On the Dynamic Configuration of Business Process Models

Abstract. Business Process Models are a relevant input for the development of information systems. Since processes are performed in increasingly dynamic business environments, the processes are required to be flexible and dynamic as well, adapting to environmental changes. Thus, it is essential to properly represent variability in Business Process Models. Moreover, in order to allow for adaptive and autonomic systems, it is of paramount importance to reason on the variability of a process, being able to select an optimal process configuration for a given context. In this paper, we present an approach for such context-aware reasoning, on which the business process configuration is driven by Non-Functional Requirements. Using independent models for expressing variability representation, configuration knowledge, contextual information, and the process itself, we present algorithms and mechanisms to perform business process configuration at runtime, without requiring human intervention. Furthermore, we describe experiments we conducted in order to assess the suitability of our approach.

Keywords: Business Process Configuration, Non-Functional Requirements, Adaptive and Flexible Information Systems, Context-Aware Information Systems.

1 Introduction

Business Process Management (BPM) is a systematic and structured approach to analyze, improve, control and manage processes with the aim of improving the quality of products and services [1]. In organizations that adopt BPM, the business process models play a central role by capturing the way activities are performed. The processes are becoming increasingly complex and heterogeneous, as they often include activities of different nature involving people, software, and hardware placed in diverse physical surroundings. Moreover, some application domains that are influenced by environmental, geographical and human factors, such as logistics and transportation (e.g., airline companies), have to maintain their processes updated and valid in order to keep running their business properly. More than just changing them, the companies need to be aware of quality constraints that affect their business processes (e.g., security, reliability, performance, and so on).

By involving elements of different nature, the processes are increasingly dynamic and therefore more prone to changes [2]. Hence, the flexibility in business process is vital in order to support this heterogeneity. The business process models adapt to changes by providing a description of (i) the parts that can be modified in the process, (ii) the criteria that drive the modifications, and (iii) the mechanisms used to perform adaptation. Business process flexibility can be achieved by several methods [3–6], usually providing ways to represent the variability of business processes and means to perform the configuration of processes to obtain new instances. However, the configuration of business process models currently relies on human experts, such as business analysts, which are often expensive and not always available. In highly dynamic and complex environments that require immediate adaption, this is no longer acceptable. For example, an emergency, such as the volcanic ash cloud crises which massively disrupted air traffic in Europe in 2010 and Latin America in 2011, calls for immediate intervention.

Several works represent variability in business processes [3, 4, 7] and propose mechanisms to modify the process models according to the situation. However, they often lack the necessary guidance to become adaptable to a given context, for example the closure of the air space above a certain height or due to bad weather. In industrial settings the configuration is usually performed on an ad hoc basis, guided solely by the analyst's experience. However, in more dynamic environments, the changes have to be performed more frequently and systematically. Moreover, current approaches that guide the configuration of process models usually just consider high level quality constrains such as cost and performance. Other important quality attributes that could affect the business process, such as security and availability, are seldom taken in account.

In previous works we introduced our approach to deal with business process variability and its configuration using NFRs [5] and contextual information [6]. We investigated how to obtain configurations for business process models that are aware of contextual changes and that meet stakeholders preferences over non-functional requirements. We have proposed a configuration process that relies on contextual information to identify change opportunities. We also claim that Non-Functional Requirements (NFR) [8] can define important constraints that the business process must comply to. Hence, we advocate the use of NFR as qualitative criteria to drive the configuration of business process models and the application context-awareness in order to deal with changes in the environment.

In this paper we break new grounds and present novel contributions. First, we define the metamodel of the proposal based in the conceptual model presented in [6]. The metamodel incorporates a detailed description of the modeling elements including new connections and the linking with another metamodel. It is very important to describe a modeling language and the correspondent tool support. Moreover, it also allows the definition of constraints using a specific language (i.e., OCL). We also improved some steps of our configuration process to include the algorithms used to perform the configuration/generation of process models. More precisely, the last step of our process is detailed to explain the computation necessary to select a configuration and generate a new process model. Last but not least, we present an assessment of our approach using a simulation of business process models execution.

The remainder of this paper is structured as follows. Section 2 introduces the background information and some basic concepts. Section 3 presents our approach. An assessment of our approach is presented in Section 4. We compare our proposal to some related works in Section 5. Finally, in Section 6 we conclude the paper.

