An Aspect-Oriented Framework for Business Process Improvement

Alireza Pourshahid[1], Gunter Mussbacher[1], Daniel Amyot[1], Michael Weiss[2]

[1] SITE, University of Ottawa, 800 King Edward Ave
Ottawa ON, K1N 6N5 Canada
[2] SCE, Carleton University, 1125 Colonel By Drive, Ottawa, ON, K1S 5B6 Canada
apour024@uottawa.ca, gunterm@site.uottawa.ca, damyot@site.uottawa.ca, weiss@sce.carleton.ca

Abstract. Recently, many organizations invested in Business Process Management Systems (BPMSs) in order to automate and monitor their processes. Business Activity Monitoring is one of the essential modules of a BPMS as it provides the core monitoring capabilities. Although the natural step after process monitoring is process improvement, most of the existing systems do not provide the means to help users with the improvement step. In this paper, we address this issue by proposing an aspect-oriented framework that allows the impact of changes to business processes to be explored with what-if scenarios based on the most appropriate process redesign patterns among several possibilities. As the four cornerstones of a BPMS are process, goal, performance and validation views, these views need to be aligned automatically by any approach that intends to support automated improvement of business processes. Our framework therefore provides means to reflect process changes also in the other views of the business process. A health care case study presented as a proof of concept suggests that this novel approach is feasible.

Keywords: Business Process Management, Aspect-Oriented Modeling, User Requirements Notation, Process Improvement, Process Redesign, Process Modeling.

1 Introduction

Business Process Management (BPM) has recently gained momentum among e-business technologies. BPM can be realized through methodologies, techniques, or software, in a way that helps organizations bring together processes and their context including people, documents, information sources, organizational structures, and applications [1]. As a methodology, BPM helps organizations gain control over their business processes by modeling, validating, analyzing, and monitoring the processes. BPM provides process visibility for the organizations, and hence makes both human-centric and electronic-centric processes more manageable [2].

A BPM methodology is typically an iterative lifecycle composed of several steps, which usually starts with modeling and validation of the business processes. The next steps in the lifecycle are the automation and execution of these processes. Then, the processes are monitored and finally, based on the monitoring results, they may be redesigned and improved to better achieve the expected goals [3]. In addition to such methodologies, Business Process Analysis (BPA) ontologies have been suggested to make the analysis effort more effective and to reduce the gap between IT and the business world [4]. Furthermore, there has been some work done to capture common improvement approaches in the form of business redesign patterns that contribute to the improvement of processes from four main perspectives, namely time, quality, cost, and flexibility [5]. Although these redesign patterns could be used as a guideline for the improvement step, most of the improvement methods do not utilize these patterns. On the contrary, these methods rely heavily on human innovation and creativity rather than on rationality. Using patterns can help to further rationalize and formalize process improvement methods [6].

BPM as software is often called a Business Process Management System (BPMS). Existing BPMSs, such as Appian enterprise BPM suite, G360, Tibco iProcess Suite, EMC BPM Suite, and Fujitsu Interstage BPM Suite [7], provide various methods for process monitoring. Monitoring in these systems is usually done by defining calculation rules for specific measurement points which quantify important business concerns called Key Performance Indicators (KPIs) [8]. However, these systems usually do not provide the means for process improvement. In addition, they do not support the application of process redesign patterns. Therefore, improvements can be done solely based on human knowledge and experience [9].

Furthermore, available process modeling notations such as BPMN, UML, EPC, YAWL, and IDEF3 do not provide means to observe or simulate the impact of such patterns on KPIs and business goals before the patterns are implemented in the system [10, 11]. They also do not allow the comparison of candidate redesign patterns with respect to their impact on high-level business goals. Therefore, support for effectively selecting the most appropriate pattern for the current business situation is very limited. Essentially, these features are not supported because there is no method to automatically reflect changes to the process model in the KPIs and business goal models – possibly even based on historical expectations. Consequently, a change to the process model requires all the other related models to be tracked and modified manually, which can be a tedious and error-prone task.

We are proposing an aspect-oriented framework based on the User Requirements Notation (URN) standard [12] to address the aforementioned issues. URN is an integrated process/goal modeling notation that was recently extended for business process monitoring [13]. Four complementary views for a business problem can be defined with the help of URN, including a process view, a goal view, a validation view, and a performance view [3]. We can use these views to monitor a process and identify process deficiencies. Aspect-oriented URN (AoURN) [14], on the other hand, extends URN with aspect-oriented concepts and can be used to define redesign patterns in a modular way, as aspects. Aspects can be merged with the existing process model to explore the impact of applying a redesign pattern on business goals, and such changes can be easily undone if the desired process improvements do not materialize, thus lowering the barrier to apply redesign patterns. Technically, AoURN's pointcut expressions define criteria that describe the parts of the process to which the redesign patterns should be applied. AoURN also provides the capability to define corresponding aspects for each of the pointcut expressions and apply them to the process model. Applying the aspect is equivalent to applying the redesign pattern, thus modifying the process and replacing the appropriate parts with a new model. Moreover, AoURN provides similar functionality for the goal and performance views. Using such an approach, one can apply several process redesign patterns to a process, align the goal/performance views accordingly, and compare the results to select the best overall pattern based on the impact of redesign patterns on overall business goals.

