A Goal-Oriented Approach for Variability in BPMN

Emanuel Santos, Jaelson Castro
Universidade Federal de Pernambuco Centro de Informatica
Cidade Universitaria. S/N, 50741-000, Recife, Brazil
Email: {ebs, jbc}@cin.ufpe.br

Juan Sanchez, Oscar Pastor
PROS, Universidad Politecnica de Valencia
Camino de Vera S/N, 46022. Valencia, Spain
Email: {jsanchez, opastor}@dsic.upv.es

Abstract

Good business processes need to be up-to-date and automated to represent the organizational environment. To obtain models that reflect the changes it is necessary continuous checking between model and reality. Thus, business process model need to be modifiable and transformable in software. In order satisfy these needs we propose to apply variability analysis over Business Process Models, represented by Business Process Modeling Notation (BPMN), using a Goal-Oriented approach. In this paper we present a high-level process that links different methods in order to describe a systematic way to update BPMN models. Our proposal tries to achieve it by representing variability in goal trees.

Keywords

business process models; goal models; variability modeling;

1. Introduction

The importance of considering the business process in an organization has been recognized by software developers. The software must fit the organization expectation and needs. As consequence, the use of business process models as the source of software development has grown and several process-based languages have been proposed [1],[2].

The Business Process Model and Notation (BPMN) [3] is an example of a language that has become popular among the business analysts. This notation is based on the representation of working process in term of the activities executed and the flow of action and data in the process. The activities and task of a process can be executed by persons or automatically by software. However, sometimes the domain where the software will run is so complex that the business processes as well as the software, which automates some activities of process, change ofte [4]. Some areas such as Software Product Lines (SPL) [5] and Autonomic Computing have employed variability analysis to deal with the change in the environment and software. SPL proposes to separate the common parts from the variable ones in order to systematically reuse them to compose solutions that fit better the environment. On the other hand, the Autonomic Computing proposes using self-adaptation and self-configuration strategies to deal with variable behaviors both at design and at run-time [6].

Some approaches [2], [7] have applied variability analysis to business process models in order to drive the evolution of the business process. However, these approaches have problems to explain how and why a specific instance of process is selected. They do not offer the guidance necessary to the business analyst to configure the process.

In order to represent variability in business process, for example described in BPMN (Business Process Modeling Notation), we propose to adopt Goal models. They are expressive enough to represent the commonality and variability through a structure of AND-OR decompositions [4]. The goal models also can represent functional and non-functional concerns, through the concepts of Hardgoal and Softgoal [8][9]. Moreover, they can explain the rationale behind the selection of a possible solution. Our approach, named GV2BPMN (Goal-Oriented Variability Analysis to BPMN), promotes the use of Goal Models to represent variability in BPMN. The aim is to use Goal models to drive the configuration of Business Process. In doing so, we offer guidance in the configuration of business process and allow a clear understanding of the choices adopted in this activity.

It is worth noticing that some approaches already relate requirements and business processes (described using BPMN [10]). However, they do not consider variability of the business process or of the software. So we try to solve part of this limitation by offering a way to represent variability at the business process level. Thus, by monitoring the evolution of the business processes we can define the (new) appropriate relevant requirements, and as consequence generate quality software, that supports the (new) processes. Hence, handling the evolution of processes over time.

The rest of paper is organized as follow: Section 2 presents background information. Our approach is described in Section 3. A running example is presented in Section 4. Section 5 discusses the most relevant related work and their limitations. Last but not least Section 6 presents conclusions and future works.

2. Background

Our approach addresses variability in business process descriptions using a goal oriented approach. In the sequel we provide some background information.

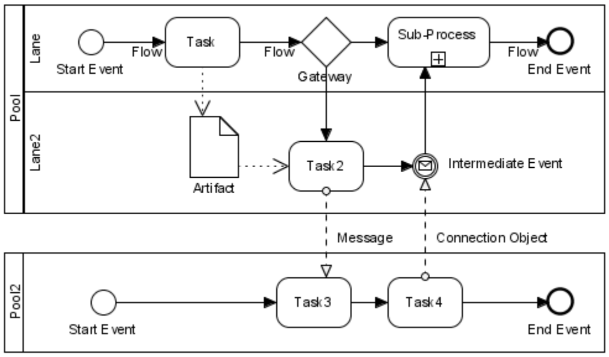


Figure 1. BPMN elements

The Business Process Model and Notation (BPMN) is based in the representation of activities flows that consider the resource, decision making and events. Figure 1 presents the organization in terms of Pool and lanes, inside these elements there are the sequence of activities that are executed in the process represented by Tasks, Flow links, and Sub-Process. The events represent the triggers and stop events of process, such as, Start, Stop and intermediate events. The resources produced or consumed by the process are represented as Artifacts that could be input and output of tasks and sub-processes. Moreover, the communication between different organizations (pools) and people (lanes) is performed by connections of Message.

Variability identification and representation has extensively been investigated in the context of domain analysis [11]. In that context, commonality and variability analysis is aimed at identifying common and varying characteristics among systems that belong to the domain under investigation. Variability also can be stated as the ability to vary and susceptibility to modification under conditions of environment. In software product lines context, the variability could be defined as “the ability of a system, an asset, or a development environment to support the production of a set of artifacts that differ from each other in a preplanned fashion” or “variability means the ability of a core asset to adapt to usages in the different product contexts that are within the product line scope” [5]. One of the major problems with the approaches the deal with variability is how to choose a solution based on the results of variability analysis. In general, there is an independent model that is used for it, named Configuration Knowledge [**?**]. We will use the goal model to represent the information to configure the business process.