2 Conceptual Framework and Background

Our proposal concerns business process configuration, using the notions of NFR and contextual information. We rely on independent models for expressing variability representation, NFRs, contextual information, and the process itself. In the following subsections we present the background on these topics. In Figure 1 we depict the metamodel of our approach. It includes the main concepts used in our approach such as Variants, Variation Point, NFR and Context. Moreover, it also describes the relationships among them and the way how our model is linked to the business process model. Hence, it consists of classes that have attributes and can be linked through association, aggregation and inheritance relationships. The BPMN model is represented using the Eclipse BPMN 2.0 meta-model (package in dark grey). The metamodel was described using the Ecore Language modeling facility of Eclipse modeling framework. The *WorkflowPattern*, *ContributionType*, and *VariationPointOperator* elements are of enumeration type, i.e. they describe a set of values that the attribute of these types can assume. Some relationships are too complex to be expressed by the modeling language. In these cases we have used the OCL language to express constraints and derivations. For example, the constraint involving *VariantsRelationship* and *Variant* classes, which states that a variant cannot requires or excludes itself was expressed in OCL. Due to space limitation we omit the OCL constraints in this paper.

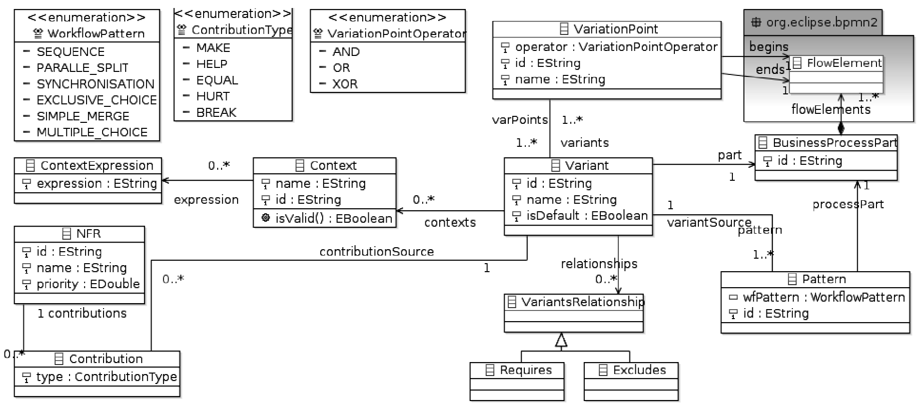

Fig. 1. The Metamodel for our Approach

As a motivating example, let us consider check-in (see Fig. 2) related processes in the airport domain. Check-in is usually the first procedure for a passenger when arriving at an airport, as airline regulations require passengers check-in by certain times prior to the departure of a flight. This duration usually spans from 30 min to 4 h depending on the destination and airline. During this process, the passenger has the ability to ask for special accommodations such as seating preferences, inquire about flight or destination information, make changes to reservations, accumulate frequent flyer program miles, or pay for upgrades. The airline check-in’s main function, however, is to accept luggage that is to go in the aircraft's cargo hold.

Several activities related to airport check-in process are relevant and useful for the proper definition of appropriate business process models (such as passenger check-in policies and procedures regarding security, luggage handling, passenger handling and document validation). Check-in options and procedures vary according to the airline as some airlines allow certain restrictions that other carriers have in place, and occasionally the same airline at two separate airports may have different check-in procedures.

These types of process runs in dynamic environment, since the processes may be affected by several factors such as load of passengers, security policies, weather and so on. In the above scenario, it could be helpful to configure the process according the context changes but also considering the quality preferences associated with the check-in process.

2.1 Business Process Modeling

A Business Process Model consists of a set of logically ordered activities that are performed to produce goods or services [3]. BPMN is a workflow based language that models business process based on flows of task and data. Figure 2 depicts an example of a process model. The tasks (rounded box) represent activities, while the decision points (diamond shapes) represent choices that can be selected, and the events (circles) represent triggers that start or end a process. In Figure 2 we have a sequence of activities that are performed during the check-in and boarding process. The process starts with request of the flight ticket in order to verify the flight information, and then the check-in is performed. After that other airport control checks are executed. The last step is to board the airplane.

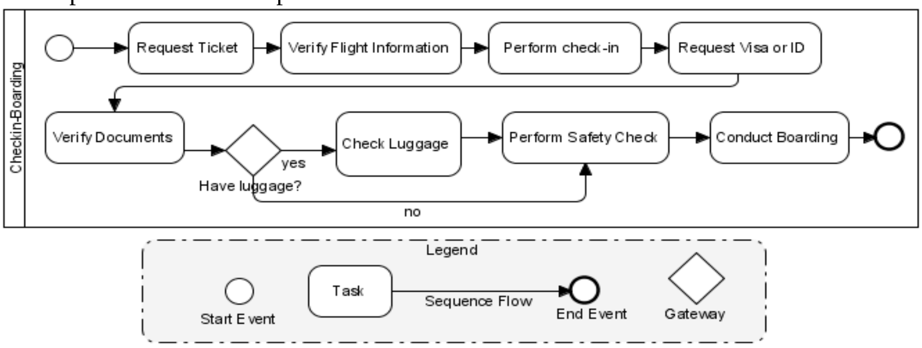

Fig. 2. BPMN model of an Airplane Check-in and Boarding process

Variability modeling in business processes models represents alternative ways of how activities are performed including the actor responsible for performing them, the resources required, and so on. We rely on Variants and Variation Points to describe the desired variability. Variation Points are the subjects of change, while variants are the objects of change [9]. In our case, both Variants and Variation Points are represented by business process model fragments. Observe that Variants can be included or removed from Variation Points. It is important highlight that the variability information is stored in a specific model without extension of the initial business process model. See in Figure 1 that the BPMN is part of another metamodel.