This paper contributes to the body of knowledge in several ways. First, we provide an innovative method for selecting the most appropriate redesign pattern among several candidate patterns, considering the impact of the patterns on process performance and business goals. Second, we further enhance the existing URN based process monitoring framework with a repository of redesign patterns that supports the exploration of proposed changes with what-if scenarios based on these patterns. Third, our suggested approach allows for dynamic and continuous adaptation of business processes to current business needs based on the defined KPIs. Finally, this work introduces the first example of AoURN pointcut expressions that span goal and process models. This allows us to not only define related aspects on goal and process views but also to apply them to both views at the same time based on criteria formulated not just for one view but for both types of views. Validation of the framework is done in part through a health care case study.

Although our approach currently concentrates on the capabilities of URN to elaborate our novel concepts and illustrate them, the ideas presented here could be used in other contexts and as a generalized methodology.

The rest of this paper is organized in four main parts. In section 2, we elaborate on the background knowledge required to understand the new concepts introduced in this paper including BPM using URN as well as AoURN. In section 3, we describe the new aspect-oriented framework for business process improvement. In section 4, we illustrate the application of this framework in a health care case study. Finally, the last section discusses our conclusions and future work.

2 Background and Related Work

URN-based Business Process Management, process redesign patterns, and the Aspect-oriented User Requirements Notation provide the basis for the framework introduced in this paper, and we elaborate on these concepts in sections 2.1 and 2.2.

2.1 BPM with URN

The User Requirements Notation (URN) is an International Telecommunication Union standard for capturing early requirements in the form of scenarios and goals [12, 15]. URN consists of two complementary sub-languages called Goal-oriented Requirement Language (GRL) and Use Case Maps (UCM) for goal modeling and scenario modeling, respectively [15]. In this paper, we will introduce basic notation elements as we go.

Fig. 1 and Fig. 2 illustrate a hospital's Data Warehouse Approval Process used as a case study in this paper. Fig. 1 is a GRL model consisting of GRL intentional elements linked together using contribution links. Elements are called intentional because they carry stakeholder intentions. Contribution links indicate the impact of intentional elements on each other – in this case, positive qualitative contributions are shown. Three types of intentional elements have been used in this model. Soft goals (⬭, e.g., Reduce Costs) describe something to be achieved that cannot be measured quantitatively but is of a qualitative nature, tasks (⬡, e.g., Approval Process) model potential solutions for achieving higher-level goals, and KPIs (⬡, e.g., Number of Mistakes) indicate metrics of the system normalized to a scale of -100 to 100.

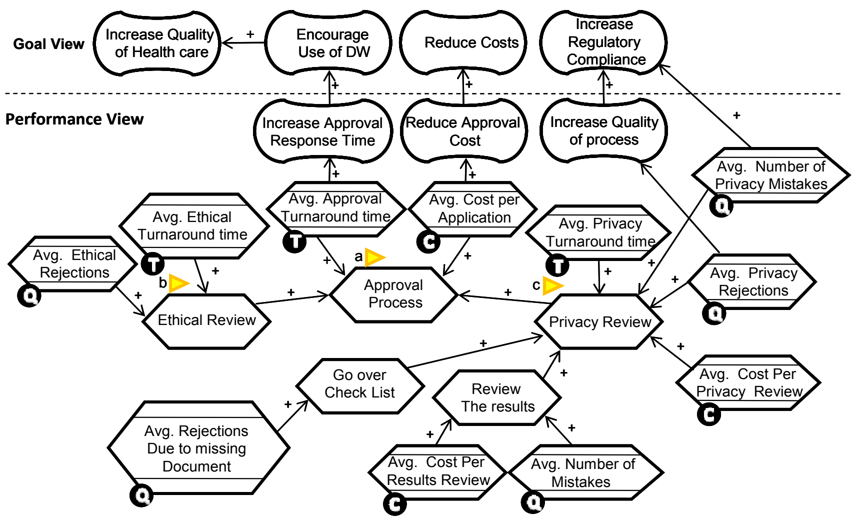

Fig. 1. Data Warehouse Approval Process – Goal and Performance View

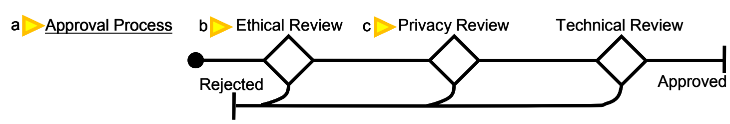

Fig. 2. Data Warehouse Approval Process – Process View

In addition, URN traceability links (▶) are used to relate tasks in this goal model to their representation in the UCM model, Fig. 2. The Approval Process is linked to the whole diagram, while the other two tasks are linked to individual model elements. The UCM model provides further behavioral details about the linked tasks. It consists of a path that begins at a start point (●) and ends with an end point (❚). Stubs (◇, e.g., Ethical Review) are containers for sub-models and denote here the three major steps in the process. Drilling into a stub leads to a submap which provides more details about the step. Note that in this example, each stub has two possible exits: one if a review approved the data warehouse access request and one if it was rejected.

Business process modeling using URN was introduced in [16] and [17] by modeling, analyzing, and evolving a supply chain management system. While the current version of the URN notation still requires better support for exception and cancellation handling in process and workflow models [18], its unique capabilities for modeling both processes with UCMs and goals with GRL in a unified way is a significant advantage over other process modeling notations [10]. The integrated view of UCM and GRL not only answers the *where*, *what*, *who*, and *when* questions of process models, but also answers *why* a particular part of a process exists. Using URN, people with sufficient business knowledge and experience can align business goals and processes [19].

