Using NFR and Context to Deal with Adaptability in Business Process Models

Emanuel Santos, João Pimentel, Diego Dermeval, Jaelson Castro
Centro de Informática,
Universidade Federal de Pernambuco
Cidade Universitária. S/N, 50741-000 Recife, Brazil
Email: {ebs,jhcp,ddmcm,jbc}@cin.ufpe.br

Oscar Pastor
Centro de Investigación en Métodos de Producción de Software,
Universidad Politécnica de Valencia,
Camino de Vera s/n 46022 Valencia, Spain
Email: oscar@pros.upv.es

***Abstract*—Business processes can be performed in different ways according to the characteristics of the organizational environment where they are placed. Representing and configuring business processes variability for a specific organization allows the appropriate execution of processes. However, selecting a configuration for business process is a challenging activity. Even though current approaches allow the representation of variability of business process models, the selection of business variants in a given context is a difficult issue. In this paper we advocate the use of Non-Functional Requirements (NFR) associated to contexts in order to configure business process. Thus, we present a configuration approach based on NFR and context models.**

Keywords- Business Process models; Variability; Context; Non-functional Requirements.

I. INTRODUCTION

Business processes have become an essential element in the development of information systems (IS). An information system designed to automate the activities of a business process can better suit the needs of an organization [1]. Nonetheless, the adoption of business processes in connection with software development involves some challenges. Since business processes may vary from one organization to another, and even within the same organization, the creation of business process models requires special attention. Hence, the representation of their variability [2] [3] is of utter importance. Moreover, in more dynamic environments, when using business process models as basis for the development of IS, the ability to adapt the business processes to changes in the organizational environment should also be taken into account.

Process-Aware Information Systems (PAIS) are software systems that manage and execute operational processes involving people, applications, and/or information sources on the basis of process models [1]. Current business process management suites such as jBPM (http://www.jboss/jbpm) and Activiti (http:www.activiti.org) have integrated the process definition and execution with the development of information systems that support the process.

Much effort has been made towards dealing with variability and enabling the adaptation of business processes to changes in the organizational environment. As result, several solutions have been proposed by industry and academy, like Schnieders and Puhlmann [3], Montero et al. [2], Hallerbach et al. [4], and La Rosa et al. [5]. These proposals describe how to represent variability and how to obtain customized processes by producing new instances or by configuring process models [6]. However, the problem of choosing the most suitable alternative—the so-called process configuration—is not solved yet. In the industry, the configuration still is performed in an ad hoc basis, guided solely by the analyst's experience.

Context-awareness is essential to provide run-time adaptation based on business process models. In fact, several approaches have related context-awareness to changes in business process models [7] [8] [9] [10] [11]. A context can be defined as a state of the world that is relevant to an actor goal [12] or as a stimulus for change [10]. It allows the identification of variables that impact the structure of business processes.

Non-Functional Requirements (NFR), related to quality requirements, define constraints that the process must comply to. In this paper we advocate the use of non-functional requirements as a suitable criterion for guiding the process configuration, since they represent the high-level characteristics from which processes are usually evaluated—e.g. cost, performance, accuracy, and the like. Besides, NFR has been extensively studied [13] and the current body of knowledge provides plenty of techniques that can be borrowed and used in this new business process domain. In fact, some recent works are already exploring the possibilities of integrating NFR and business process models [14] [15] [16] [17].

We acknowledge the importance of context-awareness in business process design in order to enable run-time adaptation. Thus, in this paper we extend our previous NFR-driven configuration approach [18] [19] to consider the contextualization of BPMN models.

Our new approach consists of a *process* to identify variation points and variants in a business process model as well as a *selection mechanism* to create new instances of a business process (configurations) with the variants that better suit the system's context and its NFR. An essential step in this approach is the identification and linking of business process variants with NFR and contextual variables, which will guide the configuration of the business process.

In Section II we are going to introduce the background and related works. Section III introduces a running example. Section IV present our new approach. We discuss the results in Section V. Finally, in Section VI we conclude the paper.

II. BACKGROUND AND RELATED WORKS

Our proposal concerns process configuration, using the notions of NFR and context. In the following subsections we present related works on these topics.

A. Process Configuration

Since each organization may have different ways to perform a process, achieving flexibility in business processes has become an objective of several approaches. Many of them dealt with the concept of variability and how to obtain customized business process models for a specific organization.

One possibility to deal with variability is using the Software Product Lines (SPL) perspective. The proposal of Schnieders and Puhlmann [3] describes extensions for Business Process modeling languages allowing the construction of process families.

Another possibility to apply the SPL concepts is representing the variability in independent models—usually, feature models. Montero et al. [2] have proposed an approach that uses a combination of feature models and business process models to deal with variability. They provide a process to build business process instances using the SPL perspective.

