depends on the time limit which is established in the search algorithm. Note that,

as mentioned, efficient filtering rules have been developed. Despite the NP-complexity of the considered problems, such filtering rules have demonstrated their effectiveness for improving performance in previous works [6,7,9].

In addition, to further improve the quality of the resulting execution alternatives regarding the uncertainty of the scenario, flexibility and robustness concerns are considered. Unlike [3,18,20,34,37,79,86,89], the current work proposes quantitative definitions in order to measure how the uncertainty of the scenario is supported by each generated enactment plan. Although some of these proposals [3,86,89] do not apply these terms to BPs, they were considered when developing the definitions of both flexibility and robustness which are provided in this paper. Therefore, execution alternatives that are not desirable for the business regarding both the quality and a set of given objective functions are avoided.

Furthermore, in contrast to related proposals [29,47,53,75,94], not only a single enactment plan but a set of optimized enactment plans are considered when generating the imperative model. This way, the flexibility of the resulting imperative model is not unnecessarily restricted.

The optimized plans which are included in the generated configurable BP model can be used, as discussed in previous works, for (1) assisting users during the process execution to optimize performance through recommendations [8,9], (2) providing predictions, e.g., predict the completion time of all the running instances [7] or (3) performing simulations, e.g., *what-if* scenarios [7].

Moreover, the automatic generation of optimized configurable BP model can deal with complex and real problems in a simple way as demonstrated in Section 6. Therefore, a wide study of several aspects can be carried out by simulation. Nonetheless, an evaluation with more complex scenarios is required to improve the generalizability of the results and is planned for future work.

However, our approach also presents a few limitations. In general, different resource patterns can be taken into account. In the proposed approach, motivated by the considered scenarios, we consider that a resource can only perform an activity at the same time (i.e., the same resource cannot be used to perform more than one activity in parallel) and that activities are executed without preemption. The business analysts must deal with a new language for the constraint-based specification of BPs, thus a period of training is required to let them become familiar with the proposed language, i.e., SDeclare.²⁹ Furthermore, the optimized configurable BP model is generated by considering estimated values for the number of instances to execute, and hence our proposal is only appropriate for processes in which this number is known a priori. As a real example, the beauty salon problem is detailed and an extensive empirical evaluation is carried out with the goal of supporting the contributions of our proposal. Some of our previous works also dealt with scenarios in which the number of process instances to be executed in a specific timeframe is known a priori (e.g., [7] describes a travel agency problem and [10] considers computer support for clinical guidelines as an application example). In a related way, activity attributes and resource availability need to be estimated. For this, our approach allows for including uncertainty regarding these attributes through well-specified functions, i.e., the user has to define a PMF for each uncertain variable. Unlike [57], our approach allows for assigning any PMF to any stochastic attribute (i.e., the PMFs are not restricted to time attributes). Although PMFs allow us to face the uncertainty of the considered scenario (i.e., the beauty salon), other approaches allow to specify the uncertainty through poorly-specified functions (e.g., using fuzzy techniques) [24,91] which may be required for dealing with scenarios having different characteristics.

²⁹ To support the graphical specification of SDeclare models the existing Declare tool (available at <http://www.win.tue.nl/declare/>) has been adapted for allowing resource specification, temporal and data constraints, as well as stochastic estimates.

Although the stochastic feature of SDeclare facilitates to specify the estimates, the problem complexity increases as the number of stochastic variables increases. However, if the actual values deviate from the estimated values during the execution of the model, P&S techniques can be applied to replan the activities at runtime by considering the actual values of the estimates, as discussed in a previous work [7].

In addition, motivated by the requirements of the considered scenarios, the data perspective which is considered in the current approach mainly includes data constraints which can be applied to input data and activity relations. However, more advanced features like dynamic data or data-flow perspective have been left out since they are not part of the design requirements of the considered scenarios and will be addressed in future work when applying our proposal to BPs with different characteristics.