Business process modeling using URN was introduced in [16] and [17] by modeling, analyzing, and evolving a supply chain management system. While the current version of the URN notation still requires better support for exception and cancellation handling in process and workflow models [18], its unique capabilities for modeling both processes with UCMs and goals with GRL in a unified way is a significant advantage over other process modeling notations [10]. The integrated view of UCM and GRL not only answers the *where*, *what*, *who*, and *when* questions of process models, but also answers *why* a particular part of a process exists. Using URN, people with sufficient business knowledge and experience can align business goals and processes [19].

jUCMNav [20] is the most comprehensive URN modeling tool available today. By formalizing the data exchange layer [21], external information systems such as data warehouses can be connected to the jUCMNav tool. Integration with external tools including a Requirement Management System (RMS), DOORS [22], and a Business Intelligence System (BIS), Cognos [9], as well as extending URN [13] helped with the development of an integrated BPM framework [3] for process validation [23] and process performance monitoring [11].

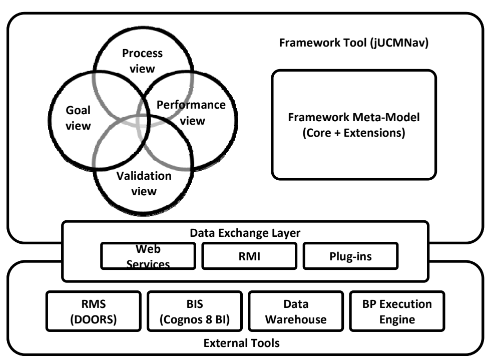

Fig. 3. Framework Core Views and Components

The developed BPM framework consists of four main views and several components (see Fig. 3). The process view captures the workflow and the sequence of steps in the business process, from very high levels of abstraction down to the task and responsibility level describing the atomic parts of the process. The goal view captures the business goals related to the process. Goal modeling can focus initially on high-level strategic goals of the business, which are later decomposed into low-level operational goals and even tasks. It is common to see the same tasks shared in both the low-level goal view and the low-level process view. Therefore, the process and goal views can be associated together, defining which part of a process impacts which business goals. In addition, the performance view introduced in [11] is associated with both the process and goal views and illustrates how processes perform with respect to the business goals using Key Performance Indicators. Finally the validation view introduced in [24] defines the requirements and restrictions against which the process view should be validated. For instance, corporate policies, laws, or service level agreements can be considered as some of these validations criteria that need to be satisfied by the process view.

The BPM framework uses the built-in evaluation mechanisms of URN to evaluate high-level business goals and other validation criteria based on the satisfaction levels of low-level goals and the values of KPIs. For example, monitoring the Data Warehouse Approval Process (see Fig. 1) yields process measures that in turn result in initial satisfaction values for the KPIs. These values are then propagated to higher-level nodes in the goal graph until the highest-level goals have been evaluated.

The BPM framework is an iterative and incremental approach with several steps for business process improvement. In the first step, the target processes for improvement are selected. This selection can be done based on the priorities of the organization. Then, the artifacts required for the improvement, including the four views, are modeled and the association links between the views are established. Subsequently, the dimensional data sources used for monitoring are prepared and the performance of the processes is monitored. In the alignment step, the views are modified to address the issues observed from the monitoring step. The framework provides guidelines required for process improvement by suggesting the necessary artifacts and context information for rational analysis of the processes, thus identifying possible improvement points. Although, this framework has been designed to use the process redesign patterns in the alignment step, it does not provide the means for helping analysts to select the best design pattern according to the process status and organization goals. The selection of the pattern still requires the analysts to go through all the patterns and heavily relies on their knowledge and expertise.

In this paper, we move toward providing help for the analysts in terms of selecting and applying the design patterns. Generalized redesign patterns as suggested in [5] as well as customized alternatives appropriate for specific processes can be utilized. The generalized patterns discussed in [5] can help with process improvement in four categories including cost, time, quality, and flexibility. Table 1 shows the three redesign patterns used in this paper as examples. A complete list of the redesign patterns with their impact on the four categories is available in [13].

Table 1. Sample Redesign Patterns and their Impact on the Four Categories

Redesign Pattern	Time	Cost	Quality	Flexibility
Knockout	⇣⇡	↑	**N/A**	⇣
Task Elimination	↑	↑	⇣	**N/A**
Control Relocation	**N/A**	**N/A**	↑	**N/A**

↑: Positive Impact ↓: Negative Impact ⇡: Maybe Positive Impact ⇣: Maybe Negative Impact

2.2 Aspect-Oriented Modeling with the User Requirements Notation

The Aspect-oriented User Requirements Notation (AoURN) is a modeling framework that extends URN with aspect-oriented concepts [14], allowing modelers to better encapsulate crosscutting concerns which are hard or impossible to encapsulate only with URN models. With AoURN, business process redesign patterns can more easily be encapsulated as concerns in their own modules and selectively applied to the existing process. AoURN treats concerns as first-class modeling elements. AoURN groups all relevant properties of a concern such as goals, behavior, and structure, as well as pointcut expressions needed to apply new goal and scenario elements to a base model or to modify existing elements. A *pointcut expression* is a pattern that must be matched in the base model if the aspect is to be applied, thus determining the base model locations to which the aspect is applied.