Goals capture, at different levels of abstraction, the various objectives that the system under consideration should achieve. They can be formulated from the high-level, strategic concerns, to low-level, technical concerns [8]. Goal-Oriented Requirements Engineering (GORE) is concerned with the use of goals for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements [12]. Goals also cover different types of concerns: functional concerns associated with the services to be provided, and non-functional concerns associated with quality of service, for example safety, security, accuracy, performance, and so forth [9].

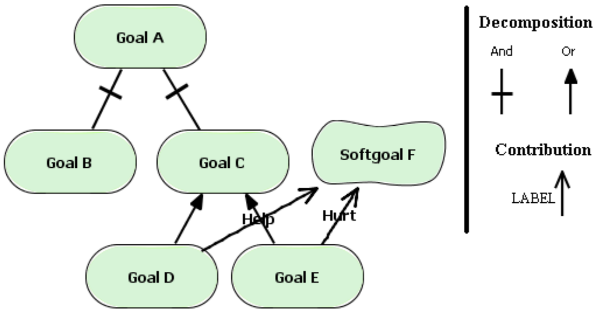


Figure 2. A goal model

Separating stable from more volatile information is an important concern for managing requirements evolution. There are several notations that use Goal as main abstraction such as i* [13], KAOS[8] and NFR Framework [9]. i* uses dependency to describe how the goals could be achieved in a social network. KAOS represents goals as graphs that use hierarchical structure to describe how they can be achieved. NFR defines the notion of softgoals that are goals without a clear cut achievement criteria, that is, the softgoal are qualitative and have different level of satisfaction. In this paper we will adopt a goal graph like the one presented in Fig. 2. This type of model is characterized by the hierarchical decomposition of goals in sub-goals using logical operators such as And, Or and XOr decomposition. Moreover, these models use softgoals to represent the non-functional goals and the contributions links (help, hurt, etc.) to indicate the how the goals can be affected by softgoals.

Goals are used to describe variability as presented in [14]. In these approaches the commonalities are expressed as And-Decompositions and the variability as Or-Decompositions. These models are not compact but have more expressiveness than feature models, frequently used to model software product lines [4].

3. GV2BPMN - Goals-Oriented Variability Analysis to BPMN

In order to deal with the variability of Business Process, here represented by BPMN models, we propose to adopt goal-oriented variability analysis to obtain and represent variants in business process. The following steps are part of GV2BPMN. It starts with the derivation of a goal model from a previous BPMN model. We consider that the organization that will be analyzed already has a business process

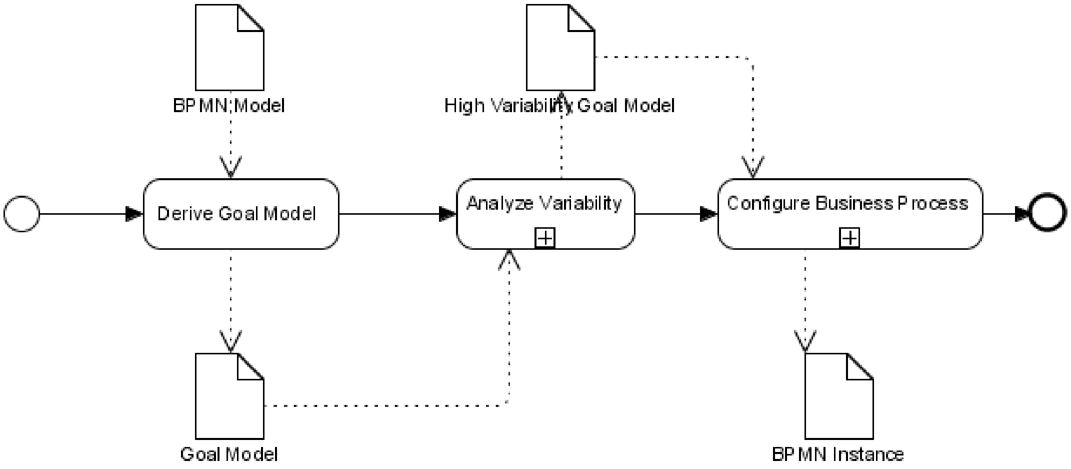


Figure 3. GV2BPMN overview

model that represents its processes. The goal model resultant will be analyzed in further steps to identify and represent the variability.

Fig. 3 presents an overview of the approach. It shows the activities and artifacts produced and consumed. These activities are presented in this Section.

3.1. Step 1. Derive Goal Model

The first step of our approach is deriving a goal model from the BPMN model. From an initial business process model, that can be an instance or an abstract model, we derive a goal model that captures the tasks present in the BPMN model. The derivation is based on previous guidelines [15]. These guidelines describe how the elements of BPMN can be mapped to a goal tree. The BPMN tasks are mapped to goals in the goal model. Gateways are mapped to decomposition of goals. Moreover, decisions gateways are represented by Or-decomposition of a base goal, where each sub-goal represents a possible path of the gateway. Fork gateways are mapped to And-decomposition where each sub-goal corresponds to a path of the gateway. Moreover, a sequence of BPMN tasks is mapped to an And-decomposition where a goal is designed to group the sequence in the same level.

The derivation of a goal model is important because it allows the explicit representation of tacit variability. Observe that in a BPMN model, elements such as sub-processes and gateways can hide domain variability. Hence, BPMN models alone looses important information already identified during the design of process. This information is critical to provide tools to improve the configuration of process, as described in further steps.