The flexibility of business process models can also be handled using configurable models [6]. La Rosa et al. [20] use a questionnaire-based approach to deal with variability and to drive the configuration of business process models. In their approach, answers for questionnaires offer ways to select among configurations of a process, described in configurable models, i.e., C-EPC or C-YAWL. After the configuration, the business process model can be individualized to obtain a single model without configuration marks.

In this paper we are concerned with the variability of business processes models as well. We propose to use independent models to represent variability instead of extensions of languages [3] or configurable models [20]. An independent model [19] provides flexibility to our approach in order to include other concepts as well (e.g., contextual information and NFR).

B. NFR and Business Process Models

Non-Functional Requirements (NFR) are requirements that describe qualities and constraints of software. The Requirements Engineering field has been using the concept of NFR to describe and analyze the requirements of systems and their relation with the functional ones. The NFR Framework is one of the most known approaches to deal with NFR [13]. It introduces the concept of Softgoal to represent the NFR and mechanisms of analysis to assess their satisfaction.

The importance of Non-Functional Requirements for business process models have been acknowledged in the last years. The Non-functional requirements have been applied in the design of business process models through extensions of business process modeling languages, allowing for richer analysis of the process model. For instance, Xavier et al. [16] and Pavlovski et al. [15] include the Non-functional requirements into the software design process. The former by using NFR catalogs during the design and the latter by extending the business process models to incorporate NFR. Neither considers the variability in their solutions. Soffer and Wand [17] also advocate the importance of NFR (expressed by Softgoals) to the business process design. They provide a conceptual framework to clarify the notion of Softgoal and use it to represent quality in business process models.

Lapouchnian et al. [21] describe how to use goal models and annotation mechanisms to derive business process models. They use NFR, represented by Softgoals, to drive the configuration of processes. They start their approach with the goal model and then derive the business process model. In contrast, we start from a business process model and derive another business process model instance. Moreover, we are not using annotations on models; all information necessary during the configuration is separated from the models.

C. Contextualization of Business Process Models

The contextualization of models consists of the inclusion of useful information about the contexts that can affect the model. The need for systems that dynamically adapt themselves is increasing dramatically. Systems and the design methods must be flexible enough to support changes. Indeed, contextualization is used as a way to allow adaptation to changes both at design time and at run-time. In order to perform context-based adaptation of business process models two news mechanisms are necessary: monitoring and actuation. The former identifies what should be changed, whilst the latter performs the changes themselves. These mechanisms are the basis of self-adaptive and autonomic systems. The monitoring mechanism requires the definition of attributes that will be monitored. This set of attributes is part of the context which can be modeled using contextual analysis [12]. However, the identification of context for business process models is not our focus in this paper. Interested readers can find some methodologies addressing this issue in the literature, e.g. [11] [9].

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 context analysis the contexts are associated with facts, which can be assessed directly, and statements, which must be decomposed in facts to be assessed. The contexts will be associated with sets of facts that can be assessed to identify valid contexts [12].

Rosemann et al. [10] describe a model to represent the context information in business process models. They describe a context in several layers, such as immediate, internal, external and environmental layers. They provide a valuable conceptual description about contexts in business processes, and discuss how to identify the contexts. Unfortunately, they did not provide reasoning mechanisms (e.g., algorithms) to be used in connection with contexts for runtime adaptation.

De La Vara et al. [11] describe an approach to include contextualization in business process models. In many ways their work is the most similar to our approach. Since it uses contextualization analysis, as proposed by Ali et al. [12], their approach incorporates the concepts of context, fact and statements to represent contextual information.

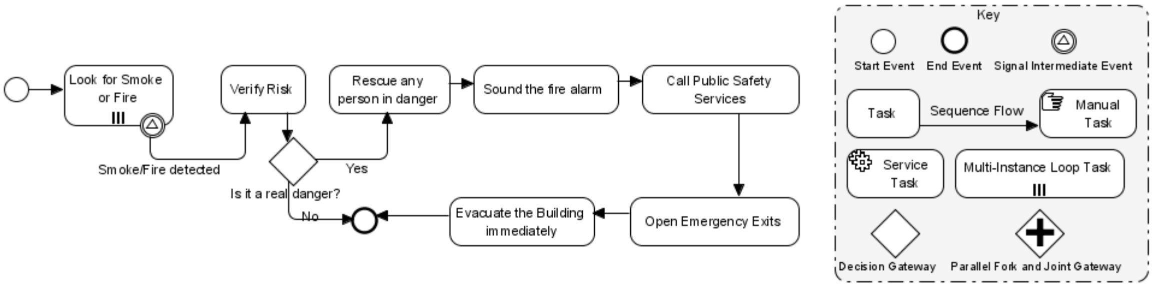

Figure 1. Initial BPMN model for the response to a fire disaster.