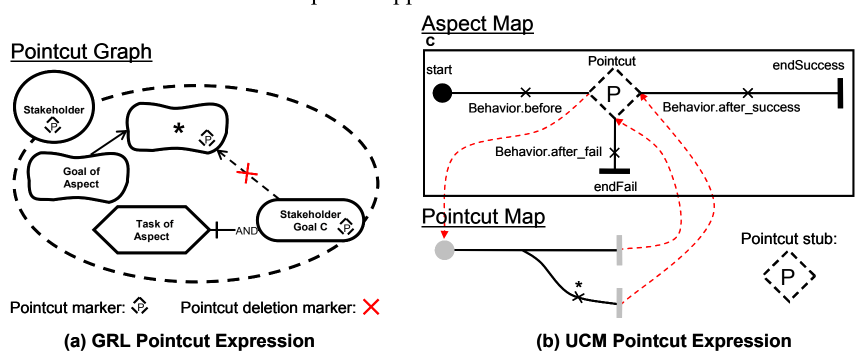

Fig. 4. GRL and UCM Pointcut Expressions

AoURN adds aspect concepts to URN's sub-notations, leading to Aspect-oriented GRL (AoGRL) [25] and Aspect-oriented UCMs (AoUCM) [26, 27]. AoURN uses standard URN diagrams to describe pointcut expressions (i.e., it is only limited by the expressive power of URN itself as opposed to a particular composition language). GRL pointcut expressions are shown on a *pointcut graph* and make use of *pointcut (deletion) markers* to indicate the pattern to be matched (see Fig. 4.a). All elements without pointcut markers are added to the matched location in the GRL model, while elements with a pointcut deletion marker are removed. Goals and tasks of an aspect may be described in more detail in separate goal graphs called *aspect graphs*.

UCM pointcut expressions define the pattern to be matched with a *pointcut map* (see Fig. 4.b). The aspectual properties are shown on a separate *aspect map*. The aspect map is linked to the pointcut expression with the help of a *pointcut stub*. The causal relationship of the pointcut stub and the aspectual properties visually defines the composition rule for the aspect, indicating how the aspect is inserted in the base model (e.g., before, after, instead of, in parallel, interleaved, or anything else that can be expressed with the UCM notation).

AoURN employs an aspect composition technique that can fully transform URN models. Locations affected by an aspect are indicated by *aspect markers* (see Fig. 5). When an aspect marker is selected, the modeler is taken to an aspect view with only those aspectual properties highlighted that are relevant to the selected aspect marker.

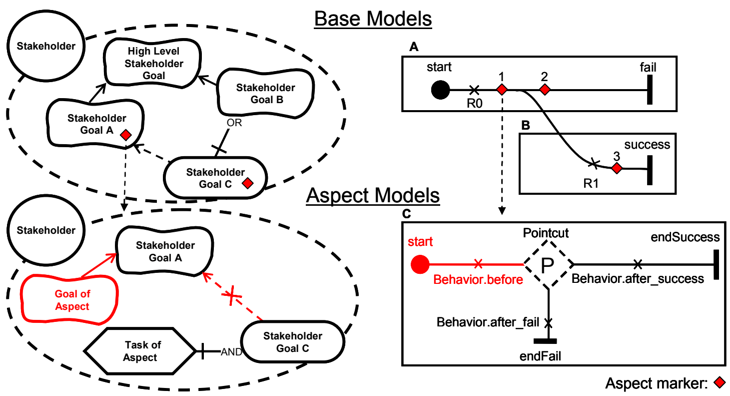

Fig. 5. Composed Model with Aspect Markers

3 Framework

The intention of the suggested framework is to improve processes that do not satisfy their goals by applying the most appropriate redesign pattern. Since this framework builds on the BPM framework introduced in section 2.1, one has to first define the three vital artifacts required for process monitoring (i.e., the process, goal, and performance views). Furthermore, the process redesign patterns must be modeled as aspects which are later applied to the existing process by defining pointcut expressions. Customized redesign patterns or alternative versions of the processes may be modeled by the users of the framework in addition to standard redesign patterns. To aid the selection of the appropriate redesign pattern, KPIs are categorized according to four redesign pattern groups (i.e., cost, time, quality, and flexibility). The filled circles with T, Q, and C in Fig. 1 indicate the time, quality, and cost categories for the KPIs, respectively.

As part of the performance modeling step, each KPI is normalized. The worst possible value, a threshold value, and the target value are defined for real world values of each KPI and then mapped to -100, 0, and 100 on the GRL scale, respectively. Now, the business processes may be monitored. Any KPI value that is not satisfactory (i.e., that is far off the target value) indicates a possible area of improvement. The KPI categories help determine the candidate redesign patterns. For instance, if the KPI with the unsatisfactory value is categorized as a time KPI, all redesign patterns with a positive impact on time are possible candidates for improving the observed values of the KPI and consequently the process. However, all of the possible candidate redesign patterns may not be applicable to the target process for which the KPI has been defined. Therefore, in the next step, we use additional characteristics of redesign patterns to reduce the number of candidates. These characteristics are identifying features of the redesign pattern that are expressed with AoURN pointcut expressions, e.g., the redesign pattern may require a certain sequence of process model elements or may require that certain KPIs perform worse than other KPIs. A redesign pattern, therefore, is only applicable to the process model, if a match of its pointcut expression can be found in the process model. After identifying the applicable patterns and if more than one possible choice exists, users can apply all the possible options one by one and decide which one is the most appropriate. As an applied aspect (i.e., pattern) changes not only the process view but also the goal and performance views, it is possible to observe the impact of the applied pattern on all views. The aspectual model of a redesign pattern may even introduce up-to-now unidentified goals to the process model, which the redesign pattern helps to achieve but have not yet been considered by the modeler, contributing further to a more comprehensive process model.