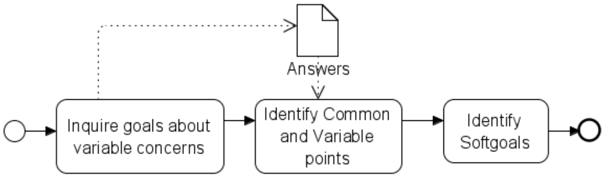

Answers
Inquire goals about variable concerns
Identify Common and Variable points
Identify Softgoals

Figure 4. Sub-process Analyze Variability

3.2. Step 2. Analyze Variability

In this activity the goal model that was derived from the BPMN model is now enriched to represent the variability of the BPMN. The variability is acquired from querying the leaf goals in the goal model about semantics of language. The Fig. 4 presents the activities necessary during the variability analysis.

3.2.1. Step 2.1 Inquire Goals about Variable Concerns. The goal is inquired with questions about the information semantics [14] in order to identify possible alternatives to the default behavior of the goal. The questions help to elicit information such as the agents that execute a task or that are affected by it (Who). How the task is performed, what are the instruments used and what the means are (How). An activity also can consume or produce a resource (What). Time and conditional events can also be concerns of the questions (When). According Liaskos et al. (2006) [14]] several information may be related to a goal. For example, Agentive (Who will do), Dative (Who will be affected by), Objective (What is affected by), Factitive (What are the results of), Process (How it will be done), Location (Where it will be done), Temporal (When it will be done), Conditional (What are the conditions to) and Extent (What are the degrees of). We have structured some questions as described in Table 1. This is an initial set of questions, they need to

Table 1. Template of Questions to inquire the goals

Frame	Questions
Agentive	Who is responsible for do *GOAL*?
Dative	Who will be affected by *GOAL*?
Process	How *GOAL* will be achieved?
	What are the instruments to *GOAL*?
	What are the means to *GOAL*?
Objective	What are the resources consumed by *GOAL*?
Factitive	What are the resources produced by *GOAL*?
Temporal	When *GOAL* will be achieved?
Location	Where *GOAL* will be achieved?
Conditional	What are the conditions to do *GOAL*?
Extent	Which degree of *GOAL* will be achieved?

refined, excluded or complemented if necessary.

3.2.2. Step 2.2 Identify common and variation points. The answers are analyzed to identify what tasks can have more than one acceptable behavior. These tasks will be named variation points, concept borrowed from Software Product Line terminology. The goal model represents the possible variable points using the decomposition mechanisms. The alternative behaviors are represented by XOR decomposition, only one of the tasks can be chosen at time. The optional behavior is represented by the OR decomposition, it means that one or more task can be chosen at time but not all need to be executed. When the execution of task are mandatory or occurs in parallel they are represented by AND decomposition. The decomposition of goals forms a tree that can be evaluated by different strategies (e.g. top-down, bottom-up, middle out), in order to verify the achievement of the main goal. With the identification of common and variable parts it is possible to isolate points (options) that will be chosen (configured).

In this step the relationships among goals are defined for each variable point. Goals in different branches of the decomposition tree can be related, for example to indicate exclusion or dependency among them. If a goal depends on the achievement of another goal, the dependency relationship is defined between them. On the other hand, if the selection of a goal excludes the selection of a goal in other branch of the tree it also is represented in the model. The definition of these relationships helps to control the complexity growth of model, in context of feature models this is known as feature integration.

3.2.3. Step 2.3 Identify softgoals. The softgoals of process are essential in our approach. They are used in further steps as configuration criteria to the BPMN. The softgoals will represent the non-functional requirements of the business process. In Kueng and Kawalek[16] there is a list of goals that are relevant to a business process from the view point of Managers and Performs. Important qualities of Business Process can be represented as softgoals such as: Autonomy, avoid cross-process exchanges; Operational Cost and Maintenance Cost, the process would have high proportion of automated activities; Consistency, the activities of process must be consistent among them. Moreover, other softgoals can be defined based in domain information. For instance, if an activity of process has strong correlation with Usability then it can be included in the model as presented in Xavier (2009) [17].

Through the contribution relationship we can identify the degree of interference among goals and softgoals. If a goal contributes to reduce the cost or increase the efficiency of a process, it can be modeled explicitly. This information helps to point out the rationale behind the configuration of business process. Once that several ways to execute a business process were modeled the softgoals can drive the decision about the configuration that fits better the organization in a determined moment.

3.3. Step 3 Configure the Business Process

In GV2BPMN the goal model is used to drive the configuration of business process. The third step is responsible by restructure the business process original based in the variability model represented by the goal model. In Software Product Lines this is known as Configuration Knowledge, which consists of external information to configure (define) the (concrete) products of a SPL.

The previous steps identified the variability present in the business process but do not modify the initial business process. The configuration process will generate instances of process that combine the initial process and the variations identified.

The strategy to configure the process can be Bottom-Up, i.e. when the business analyst chooses the goals instances that will be part of the process, and then uses propagation algorithms to identify how they affects the softgoals. Or Top-Down, when a softgoal is selected and the goals that contribute positively for it achievement are selected to be in the process. However, conflicts may arise, when for example different goals contribute both positively and negatively to the same softgoal. This kind of problem can be minimized, or even avoided during the variability analysis step, if more elaborate goal relationships are defined (such as exclusion and dependency among goals). Note that a third alternative could be to considered, a middle-out approach.