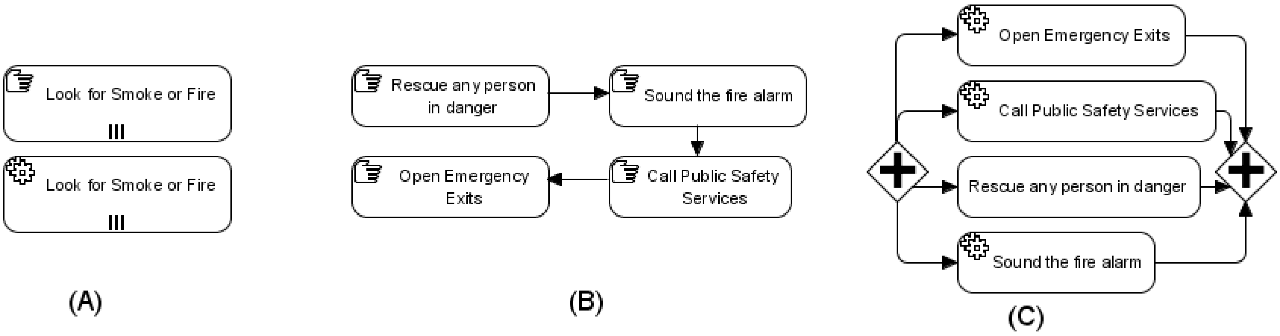

Figure 2. Possible variants for this process

De La Vara et al. [11] describe a process to introduce contextualization into business process models through context analysis. The context analysis allows the derivation of the conceptual model that can be monitored at run-time. However, in our approach in order to provide more flexibility we separate all configuration information from the model itself. Moreover, the identification variants and the linking with NFR is performed in different steps of the process analysis. We deal with the variants and variation points since the beginning of our approach.

Bessai et al. [9] proposed a context-aware business process approach that evaluates and enhances business process models for a context using work-flow patterns. Our approach is similar in a way to their work. However, we have focused on the use of NFR as a mechanism to configure the model. In order to do so, our work requires qualitative analysis to identify the contributions to NFR.

Greenwood and colleagues [22] developed a suite to monitor and adapt business process models and integrated software applications. Their approach is a goal-oriented business process management system that supports autonomic features. They have proposed their own language and rely on planning algorithms to support the self-configuration features. In contrast, our proposal has been developed over a standard model (BPMN), which allows its integration with suites that support this standard.

III. Running Example

Socio-Technical Systems (STS) are characterized by the interplay between social and technical components. In STS, the human actors are components of the system along with the software and hardware. Due to the presence of human actors, the STS present specific properties such as dynamic organizational objectives and non-determinism.

The scenario of crisis management introduced by [32] illustrates these properties for an Ambient Intelligence (AmI) socio-technical system. When a disaster (e.g., fires, floods, etc.) occurs several responses can be triggered by such a system. The response to a disaster may vary in many points according to the disaster type and severity. In this scenario, the processes to deal with the crises also vary since they need to be suitable to a specific situation. Thus, context-awareness and adaptability of processes are essential to this type of scenario. We will use this scenario as a running example to explain the steps of our approach.

The AmI consists of a monitoring system that will alert if a possible crisis situation arises. It may also identify possible measures to mitigate or respond to the crisis. Both human and automatic components are involved in the process of identification and response. For example, in the case of fires, the system's smoke detection mechanisms will identify if it is a real menace of fire and activate an alert system (e.g., alarms or call firefighters) accordingly. The humans may act according to a well defined process specially defined to respond to this type of crisis. The AmI will have to change the process according to the context. The process in Fig. 1 describes a base process to deal with a fire event in a building. Several variants can be incorporated to this process, including the way how the activities are ordered and who will perform them. Fig. 2 presents examples of variants. For instance, a certain task can be performed manually or by software service (A), they can be placed in a sequence (B) or as parallel tasks (C).