The advantage of modeling redesign patterns with aspects is that the whole pattern, i.e., the characteristics of the pattern but also its impact on the process, goal, and performance views, can be modeled in one properly encapsulated unit. This facilitates the reuse of the pattern in different applications and enables reasoning about the use of the pattern and its composition with other patterns.

A new concept introduced in this paper is that pointcut expressions span goal and process models. Until now, all AoURN pointcut expressions [14, 25, 26, 27] are either just in the process view or just in the goal view of a URN model. We, however, need to cover both goal and process models a) because redesign patterns require both views to be matched to properly describe the identifying characteristics of the pattern and b) to ensure that all views remain aligned with each other. This provides us with the ability to apply the required changes to goal and performance views after applying the redesign pattern to the process view. Such changes in the goal or performance views are required when the redesign pattern eliminates, adds, or updates tasks in the process view. Therefore, the same tasks and the corresponding KPIs should be added, eliminated, or updated in the goal and performance views, respectively.

4 Case Study

The case study is based on the real Data Warehouse Approval Process of a health care provider that assesses requests for access to the health information in the data warehouse based on patient privacy concerns, ethical concerns, as well as technical feasibility and impact.

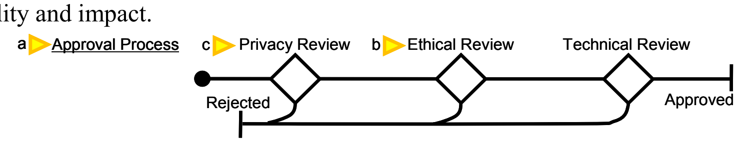

Fig. 6. Improved Data Warehouse Approval Process – Process View

The Knockout redesign pattern reorders a sequence of tasks based on their failure rate and effort. It therefore can be applied to the Data Warehouse Approval Process (see Fig. 2) if the average number of rejections caused by the Privacy Review is higher than for the Ethical Review and the Ethical Reviews takes longer to complete than the Privacy Review. In that case, the Privacy Review should be moved ahead of the Ethical Review to the first task in the sequence (see Fig. 6).

The aspectual model for the Knockout redesign pattern captures in a generic way the constraints of the Knockout pattern as described in the previous paragraph. First, the GRL pointcut expression (GRL graph at the right hand side of Fig. 7) stipulates that there is a value of a KPI in the time category that is not satisfactory (i.e., the top KPI in the GRL pointcut expression as indicated by the T in the circle and <satisfaction, respectively). *Satisfaction* is a value defined by the modeler, usually somewhere between 0 and 100 on the GRL scale. As 0 represents the threshold value and 100 represents the target value, this captures the main premise of the Knockout redesign pattern. As it positively impacts the time category, it should be applied in a situation where a KPI from the time group is not performing as desired.

Furthermore, the unsatisfactory KPI is connected to a task (the one with URN link 3) which is further connected with two other tasks (the ones with URN links 4 and 5). All three tasks are traced with the help of the URN links to the two stubs and the map in the UCM pointcut expression (the map at the bottom left in Fig. 7). Therefore, additional properties from the UCM model must be satisfied for a successful match of the pointcut expression and for the Knockout pattern to be applied. The URN links indicate that the top level task is described by a map and that the other two tasks appear as stubs on the map (i.e., the two other tasks are a refinement of the higher-level task). More specifically, the UCM pointcut expression indicates that the redesign pattern applies to a series of two stubs (i.e., two process steps) that either succeed or fail. Note that the dashed portion of the pointcut expression matches against any sequence of UCM modeling elements and therefore can be matched against the join after the Privacy Review stub in Fig. 2.

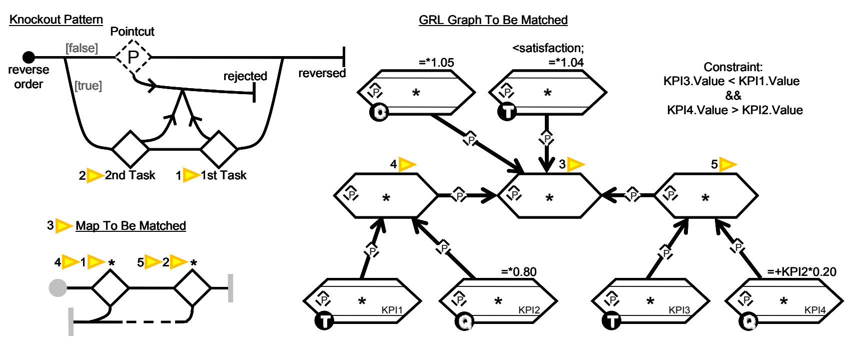

Fig. 7. Generic Aspectual Model for the Knockout Redesign Pattern

At this point, the pointcut expression would match against a large number of consecutive stubs with two out-paths as long as the overall task associated to the map of the two stubs has a time KPI with an unsatisfactory value. To further improve the accuracy of the Knockout pattern description, further matching criteria are defined for the two other tasks. Note that the task with URN link 4 is the first task and the task with URN link 5 is the second task in the sequence as defined by the UCM pointcut expression and the URN links. Each of the two tasks has two more KPIs connected, one from the time category and one from the quality category. Constraints for these four KPIs state that:

- Constraint A: the value of KPI1 (the time KPI of the first task) must be higher than the value of KPI3 (the time KPI of the second task);
- Constraint B: the value of KPI2 (the quality KPI of the first task) must be lower than the value of KPI4 (the quality KPI of the second task).