The selection can be organized in two different ways, goal or package-based. In the goal based strategy the selection is made by the business analyst using one of the strategies mentioned before. For each goal, the business analyst can select those that will be part of the process. On the other hand, the selection can be driven to definition of products as it happens in product lines. For example, each set of goals that represent a specific process can be grouped as a package and the selection be based this group. This

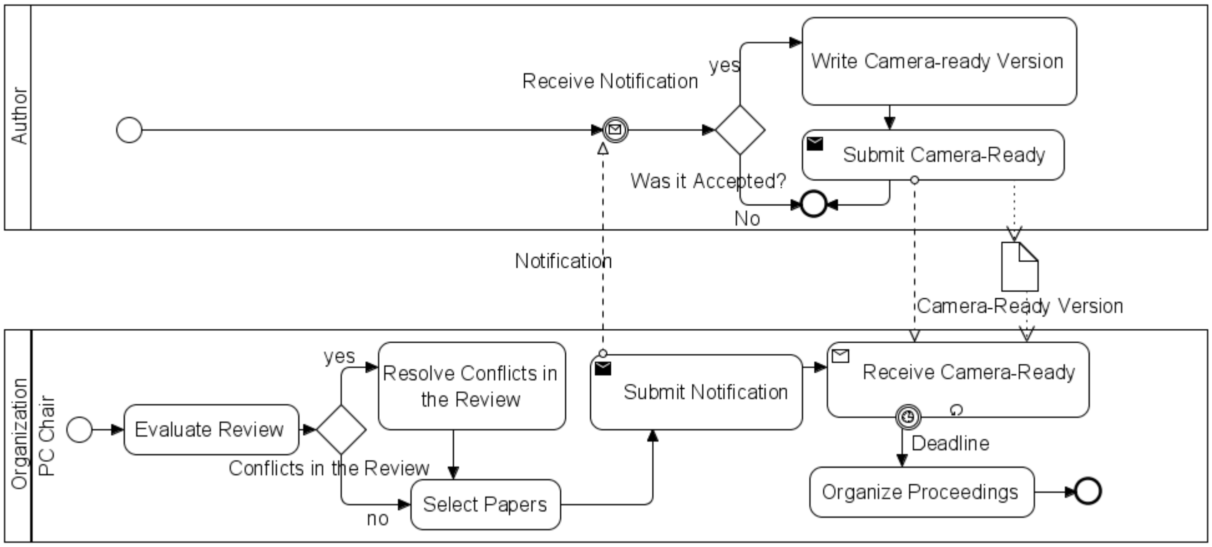


Figure 5. Part of Conference Management Example in BPMN

kind of configuration is less flexible than the one based on goals, since it reduces the variability space. That is, a smaller number of packages can be configured in comparison with the number of combinations derived from goal based strategy.

Once the goals are selected, now the problem is to obtain a new BPMN model from them. Recall, that in the first step of our process, the goals were derived from the BPMN (using guidelines where each task was related to a goal). Since we know that the goals are related with BPMN tasks, it is necessary identify how to replace it for tasks that correspond to the variations identified in the goal model.

The configuration of the business process and as consequence the generation of a new (concrete) instance of the business process must to consider the goals selected, the relationships among them, and the kind of selection used. For instance, if a goal has three variations they can be represented in the BPMN as tasks, lanes or events. However, the way how they will be disposed depends on the kind of variation it is (agentive, conditional, etc), if it requires or excludes other goals, and if the configuration will be conducted by each goal or by a set of goals.

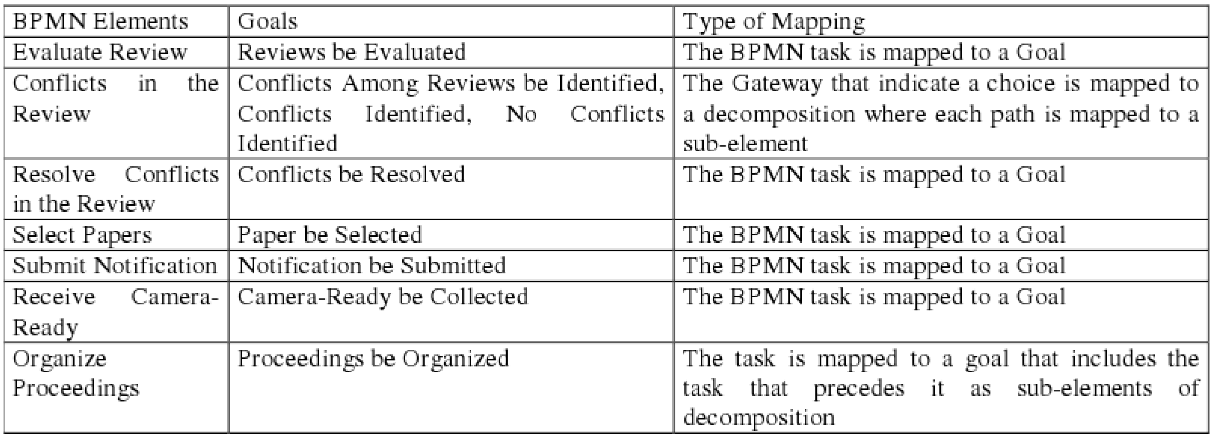
The selection by package implies that a pre-defined set of goals will be used to derive the instance of process. .The way how they will be disposed is defined before generate the new instance. It means that when a goal is selected all the goals of the same package are selected too and they must preserve the arrangement among them. In the goal-based selection, for each goal selected it is necessary to derive the tasks that will be part of process. Note that the links among the tasks must be defined individually for each case.

To build a package the following steps can be used. First, assign a label for the variations of goals that are part of a possible instance. After that, identify if there are conflicts between goals of a same label. If possible, avoid the conflicts by substituting a variation by other that is not in conflict with the previous ones. And then, verify if the goals of a label contribute to the softgoals that are important for that process. These steps can be repeated until obtain a closed set of goals for the label.