The description of a *Variation Point* includes an identifier (name), an operator (AND, OR, XOR), a point of reference (begin and end) and a list of the *Variants* that can be placed in it. *Variants* can be associated to one or more *Variation Points*. The *Variation Point* in its turn begins and ends in points of the process that can be of any type. Moreover, the *Variants* can interact with each other – e.g., they can require or exclude the presence of other *Variants* on another *Variation Points*.

In our approach, the *Variants* will be related to a *Variation Point* through a pattern. In order to describe the variants we use an identifier, the point where it should be inserted, the dependencies that may be present and a pattern. *Patterns* are used to indicate how process elements will be placed in the resulting business process model. Note that we refer to workflow patterns described in literature [10]. It is important remember that these patterns are specific for workflow languages and differ from the design patterns used in software development. Several types of workflow patterns are available such as sequence, parallel split, exclusive choice, multiple choices and so on.

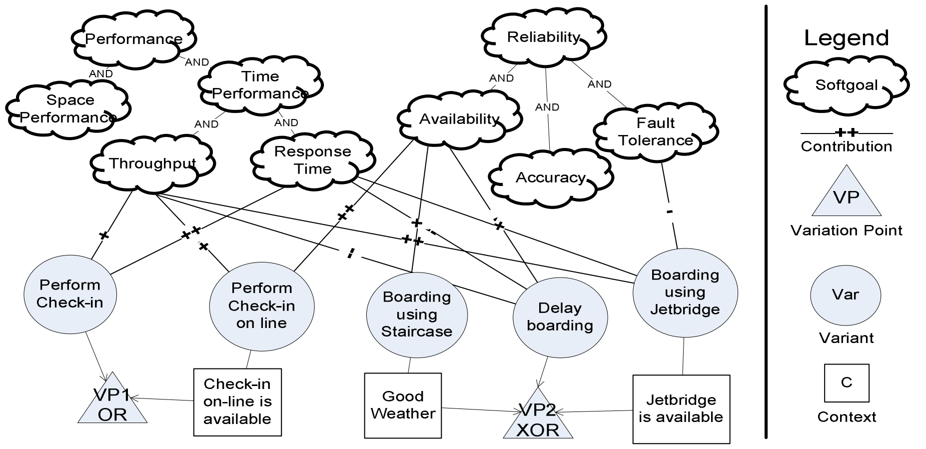

Fig. 3. Relating Variability, context and non-functional requirements.

2.2 Non-Functional Requirements - NFR

NFRs are requirements that describe qualities and constraints. Requirements Engineers have long relied on the concept of NFR to describe and analyze the requirements of systems and their relationship with the functional ones. The NFR Framework [8] introduces the concept of Softgoal to represent the NFR as well as means to assess their satisfaction. NFRs can be reused through catalogues that describe how to decompose and operationalize the NFR.

In our work we rely on the NFR Framework [8] to represent the quality attributes required by our approach. Since we are using *NFR* to configure the business process model, it also needs to be linked to the *Variants*. The relationship between the *NFR* and *Variants* is expressed by contributions, which indicate the positive and negative interaction among them. Figure 3 presents a simplified example of catalogues for *Performance* and *Reliability* linked to elements of our approach. A *Variant* can contribute to several *NFR (see Perform Check-in On Line)*, whilst a *NFR* can be contributed by several *Variants* (check *Availability*). However, a *Variant* has just one contribution value to a *NFR* at a time. In our case we adopt a numerical scale, from positive, with maximum value of 1, to negative with minimum value of -1. For example, *Delay Boarding* variant has a very negative (-1) impact on the *Availability* softgoal (constraint).

NFRs are important for business processes modeling. However, they are seldom considered during modeling. Some few approaches apply NFRs during the design by means of extensions of business process modeling languages. For instance, [11] and Pavlovski et al. [12] take the Non-functional requirements into consideration during the software design process. The former by using NFR catalogs during the design and the latter by extending the BPM to incorporate NFR. However, neither considers the variability in their solutions.