This reflects the characteristics of the Knockout redesign pattern as it applies only to situations where the second task takes less time to do (time KPI is lower) but results in rejections more often (quality KPI is higher) than the first task. If this is the case, then it is advantageous to move the second task ahead of the first task. This change to the process is described by the Knockout Pattern map (the UCM at the top left in Fig. 7). This map describes the aspectual behavior to be applied if the pointcut expression can be matched for the existing process. AoURN describes a replacement of the matched model elements with an OR-fork with [false] and [true] branches. The [false] branch describes what is being replaced (i.e., the matched model elements represented by the pointcut stub), while the [true] branch describes the new behavior. On the [true] branch the order of the two stubs from the pointcut expression are switched. URN links between the Knockout Pattern map and the pointcut expression allow matched elements to be reused in the Knockout Pattern map (i.e., the aspectual scenario).

Finally, the GRL pointcut expression in Fig. 7 also defines the anticipated impact of applying the Knockout pattern on the performance model by describing the changes to the satisfaction values for matched KPIs. The annotation =*1.04 in Fig. 7 indicates that the unsatisfactory KPI is expected to improve by 4%. These changes may be even based on historical data. For example, assuming that 20% of the submitted requests that fail ethical review also fail the privacy review, then the matched quality KPI of the ethical review (KPI2) will decrease by 20% (=*0.8) while the matched quality KPI of the privacy review (KPI4) will increase by the same number (=+KPI2*0.2), if the order of the two reviews is reversed. Furthermore, there is also an impact on the average cost of the approval process, as more requests will now be rejected earlier in the process leading to a cost decrease.

When the aspect is applied to the URN model of the Data Warehouse Approval Process, a match is found in the performance model. Approval Process, Ethical Review, and Privacy Review match against the three tasks. Avg. Approval Turnaround time, Avg. Ethical Turnaround time, Avg. Ethical Rejections, Avg. Privacy Turnaround time, Avg. Privacy Rejections, and Avg. Cost per Application match against the six KPIs. Furthermore, the map linked to Approval Process and the two stubs linked to Ethical Review and Privacy Review match the UCM pointcut expression.

The pointcut expression for the Knockout redesign pattern is very generic and therefore uses only parameterized elements. This may lead to undesired matches. If this is the case, then the pointcut expressions can be tailored to the specific needs of the current situation. For example, the * for the task with the URN link 3 could be replaced with Approval Process to narrow down the search space.

Since the pointcut expression is matched in the Data Warehouse Approval Process, the aspectual behavior is added to the process (see Fig. 8). Therefore, aspect markers are added before and after the matched model elements as defined on the Knockout Pattern map. AoURN uses slightly different aspect markers to indicate that the aspect replaces existing model elements. Solid bars are added to the aspect markers to denote tunnel entrances and exits. The aspect marker before the Ethical Review stub is a tunnel entrance as the behavior does not continue with the Ethical Review but with the aspectual behavior and only returns to the map at the tunnel exits (i.e., the two other aspect markers). Note how the two stubs on the Knockout Pattern map have been replaced by the matched model elements from the pointcut expression. The resulting model in Fig. 8 is semantically equivalent to the model in Fig. 6.

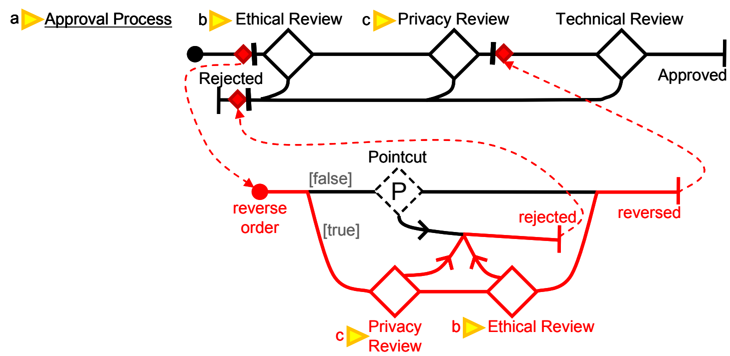

Fig. 8. Applied Knockout Pattern

Two further patterns have been applied to the sub-processes for submitting an application by the researcher and the privacy review by the privacy review board. Fig. 9 shows the "As Is" process and the "To Be" process after the application of the task elimination and control relocation patterns. The first pattern is often used to remove process steps without significant value. In this case, we have removed Review the Results since the average number of mistakes found by this review is low (see Table 2). The second pattern, on the other hand, is often used to move input validation checks to the client side [5]. In this process, we have moved Go over Check List to the researcher's application submission process. Due to space limitations, we cannot illustrate the aspect models for these two redesign patterns in this paper.

