"A Rule-Based Approach for Model Management in a User Interface - Business Alignment Framework"

Sousa, Kenia Soares ; Mendonça Filho, Hildeberto ; Vanderdonckt, Jean

ABSTRACT

When organizations change Business Processes (BP) aiming for improvements, such changes normally impact on systems' User Interfaces (UI), which represent a tangible resource for communication with customers, suppliers, partners, or investors; thus a source of competitive advantage and differentiation. To manage the link between UIs and BPs, we propose a framework that classifies the core elements and the operations performed on them, represented through rules that are used to support impact analysis. This solution has been analyzed in a large bank-insurance organization, which has enabled the proposal of an innovative strategy that integrates researches on interaction design and business process management with implications on practical scenarios of result-driven organizations.

CITE THIS VERSION

Sousa, Kenia Soares ; Mendonça Filho, Hildeberto ; Vanderdonckt, Jean. *A Rule-Based Approach for Model Management in a User Interface - Business Alignment Framework.*Task Models and Diagrams for user Interface Design. 8th International Workshop, TAMODIA 2009 (Brussels, Belgium, 23-25 September 2009). In: England, D.; Palanque, P.; Vanderdonckt, J.; Wild, P.J.;, *Task Models and Diagrams for user Interface Design. 8th International Workshop, TAMODIA 2009*, Springer-verlag2010, p. 1-14 http://hdl.handle.net/2078.1/67438 -- DOI : 10.1007/978-3-642-11797-8_1

A Rule-Based Approach for Model Management in a User Interface – Business Alignment Framework

Kenia Sousa, Hildeberto Mendonça, and Jean Vanderdonckt

Université catholique de Louvain, Louvain School of Management,
Place des Doyens, 1, 1348 Louvain-la-Neuve, Belgium
{kenia.sousa,hildeberto.mendonca,
jean.vanderdonckt}@uclouvain.be

Abstract. When organizations change Business Processes (BP) aiming for improvements, such changes normally impact on systems' User Interfaces (UI), which represent a tangible resource for communication with customers, suppliers, partners, or investors; thus a source of competitive advantage and differentiation. To manage the link between UIs and BPs, we propose a framework that classifies the core elements and the operations performed on them, represented through rules that are used to support impact analysis. This solution has been analyzed in a large bank-insurance organization, which has enabled the proposal of an innovative strategy that integrates researches on interaction design and business process management with implications on practical scenarios of result-driven organizations.

Keywords: Business Process Modeling, Model Driven Engineering, Interaction Design, User Interface Markup Language.

1 Introduction

In the competitive market, organizations are now, more than ever, struggling to innovate in order to provide added value to customers through their services and products. To corroborate this, the 2007 Aberdeen Report states that 46% of over 3,600 surveyed enterprises are managing changes to improve BPs with an additional 39% planning to do the same [1]. BPs enable innovation because they are tangible representations of strategic decisions translated into executable operations. Since BPs are executed in practice through enterprise systems they must be aligned to enable organizations to execute changes with flexibility.

However, it is no longer only system processing time that is pivotal for large organizations to improve their services for customers. A consultant from the Center of Excellence in Process-Efficient Technology has presented results of a study, conducted with 29 companies that had their enterprise systems developed aligning business processes with UIs and 23 companies with systems developed using traditional approaches. It has shown that the alignment primarily contributed to 100 to 200 percent user productivity gains [10].

Several recent strategies to align BPs with Information Technology (IT) and model-driven UI development address impacts on IT infra-structure and on the functional core

D. England et al. (Eds.): TAMODIA 2009, LNCS 5963, pp. 1–14, 2010.

with solutions that automate the generation of systems to provide alignment. On the other hand, there are several organizations that already have their BPs and systems and cannot afford to re-develop their solutions with new technologies, but need to analyze impact of changes on their BPs and systems. Therefore, we have innovated by aligning IT with BPs through UIs, originally calling the term *UI-Business Alignment*, which analyzes impact of changes independent of the way BPs are structured and modeled, and how UIs are designed and developed. Here, the concept of *alignment* [9] addresses the *fit* between the external business arena and the internal structure and the *integration* of BP with UIs by considering how choices made on UIs impact those on the business domain and vice-versa.

One of the main innovative aspects of UI-Business alignment is that it focuses on business process actors, who are also system users, instead of uniquely on the systems, which has been extensively done so far. Our framework identifies impacts in terms of UIs that are understood by anyone who interacts with systems. On the other hand, traditional IT-Business alignment strategies present impact analysis in terms of system architecture, such as web services or classes, which are understood only by specialized professionals (e.g. system analysts, software architects).