2.3 Contextualization

A Context is a partial state of world that is relevant to achieve goals [14]. In our case it is relevant information that could affect the business process execution. Contextual analysis is based on context annotations. Annotations are attached to elements of a model in order to indicate what is the relevant context information that can affect that part of the model. Hence, contextual parts of that model can be enabled or disabled. During the analysis the *contexts* can be associated to *facts* and *statements*. Facts can be directly assessed, while *statements* must be decomposed as facts. *Contexts* are linked to sets of *facts* that can be assessed to identify the validity of the *context*. In this paper, we adopted a simplified version of the proposal of Ali et al [14]. A *Context* is described in natural language, and it is composed by *Context Expressions* that allow the computation of validity of a context in a given moment. A *Context Expression* associates the values of the monitorable variables to logical expressions to assess if the context is valid or not.

3 NFR-Driven Configuration of Business Process

Our novel approach consists of five activities: *Elicit Variability, Describe Variability, Analyze Context*, *Link NFRs & Variants* and *Perform Configuration*. The first four steps are performed at design time (see Fig 4). While the last step, *Perform Configuration is executed* at runtime (see Fig. 6). Note that the configuration is driven by a clear criteria. Hence the rationale for the selected configuration becomes explicit. Moreover, it can be performed while the processes are running, i.e. it becomes run time adaptable!

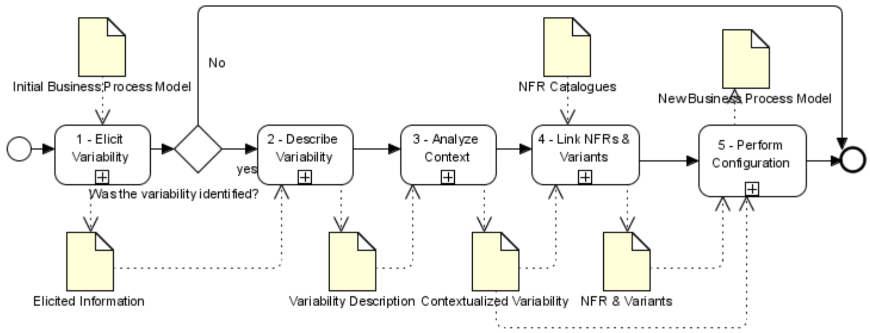

Fig.4. The process of our approach

Variability Elicitation. The process starts by identifying and discovering possible variations in a model. The objective is to uncover different ways to carry out a process, the effects of the inclusion, change or exclusion of elements of the model. A questionnaire can be used to help to identify different perspectives in the business process models that were not clear in the initial model. In order to perform this elicitation we use an information analysis framework [16] that explores different facets of the information and obtain new data about it. The elicitation produces a list of variations that needs to be represented in order to reflect the nature of business process.

Variability Description. Once the possible variants and variation points of the process are identified, they need to be described. Hence, in this step we specify the variations in terms of the standard BPMN notation. As previously explained, we represent the variations using the concepts of Variation Points and Variants. Variation Point is the place where the variation occurs. The variants are described as parts of BPMN and associated with a pattern that indicates how it will be placed in the model.

Figure 5 presents some variants for the check-in. The Check-in can be performed in several ways, it can be performed manually (variant A) in the case where the check-in system is down. It also can be executed at the airport by an operator using a support system (variant B), or by the passenger using an online check-in (variant C). There is also the possibility in combine two ways (variant D), which increases the reliability of the process.

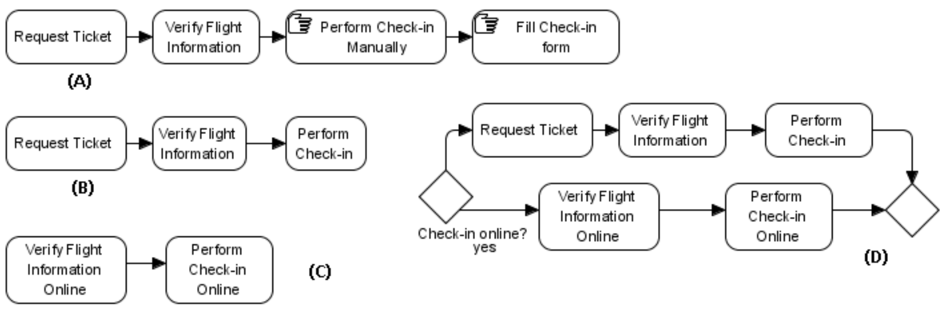

Fig.5. The process of our approach

Context Analysis. The third activity in our approach is to identify the Contexts that could affect the model. It is performed by studying the domain and the relationship of the actors and domain concepts [14]. The user or system states in the environment are described as Contexts. The Context can capture what is going on as well as the location related information (where). It can also include the available resources and other relevant information. A Context consists of expressions and variables that need to be evaluated to check if the context holds.

We need to identify the relationship between the Context and its Variants and Variation Points. Note that the task can only be performed if the Context is valid. Hence, the Contexts represent the monitorable runtime information that will enable or disable possible alternative ways to deal with the process. The evaluation of NFR is performed at design time and the result is used at the runtime. Hence, if the best solution for a NFR is disabled due to the context then another one can be considered.