The bottom-up analysis is preferred for the construction of packages once the goals need to be selected early. In the goal based strategy both bottom-up and top-down analysis can be applied. Our approach does not provide an automatic way to reassemble the BPMN from the Goal model. The goal model drives this process but the knowledge of the analysis still is necessary. However, to help in this process we propose some heuristics: Variations of a goal can be mapped to a sequence if there are dependency relationships among them (the one that depends on comes after), or to a fork gateway if no dependency is present; Variations of agents (i.e., agentive and dative frames) are mapped to lanes in the BPMN model. The goals assigned to these agents are mapped to tasks in their lanes;

If there is no interference among goals of different branches they can be mapped to sequences of tasks and gateways in the inverse process of the Step 1. AND-decompositions to task sequences, reading the model from left to right. AND-decompositions that represent decisions are mapped to decisions gateways. OR-decompositions are mapped to fork-gateways. The variations referent to conditions and time can be represent by intermediary events of BPMN that correspond to the meaning of the information.

Table 2. Correspondence among goals and BPMN elements.



BPMN Elements	Goals	Type of Mapping
Evaluate Review	Reviews be Evaluated	The BPMN task is mapped to a Goal
Conflicts in the Review	Conflicts Among Reviews be Identified, Conflicts Identified, No Conflicts Identified	The Gateway that indicate a choice is mapped to a decomposition where each path is mapped to a sub-element
Resolve Conflicts in the Review	Conflicts be Resolved	The BPMN task is mapped to a Goal
Select Papers	Paper be Selected	The BPMN task is mapped to a Goal
Submit Notification	Notification be Submitted	The BPMN task is mapped to a Goal
Receive Camera-Ready	Camera-Ready be Collected	The BPMN task is mapped to a Goal
Organize Proceedings	Proceedings be Organized	The task is mapped to a goal that includes the task that precedes it as sub-elements of decomposition

4. Running Example

The approach is presented with the help of a running example, i.e. the Conference Management System. The context is that of a conference organization where a committee responsible to organize a conference (e.g., PC Chair, PC Member) interacts with the Author in the process of submission, evaluation and publication of papers.

Depending of the type of conference, several configurations of this process can be defined. In this paper we start with a simple configuration just to present how the approach could work. In this example the management of a conference includes several phases such as the submission of a paper, its evaluation and notification of results, and finally the organization of the proceedings. As the process is large to be presented in a single picture we present just a part of process. Figure 5 shows the part of notification of acceptance that occurs at the end of the conference organization process. In this part, the papers that were evaluated by the reviewer are judged and the acceptance or rejection notification is sent to the author.

The first step is deriving the goal graph that represents the business process model. As a result a tree-like goal representation is generated. The BPMN tasks were related to goals, the precedence of tasks in the flow was represented by AND decomposition. For instance, the goals derived are grouped bellow the Proceedings be Organized goal which means that the high level goal requires the achievement of its offspring. According to their type the gateways were mapped to AND, OR, or XOR decompositions. In Figure 5 the Conflicts in the Review gateway is a point of choice. Hence, the derived goal tree was represented by XOR decomposition where the sub-elements corresponds to the alternative paths represented in the BPMN. Table 2 shows the BPMN Elements and the goals derived from them and their sources. The Figure 6 presents the final goal model. This artifact can be useful for the configuration of processes. In the sequel, we focus on just the Submit Notification task and the Notification be Submitted goal.

The next step of process consists of eliciting and analyzing the variability. First, the goals that are leafs in the goal model are selected to be analyzed. In our example the Notification be Submitted goal was selected and analyzed. The goal inquired is based on questions according to the template described in Table 1. We identified that the notification can be done by two agents: the PC Chair and the CMS (Agentive). Other facets also were identified, such as the way that the Notification will be sent to the author (by E-mail or by posting in the CMS), or the condition that triggers the submission of the notification (when the deadline arrives or when the all reviews are available). Some questions were excluded from the set because they did not seem to be relevant. For instance, the question What are the resources produced by Notification be Submitted? or What are the degree of Notification be Submitted?. These cases are not relevant in the analysis since the former will be response with only one answer (the notification itself), and the latter will not present an answer because the notification cannot be submitted in degree. Once the possible alternative behaviors were elicited the next step is to identify the relationships between the goals. For each goal recognized as a variable goal, the variation is analyzed as well as possible interference among them. When a goal requires another goal, the relationship is marked as requires, for example: the Notification be posted in the CMS system requires that the CMS be the agent of the submission. On the other hand, if a goal forbids other to be achieved then the relationship is marked as excludes. In this example in particular we do not present an excludes relationship.

In order to drive the configuration is necessary to define the softgoals that are important to the process. In our example the softgoals identified were Cost and Availability, see

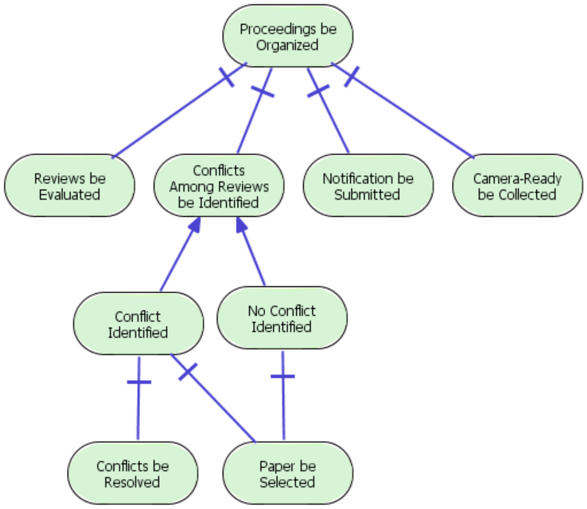


Figure 6. Goal model derived from the Proceedings organization sub-process of the Conference Management Example.