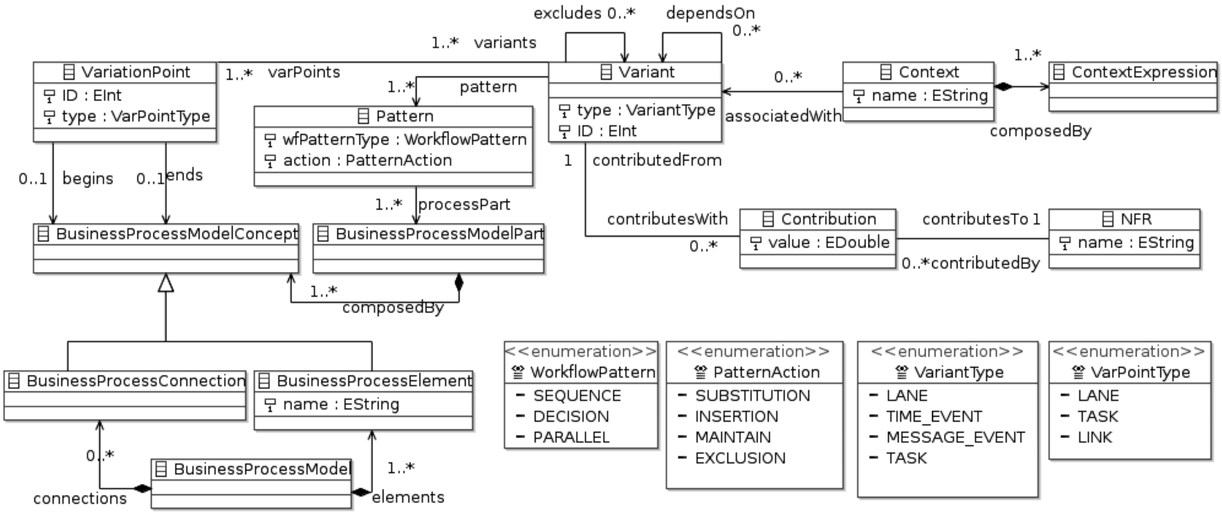

Figure 3. Conceptual model used in our approach

IV. Configuring Business Process Models with NFR

Before describing the configuration process itself, we describe the basic concepts used in our approach. Fig 3 presents these concepts and their relationships as a conceptual model.

A. Basic Concepts

Variability modeling in business processes models represents alternative ways of how to perform its activities, who will be responsible for performing them, which resources are 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 [24]. In our case, both variants and variation points are represented by business process models fragments. Observe that variants can be included or removed from variation points.

The description of a *Variation Point* includes an identifier (name), a type (task, link, sequence), a point of reference (begin and end) and a list of the Variants that can be placed in it.

Variants can be associated with 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 (*BusinessProcessModelConcept*). Moreover, the variants can interact with each other—they can require or exclude the presence of other variants on other variation points.

In our approach, the *Variants* (Fig. 3) will be associated 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 of insertion. Note that patterns of insertion are already described in the literature [25]. For example, insertion in sequence, parallelism, optional behavior, and so on.

Patterns can indicate if the process part that is in the variant will be included or removed as well as how to interact with the base process by substituting or maintaining some part of process. Indeed, some authors argue that the use of patterns may help to preserve the correctness of a process [25].

Non-Functional Requirement - NFR is another concept required by our approach. Since we are using NFR to configure the business process model, it 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 [13]. A variant can contribute to several NFR, and the NFR can be contributed by several variants. However, a variant has just one contribution value for a NFR at a time. Here, we adopted a numerical scale, from positive, with maximum value of 1, to negative with minimum value of -1. In Section IV-B we detail how to use this scale.

Finally, the concept of *Context* is associated to the variants. Context is the current state of the user or system to the objects in their surrounding environment [26]. 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 values of monitorable variables to logical expressions to assess if the context is valid or not.

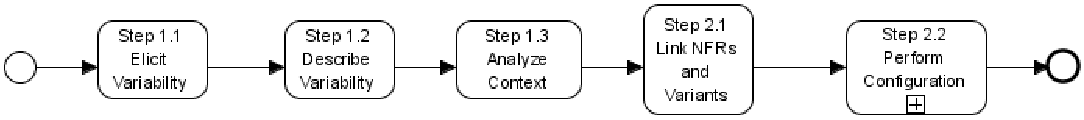

Figure 4. Process Considering Context

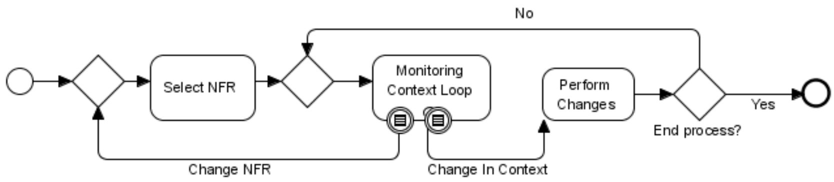

Figure 5. Perform configuration Sub-process

B. Process

The process in Fig. 4 describes the steps of our approach: *Elicit Variability*, *Describe Variability*, *Analyze Context*, *Link NFR & Variants* and *Perform Configuration*. The first four steps are performed at design time, the last one, *Perform Configuration*, at run-time (Fig. 5).