8. Conclusions and future work

In this work, a method for generating optimized configurable BP models from declarative specifications is proposed to facilitate the human work which is involved in the process design & analysis phase and to improve the resulting imperative BP models in scenarios subject to uncertainty. For this, a declarative specification is used as starting point. Typically, such specifications enable different ways of executing it while reaching certain objectives. For this, the variability of the initial declarative specification is reduced to a desirable variability (i.e., those variants which present worst values for the objective functions or which do not contribute to the flexibility are removed) through the generation of multi-objective optimized enactment plans. This process is done automatically using a constraint-based approach for planning and scheduling the process activities. For managing uncertainty, flexibility and robustness concerns are considered as key aspects of the current approach. From such set of multi-objective optimized enactment plans, those alternatives which are too strict or which only withstand an extent of the uncertainty which is already withstood by another alternative are discarded. For managing such uncertainty, flexibility and robustness concerns are considered as key aspects of the current approach. Typically the enactment plans which are kept after such filtering process share many commonalities, and hence, a configurable BP model can be created by merging such plans. The configurable BP models which are generated support the analysts managing the set of optimized plans and understanding what the different plans share, what their differences are, and why and how these differences occur. Note that although the proposed approach is focused on the control-flow and the resource perspectives, the data perspective is also partially addressed.

The proposed approach is applied to a real scenario and is validated through the analysis of different performance measures related to such real scenario. Results indicate that, despite the NP complexity of the considered problems, the plans which are generated through the proposed approach improve the actual results provided by the business in most cases and, in addition, the generated configurable BP models support most of the variability required by the manager of the business.

Regarding previous related work, on the one hand, we are not aware of any other approach for generating configurable BP models from declarative specifications. Although there exist some proposals which could be extended in such direction, most of such approaches neither consider the optimization of multiple objective functions nor the resource perspective. In addition, unlike the proposed approach, none of such approaches considers the uncertainty of the scenario through stochastic attributes. On the other hand, although several approaches exist for dealing with flexibility issues, in this paper as a novel contribution, we provide

quantitative definitions for both robustness and flexibility which allow us to measure how the uncertainty of a real scenario is supported by an enactment plan and by a configurable BP model respectively.

As for future work, we will explore various constraint-based solving techniques as well as different multi-objective optimization techniques like stochastic algorithms. Additionally, we intend to consider further resource patterns. To conclude, an extended empirical study is intended to be carried out with the goal of measuring the impact of the robustness and flexibility on the process performance in different real businesses with more variety of activities, resources and sources of uncertainty, e.g., scenarios from information processing or manufacturing processes. For this, the tool related to the proposed approach still needs some refactoring to make it ready to use by others and thus, to make it publicly available.

Acknowledgements

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] D.A. Aaker, B. Mascarenhas, The need for strategic flexibility, *California Manage. Rev.* 5 (2) (1994) 72–89.
- [2] S. AbouRizk, D. Halpin, J. Wilson, Fitting beta distributions based on sample data, *J. Construct. Eng. Manage.* 120 (2) (1994) 288–305.
- [3] H. Aissi, B. Roy, Robustness in multi-criteria decision aiding, in: Matthias Ehrgott, Jos Rui Figueira, Salvatore Greco (Eds.), *Trends in Multiple Criteria Decision Analysis*, International Series in Operations Research & Management Science, vol. 142, 2010, pp. 87–121.
- [4] T.W. Athan, P.Y. Papalambros, A note on weighted criteria methods for compromise solutions in multiobjective optimization, *Eng. Optim.* 27 (2) (1996) 155–176.
- [5] W. Back, W. Boles, G. Fry, Defining triangular probability distributions from historical cost data, *J. Construct. Eng. Manage.* 126 (1) (2000) 29–37.
- [6] I. Barba, C. Del Valle, A constraint-based approach for planning and scheduling repeated activities, in: *Proc. COPLAS*, 2011, pp. 55–62.
- [7] I. Barba, C. Del Valle, B. Weber, A. Jimenez-Ramirez, Automatic generation of optimized business process models from constraint-based specifications, *Int. J. Cooperat. Inform. Syst.* 22 (02) (2013).
- [8] I. Barba, B. Weber, C. Del Valle, Supporting the optimized execution of business processes through recommendations, in: *Proc. BPI*, 2011.
- [9] I. Barba, B. Weber, C. Del Valle, A. Jimenez-Ramirez, User recommendations for the optimized execution of business processes, *Data Knowl. Eng.* 86 (0) (2013) 61–84.
- [10] Irene Barba, Andreas Lanz, Barbara Weber, Manfred Reichert, Carmelo Del Valle, Optimized time management for declarative workflows, in: Ilia Bider, Terry Halpin, John Krogstie, Selmin Nurcan, Erik Proper, Rainer Schmidt, Phina Soffer, Stanisław Wrycza (Eds.), *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing, vol. 113, Springer, Berlin Heidelberg, 2012, pp. 195–210.
- [11] R. Bartak, O. Cepek, Incremental propagation rules for a precedence graph with optional activities and time windows, *Trans. Inst. Meas. Control* 32 (1) (2010) 73–96.
- [12] Business Process Model and Notation (BPMN), Version 2.0, 2011. <<http://www.omg.org/spec/BPMN/2.0/>> (accessed 01.05.14).
- [13] Pearl Brereton, Barbara Kitchenham, David Budgen, Using a protocol template for case study planning, in: *Proceedings of EASE 2008*, BCS-eWiC, 2008.
- [14] P. Brucker, S. Knust, *Complex Scheduling (GOR-Publications)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [15] J.M. Carlson, John Doyle, Complexity and robustness, *Proc. Natl. Acad. Sci. USA* 99 (Suppl. 1) (2002) 2538–2545.
- [16] V. Chankong, Y. Haimes, *Multiobjective Decision Making Theory and Methodology*, Elsevier, 1983.
- [17] Federico Chesani, Evelina Lamma, Paola Mello, Marco Montali, Fabrizio Riguzzi, Sergio Storari, Exploiting inductive logic programming techniques for declarative process mining, in: *Transactions on Petri Nets and Other Models of Concurrency II*, Springer, 2009, pp. 278–295.
- [18] Serafino Cicerone, Gabriele Di Stefano, Michael Schachtebeck, Anita Schöbel, Multi-stage recovery robustness for optimization problems: a new concept for planning under disturbances, *Inform. Sci.* 190 (2012) 107–126.
- [19] I. Das, J.E. Dennis, A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems, *Struct. Multidiscipl. Optim.* 14 (1997) 63–69.