Figure 7. Both of them refer to the Conference Organization Domain. Cost refers to the implementation of the solution, since small conferences do not demand high Cost and the large ones can compensate the Cost with other qualities. The Availability refer to the capacity of the information be available to the ones that need it. For instance, if the conference uses asynchronous communication means then if some problem happens to the notification of acceptance the organization and authors will have problems. Other softgoals could be indentified to other goals but to illustrate the process these two were enough.

The contribution analysis allows represent the relationship between the softgoals and the goals. The softgoal Cost can be affected negatively by the implementation of the CMS system and by the use of it, for instance the variations by CMS and by posting in the CMS has a negative contribution to Cost. In our analysis we are not considering the impact of the system in the cost of the process, which could present different result since automated process reduces the cost of operation of process. On the other hand, the variations by PC Chair and by e-mail help to minimize the Cost because they do not require a high investment to became operational. The other softgoal identified was the Availability, that can be affected by the mean as the notification is submitted and by the persons that will receive it. Thus, the variation by e-mail and to First Author contribute negatively to the Availability, since the e-mail can not be lost and communicate with just one person reduces the chance to the communication be received. The variations by posting in CMS and to All Authors contribute positively to the Availability. With a system that allows post the notification on a web site the author become the responsible to access the information. Moreover, with the all authors receiving the notification the risk of lost the information is smaller than send to only one. With these softgoals in consideration the Cost and Availability can be used to choose the best configuration for determined process. The instance can prioritize on of them or use them to trade-off analysis if the variations affect in different ways the softgoals.

Once we have finished the analysis over the Business Process the next step consists in configure instances that fits the preferences of the process for a determined criteria. In our example we used the strategy to select goal-by-goal, which is obvious since we are focusing in just one goal. With it in mind we adopt a bottom-up analysis. In the Figure 8 we present two possible configurations to the process.

The first one, Fig 8 (a), is a configuration where the variations of the goal Notification be Submitted selected were submit by e-mail to all authors by the PC Chair when the deadline arrives. This configuration prioritizes the softgoal Cost since send the notification by PC Chair and using e-mail contribute positively reduce the Cost. The variation to All Authors was select because contributes positively to Availability and does not affect the Cost. The variation when the deadline arrives was select without concern about the result since the condition does not affect the softgoals in our analysis to the process.

The other configuration possible is present in the Figure 8 (b). In this configuration the Notification be Submitted was executed by the system CMS using its own means, the notification was address to all Authors and was done when the deadline arrived. The in this configuration the priority was the Availability them the goals were selected and the impact of the Availability evaluated in the solution. The variations that contribute positively to Availability were select the variations by posting in the CMS and to All Authors were selected. Since the variation by posting in the CMS requires that the system itself be present in the solution the variation by CMS (Agent concern) were selected even it means a negative impact over the Cost. Finally, the variation when the deadline arrives was selected by the same reasons that it occurs in the instance of Fig 8 (a).

With the configuration selected, the process of obtain instances resides in transform from the goal model to the BPMN instance. The heuristics presented in the end of the step 3 (see Section 3) helps in this process. First, we need to identify what will change from the original BPMN model (Fig. 5) to the instances (Fig. 8). The dashed circles in the Fig. 8 highlight the points where the instances are different from the original.

Variations in the agentive or dative frames result in the addition or elimination of lanes in the BPMN model, then select the submission to all actors results in specialize the pool Author into the First Author and Collaborator Author lanes. Part of the tasks that were present in the original one need to be replicated and reorganized, in this case was the

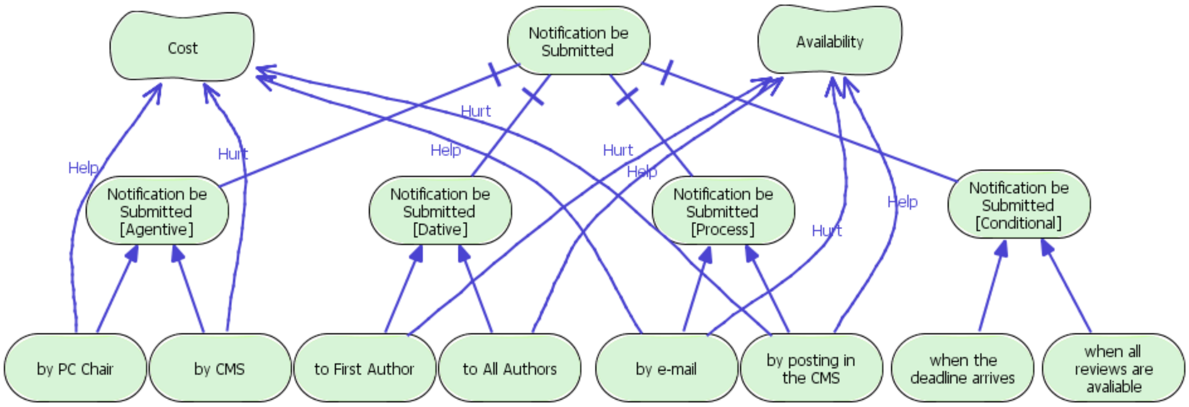

Cost
Notification be Submitted
Availability
Hurt
Help
Hurt
Hurt
Help
Help
Help
Hurt
Notification be Submitted [Agentive]
Notification be Submitted [Dative]
Notification be Submitted [Process]
Notification be Submitted [Conditional]
by PC Chair
by CMS
to First Author
to All Authors
by e-mail
by posting in the CMS
when the deadline arrives
when all reviews are avaliable

Figure 7. The goal Notification be Submitted with the variations.

event Receive Notification that was replicated to the both lanes. The addition of another author also is present in the Fig. 8 (b), where the selection of the variation by CMS also results in the addition of the lane CMS in the pool of the conference Organization. In the latter, this was required because a new task was added to the CMS, the task Publish Notification, which corresponds to the variation by posting in CMS.