TABLE I. VARIANTS

Task	Variant	VP	ID	Action/ Pattern
Look for Smoke or Fire	Fire detected by a person	VP02	VAR01	Maintain/ Sequence
	Fire detected by sensors	VP02	VAR02	Substitution/ Sequence
Rescue any person in danger	Firefighter perform Rescue	VP01	VAR03	Inclusion/ Parallelism
	Rescue any person in danger immediately	VP01	VAR04	Maintain/ Sequence
Sound fire alarm	Sounds fire alarm manually	VP01	VAR05	Maintain/ Parallelism
	AmI System Sounds fire alarm	VP01	VAR06	Substitution/ Parallelism

1) Elicit Variability.

The elicitation of variability is the activity of identifying and discovering possible variations in a model. The objective is to identify 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 [27] that explores different facets of the information and obtain new data about it. In the context of BPMN models we will use this framework to inquire the tasks, activities and sub-processes of model and identify new information about them. This framework is very simple to be applied as it is based on questions such as Who? How? When? which are common place in requirements engineering elicitation.

2) Describe Variability

Once identified the possible variants and variation points of the process, they are expressed in terms of the BPMN notation. The elicitation results in a list of variations that needs to be represented in order to reflect the nature of business process. 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. The *Rescue any person in danger* variant (Table I) for instance, can be executed in parallel if a Firefighter handles it. However, if performed by the any other person it should be handled sequentially.

3) Analyze Context

In this step the business process model is analyzed to identify the contexts that could affect the model. The contexts can be identified by analyzing the domain and the relationship of the actors and domain concepts [26]. The user or system states in the environment are described as contexts. The context can describe *What is going on?, Where are they located?, What are the resources available for use?,* and so on. A context is composed by expressions/variables that need to be evaluated to check if the context holds (Table II).

Once defined, we can identify the relation of the context with the *variants* and *variation points*. For example, if the context is valid then the task could be performed. The contexts will act as 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. If the best solution for a NFR is disabled due to the context then the second best can be applied.

In the case of change in the context all pairs of *variation points* and variant must be re-evaluated to determine the most preferred solution. For example, if the *Fire Fighters Called Automatically* context becomes invalid, the *Call Public Safety Services* task cannot be performed automatically anymore and an alternative task shall be considered.

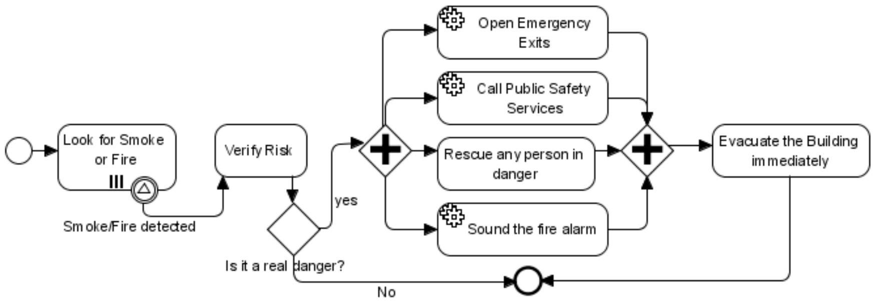

Figure 6. Example of process instance the AmI System performing some tasks in parallel.

TABLE II. CONTEXTS

Context	Context Expression	Variant
FireAlarmIsOn	SmokeSensorIsOn=true and FireConfirmed=true	VAR05 and VAR06
FirefightersCalledAutomatically	FireAlarmIsOn=true and NetworkIsUp = true	VAR07
EvacuateImmediately	FireAlarmIsOn=true and EmergencyExitsOpen=true and (RiskLevel = Medium or RiskLevel = High)	VAR12

Due to the introduction of contextual information the way how the business process models configuration is performed needs to be adapted, see the *Perform Configuration* activity. The configuration, as described in Fig. 5, requires that the contribution and context analysis have already been performed. With this information the user can select the appropriate NFR that will guide the configuration of the process. In our previous work [18][19] the process could be configured by using Bottom-Up or Top-Down analysis. The former requires process variants to be selected before the analysis of the impact over the NFR. Since a context can become invalid in some point, the variants selected using Bottom-Up analysis can be invalidated as well. Thus, due to the possibility to generate invalid instances the Bottom-Up analysis is not the preferred way to associate contexts.

4) Link NFR and Variants

In this step we identify the NFR that are important for the process and define the impact of each variant on the NFR. The NFR is linked to the business process variants earlier identified. We begin identifying which NFR will be taken into consideration. For example, we can interview people involved in the business process [28], use requirements catalogs [14] or rely on a mix of elicitation techniques. The analysis of NFR can deal with several non-functional requirements at the same time. Due to space constraints as well as to simplify the explanation, in this paper we will present just two NFR.