This paper presents the main concepts of the models involved in the UI-Business Alignment (section 2). These concepts are the core elements of the rules that represent the foundation to maintain the mappings between the models (section 3). We present related works (section 4) and conclude with a summary and future work (section 5).

2 Model-Driven Approach for Traceability

The UI-Business alignment framework is composed of a methodology that has core actions to be performed by stakeholders who need to make sure that what is specified in BPs are executed by system end-users and what is suggested by users is considered in the business context. Among these actions, the traceability is possible by associating BPs with users' tasks, associating users' tasks with UIs, simulating impact of changes in any of these models, etc. This methodology can be combined with other corporate methods and processes, such as software development processes, HCI methods, process improvement methodologies, IT-Business alignment strategies, etc.

The UI-Business alignment framework adopts a model-driven approach for UI design, in which BPs, task models and UI models are mapped. We support the association of task models with UI models through relevant model-driven UI design works. Mapping the UI models is supported by UsiXML [21], a UI definition language that represents models in a structured form and supports the flexibility necessary for defining model-driven UIs; and by the Cameleon Reference Framework [7], extended with business process modeling. Following, we explain the link between these models that form the foundation for the proposed traceability strategy.

2.1 Business Process Models and Task Models

BP is a structured set of activities, performed by organizations' stakeholders, designed to produce a specific output for a particular customer or market [8]. Task models describe how tasks can be performed to reach users' goals when using systems. Task models are designed in a task hierarchical structure that contains different levels of abstraction, which starts with tasks in the highest level that are further decomposed in sub-tasks in the intermediary levels until the lowest level.

BP could not be directly associated to UIs because they represent the business context and some of their characteristics make them a limited representation for UI design, namely: 1) concepts in BPs do not consider automation in itself. The one responsible for applying the process decides what should be automated, not exactly how the activity is performed. 2) BPs do not encompass tasks intrinsic to user interaction (e.g. cancel, save temporarily, undo). 3) In most cases, the BP is not detailed enough to describe individual behavior and even when it is, the sequence of activities may not represent the user behavior, strongly influenced by the context of use. However, the BP structure is suitably similar to the hierarchical structure in task models, which influenced the use of task models as a bridge between BP and UIs.

2.2 Task Models and UI Models

To define user interaction, user tasks contain essential information to conceive UIs, a means by which users interact with the system. There are approaches that focus on mapping task and UI models to generate UIs. Paternò & Santoro [14] specify the relationships between task model and AUI, and between the AUI and its implementation. Brown et al. [6] specify the links between task hierarchy, task-oriented specification, software architecture and code documents aiming at improving co-evolutionary design. Vanderdonckt [21] defines a mapping model that contains the mappings between UI models and their elements. It is not in the scope of this paper to detail or compare different mapping techniques, since most of them aim for UI generation and focus on UI design artifacts, different from our intended goals. But we consider them as a support for model mapping and traceability between the models.

2.3 BP Models and UI Models

BP models are associated with UI models through task models, seen in Figure 1 (it depicts the mappings between the models, not the content). Each level of the BP is mapped with the different levels in the task model, which is associated with UI components: *screen group*, a group of closely related screens and possible sub-groups to precisely classify screens; *screen*, a state of the user interaction where it is possible to perform a task or part of a task or even several tasks; *screen fragment*, a container of related elements in the screen; *screen element*, the most atomic component to perform user tasks (e.g. input or display data, navigate). UI components are linked to the final UI. The methodology is flexible enough to enable stakeholders (e.g. business analysts, UI designers) to specify at which level of granularity they will map the elements, depending on the complexity of the processes and systems, on the information available and other aspects that directly influence the impact analysis.

We have specified how these models are linked to support impact analysis when changes are requested in any of these models. The changes may happen in different directions: forward and backward approaches, in which the actions of stakeholders to change the models are organized in phases related to business process improvement methodologies (Process improvement) and User-Centered Design (User and Task Analysis, Conceptual Design, UI Design, Usability Test), as seen in Figure 1.