- [20] J. de Haan, J.H. Kwakkel, W.E. Walker, J. Spirco, W.A.H. Thissen, Framing flexibility: theorising and data mining to develop a useful definition of flexibility and related concepts, *Futures* 43 (9) (2011) 923–933. Special Issue: Flexible infrastructures.
- [21] Kalyanmoy Deb, Deb Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [22] W.M.P. der Aalst, A. Dreiling, F. Gottschalk, M. Rosemann, M.H. Jansen-Vullers, Configurable process models as a basis for reference modeling, in: Christoph. Bussler, Armin Haller (Eds.), *Business Process Management Workshops, Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, 2006, pp. 512–518.
- [23] P. Dourish, J. Holmes, A. MacLean, P. Marquardsen, A. Zbyslaw, Freeflow: mediating between representation and action in workflow systems, in: *Proc. CSCW*, 1996, pp. 190–198.
- [24] H. Fargier Dubois, H. Prade, Using fuzzy constraints in job-shop scheduling, in: *Proc. of IJCAI'93/SIGMAN Workshop on Knowledge-based Production Planning, Scheduling and Control*, 1993.
- [25] Dynadec, Comet Downloads, 2013. <<http://dynadec.com/support/downloads/>> (accessed 02.09.13).
- [26] M. Ehrgott, X. Gandibleux, Multiobjective combinatorial optimization – theory, methodology, and applications, in: *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys, International Series in Operations Research & Management Science*, vol. 52, 2003, pp. 369–444.
- [27] D. Jan Eppink, Planning for strategic flexibility, *Long Range Plan.* 11 (4) (1978) 9–15.
- [28] J. Fente, C. Schexnayder, K. Knutson, Defining a probability distribution function for construction simulation, *J. Construct. Eng. Manage.* 126 (3) (2000) 234–241.
- [29] H.M. Ferreira, D.R. Ferreira, An integrated life cycle for workflow management based on learning and planning, *Int. J. Cooper. Inform. Syst.* 15 (4) (2006) 485–505.
- [30] G. Leitmann, Some problems of scalar and vector-valued optimization in linear viscoelasticity, *IEEE Trans. Automat. Control* 23 (1977) 93–99.
- [31] H.L. Gantt, *Work, Wages, and Profits*, Engineering Magazine Co., 1913.
- [32] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman & Co., New York, NY, USA, 1979.
- [33] M. Ghallab, D. Nau, P. Traverso, *Automated Planning: Theory and Practice*, Morgan Kaufman, Amsterdam, 2004.
- [34] William Golden, Philip Powell, Towards a definition of flexibility: in search of the holy grail?, *Omega* 28 (4) (2000) 373–384
- [35] F. Gottschalk, W.M.P. van der Aalst, M.H. Jansen-Vullers, M. La Rosa, Configurable workflow models, *Int. J. Cooper. Inform. Syst.* (IJCIS) 17 (2) (2008) 177–221.
- [36] Gartner Group, Leading in times of transition: the 2010 CIO agenda (EXP Premier Report No. January 2010), in: Report, Gartner, Inc., 2010.
- [37] Stefan Gueorguiev, Mark Harman, Giuliano Antoniol, Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering, in: *Proceedings of the 11th Annual conference on Genetic and evolutionary computation, GECCO '09*, New York, NY, USA, 2009, pp. 1673–1680.
- [38] Y.Y. Haimes, L.S. Lasdon, D.A. Wismer, On a bicriterion formulation of the problems of integrated system identification and system optimization, *IEEE Trans. Syst. Man Cybern.* 1 (1971) 296–297.
- [39] A. Hallerbach, T. Bauer, M. Reichert, Capturing variability in business process models: the provop approach, *J. Softw. Maintenance Evol.: Res. Pract.* 22 (6–7) (2010) 519–546.
- [40] W. Hummer, P. Gaubatz, M. Strembeck, U. Zdon, S. Dustdar, Enforcement of entailment constraints in distributed service-based business processes, *Inform. Softw. Technol.* 55 (11) (2013) 1884–1903.