Once the NFR are identified, we perform the linkage between the process variants and the requirements. These links will be represented using matrices, which is a usual and scalable solution for representing this kind of information. Moreover, matrices allow the building of views containing only a partial representation of the variants and the requirements, simplifying its analysis.

Let us consider the qualitative scale of the NFR Framework [13]. In the NFR Framework, the most positive impact on a non-functional requirement is *Make*, a partial positive impact is *Help*, a partial negative impact is *Hurt* and the most negative impact is *Break*. These values can be mapped, respectively, to 1, 0.5, –0.5 and –1, in our scale. In order to make it more user friendly, this scale can be replaced by any other scale, provided that the required transformation is performed.

TABLE III. NFR AND VARIANTS

Variant	NFR	
	Reliability	**Fast Response**
Call Public Safety Services automatically	-1	1
Call Public Safety Services using a land line inside the building	-0,5	0,5
Call Public Safety Services by Cellphone	0,5	0,5

The NFR can be used to prioritize the variants helping with the selection of the (semi) optimal configuration. Since many alternatives can emerge during the elicitation process, the contribution analysis can be time consuming. In [19] we discuss the complexity of using NFR to select a configuration. The introduction of contextual information could further complicate it, as it is now necessary to consider the interaction between contexts and variants. However, the use of NFR as selection criteria can contribute to reduce the variability space and thus help to drive the modification process.

In the scenario of crisis management of a disaster Safety is the primary NFR. We decomposed Safety in two NFR: *Fast Response* that indicates how the variants will contribute to an immediate response; and *Reliability* that points to how reliable the variants are for this situation. Every solution that automatically performs a task or increases the parallelism will contribute positively to *Fast Response* quality attribute, since less time will be spent in these tasks. However, the automated activities rely on an infrastructure that can be affected by the disaster and collapse. Hence it reduces the *Reliability* of the process as a whole.

The best example to illustrate this possibility is *the Call Public Safety Services* (e.g., Firefighters) activity. In Table III we present the contribution for the *Reliability* and *Fast*

Response NFR from three possible variants of the Call Public Safety Service task. If performed automatically by an *AmI System* it will present a fast response. However, it depends on the network infrastructure that can be disabled by a more critical disaster. On the other hand, using a telephone could delay the response but it is more reliable since there are fixed phones as well cell phones replicated in several places which relies on independent infrastructures.

5) Perform Configuration

The configuration can be performed by selecting *Variants* or by prioritizing the *NFR*. At this point we know the *Variation Points* and the *Variants* of the business process, and how they impact the non-functional requirements. Now we will use this information to support the configuration itself.

There are various possible ways for analyzing the impact of each configuration on the NFR: for example, top-down analysis or bottom-up analysis. In the top-down analysis we select which non-functional requirement has the maximum priority, and then derive a process configuration that maximizes the selected NFR. Alternatively, in the bottom-up analysis we define a process configuration, by selecting a subset of variants, and then observe how this configuration affects the non-functional requirements.

These analyses can be performed semi-automatically. The algorithms to perform the evaluation of alternatives using non-functional requirements are already available in the literature [13] [29]. The choice of matrices as data structure allows the usage of even more sophisticated algorithms, in order to resolve dependencies and conflicts that may arise. However, it is up to the analyst to select the NFR used as criteria or the strategy – e.g. top-down fashion - or the configuration that will be evaluated - in the bottom-up analysis. In the sequel, we present fragments of the analysis.

The top-down analysis consists of obtaining an instance of the model based on the selection of non-functional requirements. Consequently, the analyst defines which NFR will be prioritized. Each variation point is evaluated to identify the variant that better fits the selected non-functional requirement, i.e., the variant with the biggest positive impact on that NFR. This evaluation can be performed automatically. However, dependencies between variants have to be taken into consideration as well. If a variant X requires variant Y, the calculation will be performed considering X and Y altogether. There are several algorithms in the literature that could be used.

Fig. 6 presents a solution process for prioritization of the *Fast Response* NFR. Observe that the tasks were placed in parallel to improve the response time. Moreover, the selected variants were the ones that include the automatic execution of the activity, represented by a gear on the top of the task. Note that this option was selected according to the analysis of the NFR contributions. Since we are using contexts the process can dynamically change and be reconfigured at runtime to adapt to the situation.

The bottom-up analysis consists of selecting a subset of variants and using the linkage matrix to calculate the impact of that configuration to the non-functional requirements. Hence, an analyst could, for instance, evaluate if the current configuration is satisfactory.