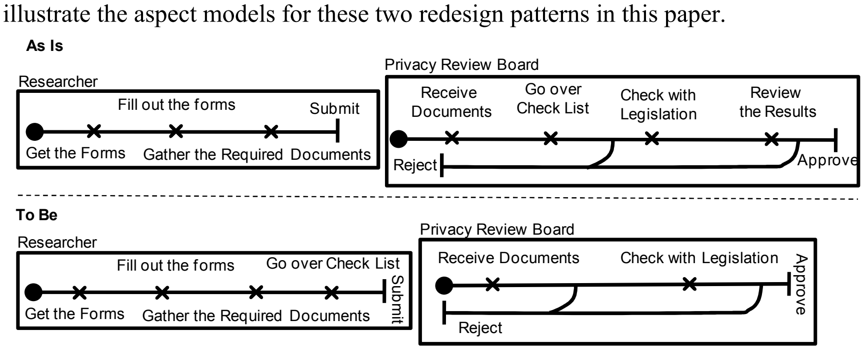

Fig. 9. Applying Task Elimination and Control Relocation to Two Sub-Processes

Table 2 shows the impact of all three patterns on the process performance and business goals as calculated by the GRL evaluation mechanism. Although the applied patterns have a positive impact on the Approval Process and Privacy Review performance, they have also a negative impact on the business goals Increase

Regulatory Compliance and Increase Quality of Process. In most cases when redesign patterns are applied, positive and negative impacts occur in parallel [5]. Our suggested approach equips business analysts with the ability to observe and explore these impacts, leading to more informed decisions about the process.

Table 2. KPI, Goal, and Task Evaluations Before and After Applying the Three Patterns

KPI	SV B	SV A	EV B	EV A	Pattern
Avg. Approval Turnaround Time	7	9	20 d	18.3 d	K – E – CR
Avg. Cost per Application	10	20	4100$	3800$	K – E – CR
Avg. Ethical Rejection	90	72	10%	8%	K
Avg. Ethical Turnaround Time	50	50	10 d	10 d	N/A
Avg. Privacy Rejection	80	98	20%	22%	K
Avg. Number of Privacy Mistakes	75	25	0.5%	1.5%	E
Avg. Privacy Turnaround Time	33	66	5 d	4 d	E - CR
Avg. Cost per Privacy Review	50	90	1000$	800$	E - CR
Avg. Number of Mistakes	100	N/A	1%	N/A	E
Avg. Cost per Results Review	-50	N/A	200$	N/A	E
Avg. Rejections due to Missing Docs	-50	75	10%	2%	CR

Goal/Task	SV B	SV A	Pattern	Impact
Increase Quality of Health Care	-3	2	K – E – CR	Positive
Encourage Use of DW	-4	3	K – E – CR	Positive
Increase Approval Response Time	-6	5	K – E – CR	Positive
Reduce Cost	-5	11	K – E – CR	Positive
Reduce Approval Cost	-7	15	K – E – CR	Positive
Increase Regulatory Compliance	74	69	E	Negative
Increase Quality of Process	99	93	E	Negative
Approval Process	30	99	K – E – CR	Positive
Ethical Review	12	12	N/A	N/A
Privacy Review	46	99	E – CR	Positive

SV: Satisfaction value – EV: Evaluation value – B: Before applying redesign pattern – A: After applying redesign patterns
K: Knockout – E: Elimination – CR: Control Relocation
Note: Pattern column indicates which pattern has caused changes in the KPI.

5 Conclusion and Future Work

Although applying redesign patterns to business processes is a natural way to achieve process improvements, existing Business Process Management and Monitoring systems do not provide the means to do so. Business processes, however, have to be improved based on monitoring results to adapt to changes. Furthermore, most business process modeling notations do not support impact analysis of business goals when process model change, since modeling of goals and the relationship between the goals and processes is not supported in the first place.

We have addressed the latter problem in our previous research [10, 11, 13] and further enhanced our proposed methodology in this paper to address the first problem. This paper proposes a framework for the automated suggestion of business process redesign patterns based on monitoring results. Process redesign patterns are modeled individually as aspects which allows for the patterns to be added to and removed from a business process without requiring significant changes to the process models that are difficult to undo, thus enabling the exploration of what-if scenarios. In order to select the most appropriate patterns, several applicable patterns may be applied and their results compared in terms of their impact on the business process and the business goals with the help of the GRL evaluation mechanism.

In our approach, we have utilized AoURN's pointcut expressions and URN KPI groups to model the defining characteristics of a pattern that may be observed in any or all of the process, performance, and goal views. The defining characteristics must be matched before an aspect can be applied to the process model. When the aspect (i.e., a pattern) is applied to the process model, the process, goal, and performance views are updated simultaneously and therefore remain synchronized. To the best of our knowledge, pointcut expressions that span various types of models as required in our approach are a novel idea for AOM (Aspect-oriented Modeling).

In future work, we intend to further automate the selection of the best pattern by applying all applicable patterns and comparing their impact on the process, goal, and performance views. Furthermore, the framework can likely be used to support adaptive business processes and architectures. Additional experiments on real process modifications will also enable us to better validate this approach and further generalize it to be adapted to other modeling languages.

Acknowledgments. This research was partially supported by NSERC through its programs of Discovery Grants and Postgraduate Scholarships.