- [41] M.T. Jensen, Improving robustness and flexibility of tardiness and total flowtime job shops using robustness measures, *J. Appl. Soft Comput.* 1 (2001) 35–52.
- [42] M.T. Jensen, Generating robust and flexible job shop schedules using genetic algorithms, *IEEE Trans. Evol. Comput.* 7 (3) (2003) 275–288.
- [43] A. Jimenez-Ramirez, I. Barba, C. Del Valle, B. Weber, OptBPPlanner: automatic generation of optimized business process enactment plans, in: *Proc. ISD*, Springer, 2012, pp. 429–442.
- [44] A. Jimenez-Ramirez, I. Barba, C. Del Valle, B. Weber, Generating multi-objective optimized business process enactment plans, in: Camille Salinesi, MoiraC. Norrie, scar Pastor (Eds.), *Advanced Information Systems Engineering, Lecture Notes in Computer Science*, vol. 7908, Springer, Berlin Heidelberg, 2013, pp. 99–115.
- [45] Azharul Karim, Kazi Arif-Uz-Zaman, A methodology for effective implementation of lean strategies and its performance evaluation in manufacturing organizations, *Business Process Manage. J.* 19 (1) (2013) 169–196.
- [46] J. Koski, Defectiveness of weighting method in multicriterion optimization of structures, *Commun. Numer. Methods Eng.* 1 (6) (1985) 333–337.
- [47] R. Krogt, J. Geraghty, M.R. Salman, J. Little, On supporting lean methodologies using constraint-based scheduling, *J. Schedul.* 13 (2010) 301–314.
- [48] Akhil Kumar, Wen Yao, Design and management of flexible process variants using templates and rules, *Comput. Indus.* 63 (2) (2012) 112–130. *Managing Large Collections of Business Process Models Managing Large Collections of Business Process Models*.
- [49] Marcello La Rosa, Marlon Dumas, Arthur HM ter Hofstede, Modelling business process variability for design-time configuration, *Handbook of Research on Business Process Modeling*, 2008.
- [50] P. Laborie, J. Rogerie, P. Shaw, P. Vilim, Reasoning with conditional time-intervals. Part ii: An algebraical model for resources, 2009.
- [51] Evelina Lamma, Paola Mello, Marco Montali, Fabrizio Riguzzi, Sergio Storari, Inducing declarative logic-based models from labeled traces, in: *Business Process Management*, Springer, 2007, pp. 344–359.
- [52] L. Thiele, E. Zitzler, M. Laumanns, An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method, *Eur. J. Oper. Res.* 169 (3) (2006) 932–942.
- [53] R. Lu, S. Sadiq, G. Governatori, On managing business processes variants, *Data Knowl. Eng.* 68 (7) (2009) 642–664.
- [54] R. Lu, S. Sadiq, V. Padmanabhan, G. Governatori, Using a temporal constraint network for business process execution, in: *Proc. ADC*, 2006, pp. 157–166.
- [55] L.T. Ly, S. Rinderle, P. Dadam, Integration and verification of semantic constraints in adaptive process management systems, *Data Knowl. Eng.* 64 (1) (2008) 3–23.
- [56] C. Maio, C. Schexnayder, K. Knutson, S. Weber, Probability distribution functions for construction simulation, *J. Construct. Eng. Manage.* 126 (4) (2000) 285–292.
- [57] M.A. Marsans, G. Balbo, G. Conte, *Modelling with Generalized Stochastic Petri Nets*, Wiley Series in Parallel Computing, New York, NY, USA, 1995.
- [58] J. Mendling, G. Neumann, W.M.P. van der Aalst, Understanding the occurrence of errors in process models based on metrics, *Lect. Notes Comput. Sci., LNCS* 4803 (2007) 113–130.
- [59] Achille Messac, Cyriaque Puemi-Sukam, Emanuel Melachrinoudis, Aggregate objective functions and pareto frontiers: required relationships and practical implications, *Optim. Eng.* 1 (2000) 171–188.