V. Discussion

We have proposed an approach that keeps the variability representation, configuration knowledge and context-information separated from the business process models. In doing so, our approach trades off intuitiveness for the sake of flexibility. For instance, we do not need to extend the process language to deal with variability; neither link it to Non-Functional Requirements. With the use of language patterns (e.g., work-flow patterns) and analysis algorithms (e.g., SAT solvers) we can apply our approach to other business process modeling languages.

The interference among NFR is a well known problem in requirements engineering. However, several tactics are available to handle it. The NFR framework [13], for instance, uses the contribution and correlation links associated with analysis algorithms to identify possible conflicts. Other possible solution is the use of analysis methods such as Win-Win or AHP to identify and solve conflicts [30].

The use of matrices as data structures provides the flexibility necessary to include information that could be in different models into the same representation. The definition of the matrices may seem to be ad hoc, but the fields and their relationships are defined according to the conceptual model presented in Fig. 3.

VI. Conclusion and Future Work

In this paper we have included support for context models in an approach to guide the configuration of business processes using Non-Functional Requirements. 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.

Our approach is part of an ongoing work. As such, much remains to be done. Some may claim that the approach may be time consuming, as each element in the business process may experience several variations. This elicitation effort is also related to the number of non-functional requirements under considered. 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. Moreover, we believe that further improvements under way - such as the automation of some of its steps and the adoption of mechanisms to handle complex models [31], could minimize these shortcomings. We focused on the definition of monitoring information as well as how to link it to rest of the approach. Later we will deal with the implementation aspects of monitoring activity. Our next step is to implement a prototype using the monitoring mechanism of jBoss-jBPM to include context variables in the runtime engine.

Another limitation is that our approach requires an experienced analyst, acquainted with the domain being modeled. Fluent in the BPMN notation is also a must.

We consider that the hardest part of this approach is defining the degree of impact of each variant on the NFR. This could be softened through the creation of a catalog that suggests, for each kind of activity in a business process, the impact that activity has on a list of NFR.

ACKNOWLEDGMENT

We would like to thank funding agencies that supported this research: CNPq, CAPES, FACEPE and UPV PAID-02-10.

REFERENCES

[1] M. Dumas, W. M. P. van Der Aalst, and A. H. M. T. Hofstede, Process-Aware Information Systems: Bridging People and Software through Process Technology. John Wiley & Sons, 2005.

[2] I. Montero, J. Peña, and A. Ruiz-Cortés, "Representing Runtime Variability in Business-Driven Development Systems," in Proceedings of the Seventh International Conference on Composition-Based Software Systems (ICCBSS08). IEEE Computer Society Press, 2008, pp. 605–608.

[3] A. Schnieders and F. Puhlmann, "Variability Mechanisms in E-Business Process Families," in Proceedings of the 9th International Conference on Business Information Systems, BIS 2006, 2006.

[4] A. Hallerbach, T. Bauer, and M. Reichert, "Capturing variability in business process models: the Provop approach," Software Process: Improvement and Practice, vol. 22, no. 6-7, pp. 519–546, 2009.

[5] M. La Rosa, "Managing Variability in Process-Aware Information Systems," PHD Thesis, Queensland University of Technology, 2009.

[6] F. Gottschalk, W. M. P. van Der Aalst, M. H. Jansen-Vullers, and M. La Rosa, "Configurable Workflow Models," International Journal of Cooperative Information Systems, vol. 17, no. 02, p. 177, 2008.

[7] A. Hallerbach, T. Bauer, and M. Reichert, "Context-based Configuration of Process Variants," in Proc. of the 3rd International Workshop on Context-Aware Business Process Management (TCoB'08), no. Activity 1, Barcelona, Spain, 2008, pp. 31–40.

[8] K. Ploesser, M. Peleg, P. Soffer, M. Rosemann, and J. Recker, "Learning from Context to Improve Business Processes," pp. 1–7, Jan. 2009.

[9] K. Bessai, B. Claudepierre, O. Saidani, and S. Nurcan, "Context-aware business process evaluation and redesign," in Proc. of the 9th Workshop on Business Process Modeling, Development and Support (BPMDS 2008). Montpellier, France: Springer, 2008, pp. 1–10.

[10] M. Rosemann, J. Recker, and C. Flender, "Contextualisation of business processes," International Journal of Business Process Integration and Management, vol. 3, no. 1, p. 47, 2008

[11] J. L. De La Vara, R. Ali, F. Dalpiaz, J. Sánchez, and P. Giorgini, "Business Processes Contextualisation via Context Analysis," in Proceedings of the ER 2010, 2010, pp. 1–6.