References

1. van Herk, D.: Business Activity Monitoring Buzz or Business. Master Thesis Information Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands (May 2006)
2. Debevoise, T.: Business Process Management with a Business Rules Approach. Business Knowledge Architects, Canada (2005)
3. Pourshahid, A., Chen, P., Amyot, D., Peyton, L., Ghanavati, S., Weiss, M., Forster, A.: Toward an integrated User Requirement Notation framework and tool for Business Process Management. In: International MCETECH Conference on e-Technologies, pp. 3-15. IEEE, Washington (2008)
4. Pedrinaci, C., Domingue, J., Medeiros, A.: A Core Ontology for Business Process Analysis. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) The Semantic Web: Research and Applications. LNCS vol. 5021, pp. 49-64. Springer (2008)
5. Reijers, H.: Process Design and Redesign. In: ter Hofstede, A.H., Dumas, M., van der Aalst, W.M. (eds.) Process-Aware Information Systems, pp. 207-234. John Wiley & Sons, Inc., Hoboken, New Jersey, USA (2005)
6. Forster, F.: The Idea behind Business Process Improvement:Toward a Business Process Improvement Pattern Framework. http://www.bptrends.com/publicationfiles/04-06-ART-PatternFramework-Forster.pdf (accessed February 2009)
7. Bruce Silver Associates: BPMS Watch Ratings for Q2 2008, http://www.bpminstitute.org/bpmsreport.html (accessed January 2009)
8. Kronz, A.: Managing of Process Key Performance Indicators as Part of the ARIS Methodology. In: Scheer, A.-W., Jost, W., Heβ, H., Kronz, A. (eds.) Corporate Performance Management, pp. 31-44. Springer (2006)

9. Chen, P: Goal-oriented business process monitoring: An approach based on user requirement notation combined with business intelligence and web services. MSc. thesis, SCS, Carleton University, Ottawa, Canada (2007)
10. Pourshahid, A., Chen, P., Amyot, D., Peyton, L., Ghanavati, S., Weiss, M., Forster, A.: Business Process Management with User Requirement Notation. To appear in: Electronic Commerce Research, Springer (2009)
11. Pourshahid, A.: A URN-Based Methodology for Business Process Monitoring. MSc. thesis, EBT, University of Ottawa, Ottawa, Canada (2008)
12. ITU-T: Recommendation Z.151 (11/08): User Requirements Notation (URN) - Language definition. ITU-T, Geneva, Switzerland (2008)
13. Pourshahid, A., Chen, P., Amyot, D., Weiss, M., Forster, A.: Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools. In: 10th Workshop on Requirement Engineering (WER'07), pp. 80-91. Toronto, Canada (2007)
14. Mussbacher, G.: Aspect-Oriented User Requirements Notation: Aspects in Goal and Scenario Models. In: Giese, H. (ed.) Models in Software Engineering: Workshops and Symposia at MoDELS 2007, LNCS 5002, pp. 305-316. Springer (2008)
15. Amyot, D.: Introduction to the User Requirements Notation: Learning by Example. Computer Networks 42(3), 285-301 (2003)
16. Weiss, M., Amyot., D.: Designing and Evolving Business Models with URN. In: Montreal Conference on E-Technologies (MCETECH), pp. 149-162. Montreal (2005)
17. Weiss, M., Amyot, D.: Business Process Modeling with URN. International Journal of E-Business Research 1(3), 63-90 (2006)
18. Mussbacher, G., Amyot, D.: Assessing the Applicability of Use Case Maps for Business Process and Workflow Description. In: International MCETECH Conference on e-Technologies, pp. 219-222. IEEE CS, Washington (2008)
19. Liu, L., Yu, E.: Designing information systems in social context: a goal and scenario modelling approach. Information Systems 29(2), 187-203 (2004)
20. Roy, J.-F., Kealey, J., Amyot, D.: Towards Integrated Tool Support for the User Requirements Notation. In: Gotzhein, R., Reed, R. (eds.) System Analysis and Modeling: Language Profiles, LNCS vol. 4320 pp. 198-215. Springer (2006)
21. Kealey, J.: Enhanced Use Case Map Analysis and Transformation Tooling. MSc. thesis, SITE, University of Ottawa, Ottawa, Canada (October 2007)
22. Roy, J.-F.: Requirement Engineering with URN: Integrating Goals and Scenarios. MSc. thesis, SITE, University of Ottawa, Ottawa, Canada (March 2007)
23. Ghanavati, S: A compliance Framework for Business ProcessesBased on URN. MSc. thesis, SITE, University of Ottawa, Ottawa, Canada (May 2007)
24. Pourshahid, A., Peyton, L., Ghanavati, S., Amyot, D., Chen, P., Weiss, M.: Model-Based Validation of Business Processes. To appear in: V. Shankararaman, J.L. Zhao and K.K. Lee (eds.) Business Process Management: Concepts, Technology, and Application. Advances in Management Information Systems, M.E. Sharpe Inc. (2009)
25. Mussbacher, G., Amyot, D., Araújo, J., Moreira, A., Weiss, M.: Visualizing Aspect-Oriented Goal Models with AoGRL. In: Second International Workshop on Requirements Engineering Visualization. IEEE, Washington (2007)
26. Mussbacher, G., Amyot, D., Weiss, M.: Visualizing Early Aspects with Use Case Maps. In: Rashid, A., Aksit, M. (eds.) Transactions on Aspect-Oriented Software Development III. LNCS vol. 4620, pp. 105-143. Springer (2007)
27. Mussbacher, G., Amyot, D., Whittle, J., Weiss, M.: Flexible and Expressive Composition Rules with Aspect-oriented Use Case Maps (AoUCM). In: Moreira, A., Grundy, J. (eds.) Early Aspects: Current Challenges and Future Directions. LNCS vol 4765, pp. 19-38. Springer (2007)