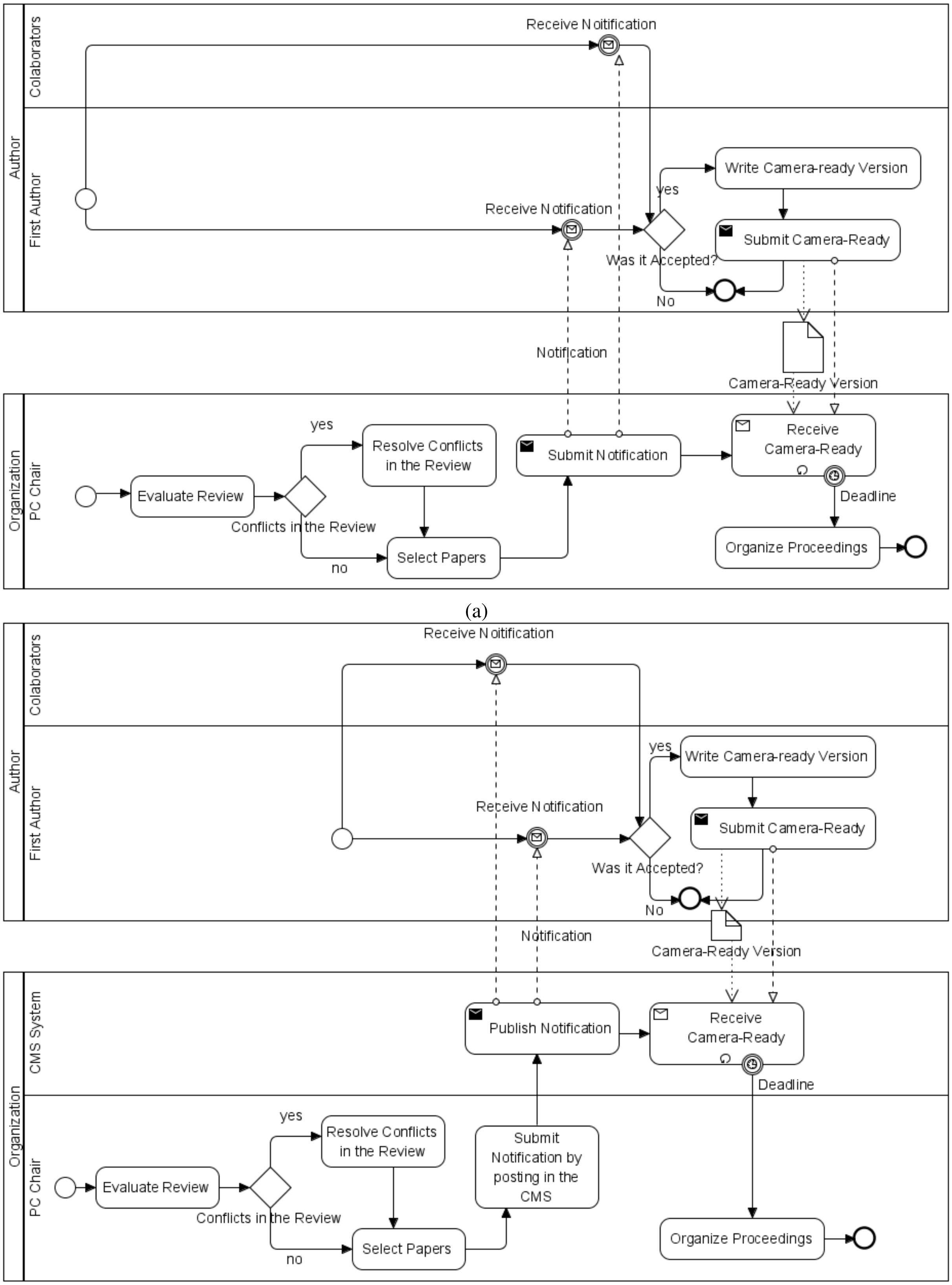
The conditions were represented by intermediary events, in the example the event Deadline after the task Select Papers to represent the variation when deadline arrives.

The other variations substitute the original tasks by the new ones that are arranged according their sequence. In the Fig 8 (a) the sequence (Deadline, Submit Notification by E-mail) substitutes the task Submit Notification. As Deadline is a condition to start (e.g., trigger) it was placed before the other variations. In the Fig 8 (b) the sequence (Deadline, Submit Notification by CMS, Publish Notification) substitute the task Submit Notification.

5. Related Work

In Lapouchinian et al.[18] there is an approach that represents business process in terms of its goals. Variability rich business processes are modeled using goal graphs. As the goal graphs are not expressive enough to represent flow and sequence, they apply annotation in the model in order to cover this gap. The aim of this approach is obtain configuration mechanisms that reflect the business process. The result is a configuration mechanism that abstracts the complexity of configuring software from the end-users. Their approach can generate business process (described in BPEL) based transformation of the goal model. Schnieders and Puhlmann [7] present a mechanism to represent high variability business process models using BPMN. In this approach they present mechanisms to represent variability in flow-based languages. They rely on extension, inclusion, parameterization and design patterns. These mechanisms enrich the BPMN model and allow represent variability with a specific representation for each type of variability. They propose using feature models to obtain the variability but do not explain how to do it. Moreover, their approach is focused on the process itself, without consider the requirements phase. Montero et al. [2] describe a methodology to obtain and represent variability in business process models, represented by BPMN language. They are concerned with the derivation of requirements for software related the business process. To represent the business process they adopt feature models and use cases model to describe requirements. The selection mechanism is the selection of features, then if a feature needs to be present in the solution it is selected and the model is restructured to support the changes. As formalism to do it they adopt finite state machines.