Linking NFRs and Variants. In this fourth step we identify the NFRs that are critical for the process. Moreover, we define the impact of each Variant to the NFR by means of contribution links. This information can be gathered interviewing experts involved in the business process, using requirements catalogues or any mix of elicitation techniques. Note that the NFR analysis can indicate that several (possibly conflicting) non-functional requirements are to be met.

Once the NFR are identified, we perform the linkage between the process variants and the requirements. These links will be represented using matrices (not shown due to space limitation), which is a usual and scalable solution for representing this kind of information. Moreover, matrices allow the construction of views containing only a partial representation of the variants and the requirements, simplifying its analysis.

NFRs can be used to prioritize the Variants, which lead to the selection of the configuration. Since many alternatives can emerge during the elicitation process, the contribution analysis can be time consuming. However, we claim that the use of NFRs as selection criteria can help to reduce the variability space and thus drive the modification process.

Configuration of Business Process. The configuration of process is a critical step in our approach. All the collected and modeled information is used to obtain new process models. In this last activity we consider the Variation Points and the Variants of the business process, and assess how they impact the non-functional requirements. This information can be used to support the configuration itself (see Fig. 6). It can be performed based on Variants selection or the most critical NFRs.

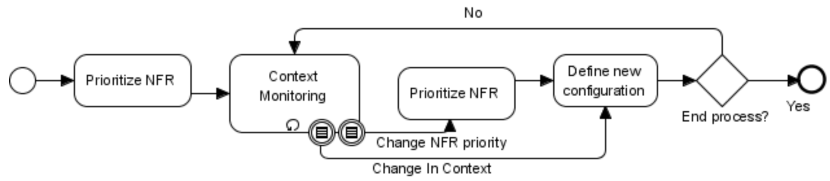

Fig.6 . The Perform Configuration sub-process

Some solutions only rely on expert judgment and NFRs to resolve conflict at design time [8]. Since we are dealing with runtime adaptability, it may not be possible to rely on experts (e.g., they could not be available anymore). In our approach we require the *NFR* prioritization to be conducted before entering in the monitoring loop (see Fig. 6). It is performed by an analyst, who assigns weights to each *NFR* according to their priority weights. Moreover, the *Variation Points* must be associated to a *Context* independent *Variant*.

There are several ways to sort out priorities variants using *NFRs.* A common solution is to rely on weighted averages, where contributions can be counted and weighed according the *NFR*. Although, this method is intuitive it could hide the interaction between the *NFRs*. In order to obtain a global ranking that takes into account the local interactions we adopted a multi-criteria decision making method. We chose the Analytic Hierarchical Process (AHP) [17] method which generates a global preference measure based on the choice among alternatives. The AHP was selected because it fits well with the structures used in our approach. For example, the hierarchical criteria are represented by the NFR decomposition, while the preferences among alternatives represented by the contributions of Variants to NFR. Moreover, the use of priorities over NFR is also taken in account by the AHP.

According to process described in Figure 6, the next step is to start the Context Monitoring loop that will detect changes in the Context and NFR priorities. Note that if changes in the contexts are detected, a selection of a new configuration is required. Each Variation Point is evaluated to identify the Variant that better fits the non-functional requirement, i.e., the Variants with the highest positive impact on that given NFR. This evaluation is automatically performed.

Let us consider N as the set of *NFRs*, Var as the set of *Variants,* VP the set of *Variation Points*, C as the set of *Contexts*, and $\mathrm{contrib}(\mathrm{v},\mathrm{n})$: the value of the contribution link from the variant v to the *NFR* n. The contribution function varies in the following range: $1 \geq \mathrm{contrib} \geq -1$. The NFR have weights (w) associated to them to express their different priorities.

```
1  for all c ∈ C do
2     validate(c)
3  end for
4  for all vp ∈ VP do
5   for all v_i ∈ Var, where v is part of vp do
6    if isValid(v_i) then
7      validVar[i] = v_i
8      for all n_j ∈ N do
9          contribMatrix[i][j] = contrib (v_i, n_j)
10     end for
11   end if
12    end for
13  end for
14 ranking[] = AHP(contribMatrix[][], validVar[], w)
15  solve(VP, ranking[ ])
```

Fig. 7. Configuration algorithm

First, the algorithm (see Fig. 7) computes the valid contexts (lines 1-3). After that, the Variation Points are evaluated to identify the valid Variants (lines 4-6). A Variant can be valid for one VP and invalid for another. There are two ways to be valid: being associated to a valid context, or having no context associated to it (default situation). The valid *Variants* are included in a specific set (line 7), and their contributions to the *NFR* are computed (lines 8-10). The set of valid *Variants* and their contributions to

NFR are the input for the AHP method that will compute a global ranking of preference among *Variants*. In its turn, this ranking is the input to a solver that will derive a valid solution considering the relationships (i.e., exclude and include) among the *Variants*. Another trigger for the changes is priority modification. Note that if the priority of a *NFR* changes it is dealt with similarly to the contextual change.