- [60] 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.
- [61] Marco Montali, Federico Chesani, Paola Mello, Fabrizio M. Maggi, Towards data-aware constraints in declare, in: *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, 2013, pp. 1391–1396.
- [62] Marco Montali, Maja Pesic, Wil M.P. van der Aalst, Federico Chesani, Paola Mello, Sergio Storari, Declarative specification and verification of service choreographies, *ACM Trans. Web* 4 (1) (2010) 3:1–3:62.
- [63] M. Pesic, Constraint-Based Workflow Management Systems: Shifting Control to Users, PhD thesis, Eindhoven University of Technology, Eindhoven, 2008.
- [64] M. Pesic, M.H. Schonenberg, N. Sidorova, W.M.P. van der Aalst, Constraint-based workflow models: change made easy, in: *OTM Conferences* (1), 2007, pp. 77–94.
- [65] Process Specification Language project, 1977. <<http://www.nist.gov/psl/>> (accessed 01.05.14).
- [66] Manfred Reichert, Barbara Weber, *Enabling Flexibility in Process-Aware Information Systems*, Springer, Berlin, Heidelberg, 2012.
- [67] H. Reijers, *Design and Control of Workflow Processes*, Springer-Verlag, Berlin, Heidelberg, 2003.
- [68] Marcello La Rosa, Marlon Dumas, Reina Kaarik, Remco M. Dijkman, Merging business process models, in: *18th International Conference on Cooperative Information Systems (CoopIS '10) 2010*, Springer-Verlag Heidelberg, Crete, Greece, 2010, pp. 96–113.
- [69] Marcello La Rosa, Marlon Dumas, Arthur H.M. ter Hofstede, Jan Mendling, Configurable multi-perspective business process models, *Inform. Syst.* 36 (2) (2011) 313–340. Special Issue: Semantic Integration of Data, Multimedia, and Services.
- [70] Marcello La Rosa, Marlon Dumas, Reina Uba, Remco M. Dijkman, Business process model merging: an approach to business process consolidation, in: *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2012.
- [71] M. Rosemann, W.M.P. van der Aalst, A configurable reference modeling language, *Inform. Syst.* 32 (2007) 1–23.
- [72] F. Rossi, P. van Beek, T. Walsh (Eds.), *Handbook of Constraint Programming*, Elsevier, 2006.
- [73] N. Russell, W.M.P. van der Aalst, A.H.M. ter Hofstede, D. Edmond, Workflow resource patterns: identification, representation and tool support, in: *Proc. CAiSE*, 2005, pp. 216–232.
- [74] Irina Rychkova, Gil Regev, Alain Wegmann, Using declarative specifications in business process design, *IJCSA* 5 (3b) (2008) 45–68.
- [75] S.W. Sadiq, M.E. Orlowska, W. Sadiq, Specification and validation of process constraints for flexible workflows, *Inform. Syst.* 30 (5) (2005) 349–378.
- [76] M.A. Salido, Introduction to planning, scheduling and constraint satisfaction, *J. Intell. Manuf.* 21 (1) (2010) 1–4.
- [77] Arnd Schnieders, Frank Puhlmann, Variability mechanisms in e-business process families, in: *Proc. International Conference on Business Information Systems (BIS 2006)*, 2006, pp. 583–601.
- [78] Helen Schonenberg, Ronny Mans, Nick Russell, Nataliya Mulyar, Wil van der Aalst, Process flexibility: a survey of contemporary approaches, in: *Advances in Enterprise Engineering I*, Springer, 2008, pp. 16–30.
- [79] Helen Schonenberg, Barbara Weber, Boudewijn Dongen, Wil Aalst, Supporting flexible processes through recommendations based on history, in: Marlon Dumas, Manfred Reichert, Ming-Chien Shan (Eds.), *Business Process Management, Lecture Notes in Computer Science*, vol. 5240, Springer, Berlin Heidelberg, 2008, pp. 51–66.