Based in the related work we identify some interesting point to investigate. The first point is the use of goal models as an alternative to the feature models as occurs in Schnieders and Puhlmann[7] and Montero et al. [2]. We are trying to adopt a single abstraction to represent both business process activities and requirements. This will help to maintain the traceability and seamless between the models, moreover offers an elegant representation of non-functional requirements through the softgoal concept. Moreover, Lapouchinian et al. [18] adopts annotations into the goal models. However, we are avoiding this kind of solution because we believe that it can compromise the readability of model. Instead we propose to represent the flows in the BPMN model and adopt traceability tables to link the elements of BPMN model and the goals. Finally, we are concerned with business process models that are ease to be updated. The combinatorial explosion of solutions can make it hard to select the elements manually. Thus, instead of a manual selection mechanisms as Schnieders and Puhlmman[7], we prefer adopt a semi-automatic one, similar to proposed by Lapouchinian et al.[18].



(a)

(b)

Figure 8. Instances of BPMN model with the variations selected

6. Conclusion

In this paper we presented an approach to include variability in business process models through goal models. Based on a Business Process Model, describe in BPMN, we derive goal models that are used to analyze variability and then obtain business processes that can be redesigned to meet the needs of the user. This approach promotes the linking of the business process models and the goals models keeping the trace between them.

In the example we derived two instances of a previous business process using as selection criteria softgoals. One of them present a software system, the CMS, which automates some tasks of the process. Using business process is possible identify and requirements to the information system that will automate the activities of a organization [1]. In this ways, the GV2BPMN could be understood as early activity of the requirement engineering. However, at moment we are not trying to deal with the variability in requirements itself since it is a more complex issue that can be investigated in future versions of the approach.

6.1. Future Works

Further developments are required. First of all, to connect the selected methods we adopt ad hoc transformations between the models, it was necessary in some steps of our approach. A future work is formalizing the transformations from a model to another to prove the accuracy of the transformation. It also includes the development of a traceability reference model to keep the relevant traces stored and be easily recoverable. Then, we need to extend or complete an exiting variability analysis method to surpass the existing limitations [14]. Requirements models can be included in the approach, which could result in a way to traces from requirements to business process even the process change through the time. Finally, to be useful to the business analysts we need to develop tool support to the phases of our approach due to the complexity of models.

Acknowledgment

This paper was partially supported by CNPQ and CAPES.

References

[1] J. L. D. La Vara, J. Sanchez, and O. Pastor, "Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems," in *Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE'08)*, 2008, pp. 213–227.

[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)*, Anonymous, Ed. IEEE Computer Society Press, 2008, pp. 605–608. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ICCBSS.2008.28

[3] C. O. OMG, "Business Process Model and Notation (BPMN) - Specification v1.2," 2009.

[4] Y. Yu, A. Lapouchnian, S. Liaskos, J. Mylopoulos, and J. C. S. D. P. Leite, "From goals to high-variability software design," *Lecture Notes in Computer Science*, vol. 4994, p. 1, 2008. [Online]. Available: http://www.springerlink.com/index/qt4121v3ur347g7w.pdf

[5] F. Bachmann and P. C. Clements, "Variability in software product lines," Software Engineering Institute, Pittsburgh, USA, Tech. Rep. September, 2005.

[6] C. Cetina, P. Giner, J. Fons, and V. Pelechano, "Autonomic computing through reuse of variability models at runtime: The case of smart homes," *Computer*, vol. 42, pp. 37–43, 2009.

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

[8] A. V. Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*, Anonymous, Ed. John Wiley & Sons, 2009.

[9] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 1999, vol. 5.

[10] J. L. D. La Vara and J. Sanchez, "Business process-driven requirements engineering: a goal-based approach," in *Proceedings of the 8th Workshop on Business Process Modeling*, 2007.

[11] R. Prieto-Diaz, "Domain Analysis: An Introduction," *ACM SIGSoft Software Engineering Notes*, vol. 15, no. 2, pp. 47–54, 1990.

[12] A. van Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," in *Proceedings of 5th IEEE International Symposium on Requirements Engineering*, Anonymous, Ed., 2001, pp. 249–263.

[13] E. Yu, J. Castro, and A. Perini, "Strategic Actors Modeling with i*," in *Tutorial Notes, 16th Intl. Conf. on Requirements Engineering*. Spain: IEEE Computer Society, 2008, pp. 01–60.

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

[15] J. L. D. La Vara and J. Sanchez, "Derivacion de modelos de tareas a partir de modelos BPMN," in *Proceeding of 1st Taller sobre Procesos de Negocio e Ingeniería del Software (PNIS 2007)*, Anonymous, Ed., 2007.

[16] P. Kueng and P. Kawalek, “Goal-based business process models: creation and evaluation,” *Business Process Management Journal*, vol. 3, no. 1, pp. 17–38, 1997.

[17] L. Xavier, “Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN e RNF,” MSC, Universidade Federal de Pernambuco, 2009.

[18] A. Lapouchnian, Y. Yu, and J. Mylopoulos, “Requirements-Driven Design and Configuration Management of Business Processes,” in *Proceedings 5th International Conference Business Process Management, BPM 2007*, Anonymous, Ed., 2007, pp. 246–261.