Once the set of *Variants* is selected they are grouped in a new instance of the business process model. Each *Variant* has its pattern evaluated and the appropriate action is selected. The set of flow elements that is composed in the variants can be placed in parallel to any other variant in the same variation point. The action substitution means that the original task will be replaced by this one. We have implemented these changes in the model using the Query/Validate/Transform-Operational (QVTO) model transformation language [18]. Due space limitation we do not present the model transformations but it and other support material are available at [21].

We have also used the Eclipse platform modeling tools to develop an editor that creates configuration models based on the meta-model of Figure 1. The initial BPMN is inputted and transformations associated with the patterns are applied.

4 Approach Evaluation

In this section we present an experiment that was performed in order to evaluate some characteristics of our approach. An important question to answer is, do the models produced by our approach are more adapt to the environment than a standard process?

The experiment was executed using the Bonita Open Solution 5.6 business process simulation engine. This environment allows the representation of the process data, simulating the environment resources available to execute the activities, and the configuration of time. Hence, we were able to control the variables of the experiment, allowing for a reproducible study.

The objects of study are *business process configurations* generated using our automatic configuration approach. These models were compared with respect to a *basic* process, using the *same* scenarios. The purpose is to evaluate the business process configurations generated with our approach, verifying if they actually *improve* the process in different context settings. The quality focus is the *performance* of a given process configuration, running in a simulated environment. Two dimensions of performance were considered: the *time* required to execute a process; and the *resources* required to execute a process. The *perspective* is the researcher's point of view. The models used in this experiment were produced by one of the *authors* of this paper, and are an extended version of the process depicted in Fig. 2.

Hypotheses, Variables and Measures. In this experimental study we focused on the following research questions, defining the respective sets of null and alternative hypotheses:

- RQ1: Considering the use of NFRs, do the models produced by our approach consumes less *resources* and require less *time* to execute in comparison with a standard business process configuration?

- RQ2: Considering the adaptation to a context change, do the models produced by our approach consumes less *resources* and require less *time* to execute in comparison with a standard business process configuration?

From these research questions, we generated the following hypothesis:

- $H_0$1: The business process models produced by our approach, when considering the use of NFRs to configure the models, do not consume less resources or less time when compared the standard process;
- $H_0$2: The business process models produced by our approach, when considering the adaptation to a context change, do not consume less resources or less time when compared the standard process.

If the null hypothesis can be rejected with relatively high confidence, it is then possible to formulate alternative hypotheses:

- H_a1: The business process models produced by our approach, when considering the use of NFRs to configure the models, consume less resources and less time when compared the standard process
- H_a2: The business process models produced by our approach, when considering the adaptation to a context change, consume less resources and less time when compared the standard process.

The independent variable of this study is the modeling method which can assume one of the values in {*Standard, Adaptable*} – standard is a traditional BPMN model, without variability; adaptable is a process model generated with our approach, including variability, contextual and non-functional information. The dependent variable is the performance measured using the execution *time* (in hours) and *resources* consumption (in cost). Resources consumption includes personnel costs, equipment costs, fares, and so on. Both dependent variables were calculated by the simulation engine, considering estimated time and resources required to perform each activity of the process. Also, both dependent variables are inversely proportional to the *performance*, i.e. the lowest the value of execution *time* and of *resources* consumption, the better the *performance*. The measure variables were the average *execution time* by instances in hours and the average *resources cost* by instance in dollars.

Design and Execution. A basic requirement of an experiment is the ability to control the object of study and its parameters. Since our approach allows modifying several parameters first we need to control what parameter will be modified. We designed the experiment to assess the impact of contextual changes and the impact of NFRs priority change. Two scenarios were designed: the case where there are no changes in context (S0) in the environment and the situation where there are changes (S1). In the S0 scenario we block the value of contextual information and change the value of NFRs' priorities. This way we can see the impact of changing the NFRs' priorities over the process. In the S1 scenario, we change the value of a context variable and repeat the simulations to verify the impact of contextual change over the resulting process models.

A standard check-in and boarding process described in the literature [2] was considered the standard business process model (see Fig. 1). Afterwards, we used our approach to design a new business process models based on the change in the S1 scenario. For each, scenario we simulated the case with and without *contextual* change. For instance, the variants *perform on-line check-in* and *use jetbridges to* *board* can improve the *performance* of the process by reducing the waiting *time*. However, this type of alternative can increase the *costs* or may be unavailable at the moment when the process needs to be executed. In the scenario with contextual change we consider the *on-line check-in* is unavailable; this change will force the selection of the variant *Perform Check-in* task at the airline counter. We decided to use only a single contextual change in order to isolate the change effect, and avoid uncontrollable results. However, if the contexts do not have cross interaction other context changes could be defined as well.