[12] R. Ali, F. Dalpiaz, and P. Giorgini, "A goal-based framework for contextual requirements modeling and analysis," Requirements Engineering, vol. 15, pp. 439–458, 2010.

[13] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, 2000, vol. 5.

[14] E. Cardoso, J. Almeida, G. Guizzardi, and R. S. S. Guizzardi, "Eliciting Goals for Business Process Models with Non-Functional Requirements Catalogues," in: Proceedings of the 10th International Workshop, Bpmds 2009, Held at Caise 2009 June 8-9, 2009,. Amsterdam, the Netherlands: Springer, 2009, p. 33.

[15] C. J. Pavlovski and J. Zou, "Non-functional requirements in business process modeling," in Proceedings of the fifth on Asia-Pacific conference on conceptual modelling (APCCM '08). Australian Computer Society, Inc., 2008, pp. 103–112.

[16] L. Xavier, F. Alencar, J. Castro, and J. H. C. Pimentel, "Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN e NFR," (in English, Integration of Non-Functional Requirements and Business Processes: Integrating NFR and BPMN)) in Proceedings of the 13th Workshop on Requirement Engineering, WER 2010, Cuenca, Equador, 2010, pp. 29–40.

[17] P. Soffer and Y. Wand, "On the notion of soft-goals in business process modeling," Business Process Management Journal, vol. 11, no. 6, pp. 663–679, 2005.

[18] E. Santos, J. Castro, J. Sánchez, and O. Pastor, "A Goal-Oriented Approach for Variability in BPMN," in Proceedings of the 13th Workshop on Requirements Engineering - WER 2010, G. Hadad, O. Dieste, and J. P. Carvallo, Eds., Cuenca, Equador, 2010, pp. 17–28.

[19] E. Santos, J. Pimentel, J. Castro, J. Sánchez, and O. Pastor, "Configuring the Variability of Business Process Models Using Non-Functional Requirements," in Proceedings of the 15th International Conference on Enterprise, Business-Process and Information Systems Modeling., EMMSAD 2010 June 7-8, 2010,Hammamet, Tunisia: Springer Berlin Heidelberg, 2010, pp. 274–286.

[20] M. La Rosa, W. M. P. van Der Aalst, M. Dumas, and A. H. M. ter Hofstede, "Questionnaire-based variability modeling for system configuration," Software and Systems Modeling, vol. 8, no. 2, pp. 251–274, 2009.

[21] A. Lapouchnian, Y. Yu, and J. Mylopoulos, "Requirements-Driven Design and Configuration Management of Business Processes," in Proceedings of the 5th International Conference on Business Process Management, BPM 2007, 2007, pp. 246–261.

[22] D. Greenwood and G. Rimassa, "Autonomic Goal-Oriented Business Process Management," in Third International Conference on Autonomic and Autonomous Systems (ICAS'07). Ieee, Jun. 2007, pp. 43–49.

[23] S. A. White and D. Miers, BPMN Modeling and Reference Guide: Understanding and Using BPMN. Future Strategies Book Division, 2008.

[24] K. Pohl, G. Bockle, and F. J. V. D. Linden, Software product line engineering. Springer, 2005, vol. 49, no. 12.

[25] P. Wohed, W. M. P. van Der Aalst, M. Dumas, A. H. M. ter Hofstede, and N. Russell, "Pattern-based Analysis of BPMN," Tech. Rep., 2005.

[26] Y. Bai, J. Yang, and Y. Qiu, "OntoCBR: Ontology-Based CBR in Context-Aware Applications," 2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008), pp. 164–169, 2008

[27] S. Liaskos, A. Lapouchnian, Y. Yu, E. Yu, and J. Mylopoulos, "On Goal-based Variability Acquisition and Analysis," in Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006, 2006, pp. 92–96.

[28] F. Aburub, M. Odeh, and I. Beeson, "Modelling non-functional requirements of business processes," Information and Software Technology, vol. 49, no. 11-12, pp. 1162–1171, 2007.

[29] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Reasoning with goal models," in Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002). Springer, 2002, pp. 167–181.

[30] T. L. Saaty, "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., vol. 102, no. 2, pp. 251–318, 2008.

[31] J. Castro, M. Kolp, L. Liu, and A. Perini, Dealing with Complexity Using Conceptual Models Based on Tropos. Springer - Verlag Berlin Heidelberg, 2009, pp. 335–362.

[32] F. Dalpiaz, R. Ali, Y. Asnar, V. Bryl, and P. Giorgini, "Applying Tropos to Socio-Technical System Design and Runtime Configuration," *Proceeding of the Italian workshop Dagli OGGETTI agli AGENTI (WOA'08)*, Palermo, Italy: 2008.