- [80] Dennis M.M. Schunselaar, Fabrizio Maria Maggi, Natalia Sidorova, Wil M.P. Aalst, Configurable declare: designing customisable flexible process models, in: Robert Meersman, Herv Panetto, Tharam Dillon, Stefanie Rinderle-Ma, Peter Dadam, Xiaofang Zhou, Siani Pearson, Alois Ferscha, Sonia Bergamaschi, Isabel F. Cruz (Eds.), *On the Move to Meaningful Internet Systems: OTM 2012*, Lecture Notes in Computer Science, vol. 7565, Springer, Berlin Heidelberg, 2012, pp. 20–37.
- [81] Rachna Shah, Peter T Ward, Lean manufacturing: context, practice bundles, and performance, *J. Oper. Manage.* 21 (2) (2003) 129–149.
- [82] Pradyumn Kumar Shukla, Kalyanmoy Deb, On finding multiple pareto-optimal solutions using classical and evolutionary generating methods, *Eur. J. Oper. Res.* 181 (3) (2007) 1630–1652.
- [83] K.I. Smith, R.M. Everson, J.E. Fieldsend, C. Murphy, R. Misra, Dominance-based multiobjective simulated annealing, *IEEE Trans. Evol. Comput.* 12 (3) (2008) 323–342.
- [84] M. Souki, Operating theatre scheduling with fuzzy durations, *J. Appl. Oper. Res.* 3 (2011) 177–191.
- [85] W. Stadler, Caveats and boons of multicriteria optimization, *Comput.-Aid. Civil Infrastruct. Eng.* 10 (4) (1995) 291–299.
- [86] Mark Stevenson, Martin Spring, Flexibility from a supply chain perspective: definition and review, *Int. J. Oper. Product. Manage.* 27 (7) (2007) 685–713.
- [87] Balram Suman, Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem, *Comput. Chem. Eng.* 28 (9) (2004) 1849–1871.
- [88] C. Sun, M. Aiello, Towards variable service compositions using vxbpel, in: *Proceedings of the 10th International Conference on Software Reuse: High Confidence Software Reuse in Large Systems, ICSR '08*, 2008, pp. 257–261.
- [89] David Upton, *The management of manufacturing flexibility*, *California Manage. Rev.* 36 (2) (1994) 72–89.
- [90] W.M.P. van der Aalst, A.H.M. ter Hofstede, M. Weske, Business process management: a survey, in: *Proc. BPM*, 2003, pp. 1–12.
- [91] L.A.E. Vullings, C.A. Blok, C.G.A.M. Wessels, J.D. Bulens, Dealing with the uncertainty of having incomplete sources of geo-information in spatial planning, *Appl. Spatial Anal. Policy* 6 (1) (2013) 25–45.
- [92] J. Wainer, F. De Lima Bezerra, Constraint-based flexible workflows, in: *Proc. CRIWG*, 2003, pp. 151–158.
- [93] Mathias Weske, *Business Process Management: Concepts, Languages, Architectures*, Springer, 2007.
- [94] M. Westergaard, F.M. Maggi, Looking into the future: using timed automata to provide a priori advice about timed declarative process models, in: *International Conference on Cooperative Information Systems (CoopIS 2012)*, 2012.
- [95] M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, New York, 1982.
- [96] Shlomo Zilberstein, Using anytime algorithms in intelligent systems, *AI Mag.* 17 (3) (1996) 73–83.