In order to describe the impact of the context in the process we based our estimate in the data related to the air traffic delay recently experienced in European airports [19]. The simulation data requires the estimative of the tasks duration, the resources costs, and the number of involved personal. For example, the collected information could help to estimate how much time is necessary to perform the task *Conduct Boarding*. As consequence of being based in real information the simulation scenarios are more close to the real situation.

Results and Analysis. The simulation was performed for the 6 processes divided in two groups: 3 processes for scenario S0 and 3 processes for scenario S1. Each process was instantiated 100 times to obtain the average values presented in Table 1 and 2. Table 1 shows the results of the simulation execution for the S0 scenario and the Table 2 the result of S1 scenario. The simulated interval was one month, with working time of 24 hours per day. We consider that this process could be performed at anytime of the day since many airports operate 24 hours a day.

Table 1. Simulation results for scenario S0.

Modeling method NFR considered	Standard None	Adaptable Performance	Adaptable Reliability
Execution Time by instance	0.916	0.616	1.049
Resources Cost by instance	9.27	7.425	7.175

Table 2. Simulation results for scenario S1.

Modeling method NFR considered	Standard None	Adaptable Performance	Adaptable Reliability
Execution Time by instance	0.916	0.666	0.916
Resources Cost by instance	9.321	9.642	9.212

Considering the *Execution time by instance* variable we can identify a very small difference between the average values for the scenario with and without contextual change. This is expected, since in the scenario without contextual change all process variants are available.

Comparing the configurations generated using our (adaptable) approach with the standard configuration in Table 1, we can see that the configuration driven by NFR *reduces* the *execution time*. In the case on which *Performance* is the NFR with higher priority, several variants that contributes positively to improve the process performance reduces the overall execution time – for instance, it is faster to *board with a jetbridge* than using *staircase*. The same goes for the configuration generated prioritizing *reliability*, since in this process configuration there are activities that contribute to reduce interruptions and to improve availability in the process.

If we consider the *resources cost* results, we note that the configurations without contextual change have a smaller cost. This happens because the contextual change suppresses the variant that points to the *Perform On-line Check-in* activity. Moreover, the *on-line check-in* has a smaller cost than a *check-in performed* at the airline counter. In general, the configured processes have different behaviors for the *cost*, for instance, the *performance* prioritized process uses variant that prioritize the *performance* without regarding the *cost* of the alternatives.

Discussions. In this section we relied on simulation as an empirical tool to present a preliminary assessment of our approach. We detected there is a relationship between how the process model behaves according to the changes (i.e., NFRs and Contextual information), and its performance. Considering the resource consumption and execution time we can reject the null hypotheses $H_0 1$ and $H_0 2$, and accept the alternative hypotheses $H_a 1$ and $H_a 2$.

Moreover, we conclude that our adaptable approach has slightly improved the execution time for the check in procedure. Since, this is a process to be executed by all passengers to be flown, the small gains can add up to enormous benefits.

There are threats to the validity of our study. For example, the *reliability* of measures – our measures were compiled from the simulation reports generated by the simulator. In order to deal with this threat we selected just average measures and discarded outliers. There are also *design* threats, such as the interaction among treatments and the mono-method bias. In order to reduce these threats we combine the treatment with different settings including the use of different NFRs such as Performance, Reliability and as parameters for the process configuration, and repeating the simulation with and without contextual changes

5 Related Work

It is well known that some approaches also rely on Software Product Line principles to deal with variability in BPM. For instance, Schnieders and Puhlmann [7] propose the extension of business process modeling languages to describe variability. However, they do not provide mechanisms to drive the configuration of process models. Neither consider NFRs and contextual information as we do.

There are also some works that provide mechanisms to help to drive the configuration of Business Processes. For example, La Rosa et al. [3] propose a questionnaire based approach that relies on configurable process models to obtain new instances of BPM. The users receive guidance during the configuration by answering questionnaires. Their approach is intuitive but it has a limited application for run-time self-configuration since it requires the user intervention to produce new versions of the model. Note that our proposal also supports user interaction to guide the configuration process, e.g. the change in NFR prioritization. However, human interaction is not required at run-time stage to generate new process models.

Lapouchnian et al. [20] offers a goal oriented approach to configure BPM. They obtain business process models from annotated goal models. Moreover, they configure the process model using NFRs represented by softgoals. Our approach also relies on NFRs. However, we adopt a completely different structure to represent the BPM and variability. We start from a reference process model and represent the variants as process chunks instead of using a goal model. In doing so we maintain the representation in the same abstraction level without the need to annotate or convert the models. Besides, we also use contextual information to support dynamic configurations.

De La Vara et al. [15] proposes an approach for contextualization of BPM. They rely on the context analysis of Ali et al. [14] to represent the contextual information. Additionally, they guide the definition of contexts and inclusion of contextual information in the BPM and allow characterization of variants based on the context information. Our approach also relies on contexts. However we deployed a different strategy. We defined the variants and variation point before the inclusion of context information. Moreover, we used NFRs to guide the selection of a configuration and we did not extend the business process modeling language to include the contextual information. In doing so our approach can import BPMN models designed by any tool based in the Eclipse framework BPMN 2.0 metamodel.

6 Conclusions and Future Work

In this paper we proposed a novel and flexible approach for the configuration of business process models. It relies on contextual information and NFRs. A process was outlined. It includes the elicitation of variability information, which is central to the configuration process itself. Besides guiding the configuration with clear criteria, this approach also provides the rationale for the selected configuration.

We have proposed an approach that keeps the variability representation and context-information separate from the business process models. In doing so, we traded off intuitiveness for the sake of flexibility. For instance, we did not need to extend the BPNM to deal with variability. Moreover, we relied on patterns (i.e., work-flow patterns) and analysis algorithms (i.e., SAT solvers). Hence, we envisage that our approach could he used with different business process modeling languages, i.e. few modifications are expected.

Different from some other approaches that support the NFR evaluation (e.g.,[8]), we do not require the user intervention to solve conflicts during the configuration phase. If necessary our approach could also support the interaction with user to update their preferences, i.e., to change priorities of NFR. However, it is not mandatory to produce a new version of the model.

We consider that the most critical part of our approach is to relate the degree of impact of each variant to the NFRs. This could be eased through the creation of catalogs which could help to define, for each kind of activity in a business process, the impact of that activity on specific NFRs.

Some may claim that the approach might be time consuming, as each element in the business process may experience several variations. Certainly, the elicitation effort is also related to the number of non-functional requirements under consideration. However, this seems to be an inherent problem of any approach that deals with variability, since the amount of variations that may arise in real situations is potentially large. We believe that further improvements, currently under way, such as the automation of some of its steps and the adoption of mechanisms to handle complex models, could minimize these shortcomings.

References

1. Chang, J. F.: Business process management systems: strategy and implementation. Auerbach Publications, Baton Rogue, USA (2006).
2. Rosemann, M., Recker, J., Flender, C.: Contextualisation of business processes. International Journal of Business Process Integration and Management. 3, 47 (2008).
3. La Rosa, M., van Der Aalst, W., Dumas, M., ter Hofstede, A.: Questionnaire-based variability modeling for system configuration. SoSyM.. 8, 251–274 (2009).
4. Montero, I., Peña, J., Ruiz-Cortés, A.: Business Family Engineering: Does it make sense? Proc. of the 1st Taller Procesos de Negocio e Ingeniería del Software (PNIS) (2007).
5. BLIND REVIEW
6. BLIND REVIEW
7. Schnieders, A., Puhlmann, F.: Variability Mechanisms in E-Business Process Families. Proceedings of the 9th Int. Conf. on Business Information Systems, BIS 2006 (2006).
8. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers (2000).
9. Pohl, K., Bockle, G., Linden, F.J.V.D.: Software product line engineering. Springer - Verlag Berlin Heidelberg (2005).
10. Wohed, P., van Der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: Pattern-based Analysis of BPMN. , Brisbane, Australia (2005).
11. BLIND REVIEW
12. Pavlovski, C.J., Zou, J.: Non-functional requirements in business process modeling. Proceedings of the 5th APCCM ’08. pp. 103-112. Australian Computer Society, Inc. (2008).
13. Soffer, P., Wand, Y.: On the notion of soft-goals in business process modeling. Business Process Management Journal. 11, 663-679 (2005).
14. Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. Requirements Engineering. 15, 439-458 (2010).
15. De La Vara, J.L., Ali, R., Dalpiaz, F., Sanchez, J., Giorgini, P.: COMPRO : A Methodological Approach for Business Process Contextualisation. Proc. of the 18th International Conference on Cooperative Information Systems (2010).
16. Liaskos, S., Lapouchnian, A., Yu, Y., Yu, E., Mylopoulos, J.: On Goal-based Variability Acquisition and Analysis. Proc.of the RE 2006. pp. 92-96 (2006).
17. Saaty, T.L.: Relative Measurement and Its Generalization in Decision Making Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors The Analytic Hierarchy/Network Process. Rev. R. Acad. Cien. 102, 251-318 (2008).
18. OMG, Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. (2007).
19. Tony Leggat, Catherine Yven, Y..: A Matter of Time : Air Traffic Delay in Europe. EUROCONTROL (2009).
20. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-Driven Design and Configuration Management of Business Processes. Proceedings of the 5th International Conference on Business Process Management, BPM 2007. pp. 246-261 (2007).
21. BLIND REVIEW