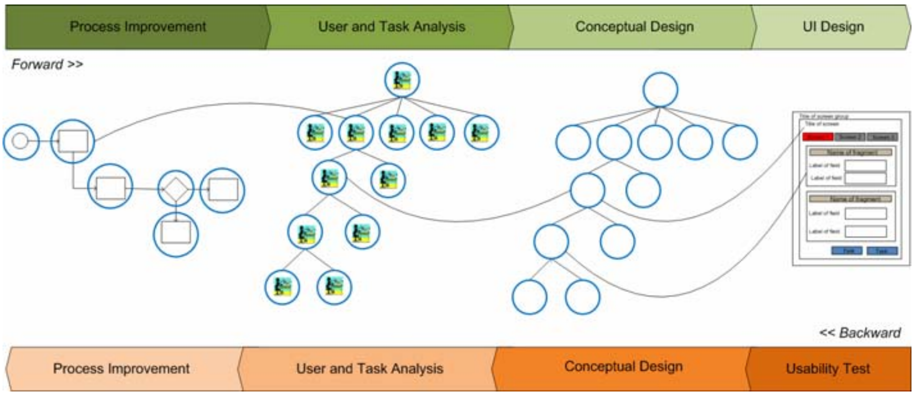

Fig. 1. Approaches used in the association of business processes with user interfaces

In the forward approach, changes on BPs may impact task models and UIs. It helps identifying what impact the optimization of processes has on the user interaction. Such changes can be done from a variety of reasons: new or alternative ways of doing things, new business opportunities, organizational changes, new regulations; etc. In the backward approach, changes on UIs may impact task models and BPs. It helps identifying how the improved usability of UIs suggested by systems users (e.g. customers, employees) impacts BPs. Reasons for such changes include defects to be fixed, better user understanding of the systems' features, new technology, etc.

We defined a set of rules to support the impact analysis by associating every BP model element with a UI element via a task model; this fine-grained granularity expresses details in the results of the impact analysis. With these rules, we can analyze: when a node is changed, what can be done in the other models to maintain alignment (e.g. when a screen is deleted, the related activity is no longer supported). The traceability is demonstrated by navigating in the chain of links, so when a node in a model is selected, the traceability lists what is related to it in the other models.

3 Managing Models

This framework is adaptable to changes in the organizational context, thus it accepts that the way processes are modeled and structured in different layers may change and that the philosophy of user interaction may be evolved, thus leading to the need to update how these core models are managed. For this purpose, we have selected an expert system approach based on production rules, where these rules specify how these models are managed to provide more flexibility. Rules support a rich representation of first-order logic to describe general knowledge about a certain domain [16]. The domain has a knowledge representation, whose elements are linked through formal symbols to compose rules as a collection of propositions [5]. There are many rule engines available on the market and open source as well. Most of them can be embedded in applications, allowing them to execute their own rules and get benefit from the decision-making features. In our case, when there are changes in the structure of the models that impact the form of managing them, the rules can be directly updated, without the need for maintenance of the traceability tool.

The basic properties of transformation rules are [12]: *tunability*, the possibility to adapt the transformation by parameters; *traceability*, the possibility to trace one element in the target model back to its causing element in the source model; i*ncremental consistency*, information added manually to the target model is not overwritten if the transformation is executed again (e.g. after the model is updated); *bidirectionality*, transformation applied from source to target and vice-versa.

As a contribution, we have defined 3 types of rules that adhere to these properties:

1. *Transformation*: transform BPMN [13] business processes in task models;
2. *Change Management*: do impact analysis of changes made on the models;
3. *Verification*: maintain consistency in the links of tasks with UI components.

These rules have been defined based on the links between the models, as explained in the previous section. The specific mapping between BP elements and task model elements have been presented and detailed in [18]. The mapping between BP and task model elements has been assessed through the association of: BPMN core elements (flow objects, connecting objects, artifacts) with task relationships (e.g. sequence flow with enabling and enabling with information passing); BP activity attributes with task properties (e.g. conditional flow with optional task); and process task type with task type (e.g. service task with application task). Such an assessment enabled the identification of similarities in the semantics of these two notations and the different aspects that may indicate a prospective need of adaptation in one them. For example, a Data-Based Exclusive Gateway represents that only one of the various paths can be taken and it can be represented as a deterministic choice between tasks in the task model, but for this representation to be complete, it is necessary to allocate a condition for each target task in the task model, not only a condition in the relationship between two tasks because each target activity in a gateway can have a different condition. In addition, such associations depict a many-to-many mapping between BPMN and task model elements as depicted in the association of both BPMN 'sub-processes' and 'tasks' to 'tasks' in the task model; and the association of BPMN 'sequence flow' to both 'enabling' and 'enabling with information passing' in the task model, depending on the condition type of the sequence flow; etc.

From these mappings, we have defined 53 transformation and change management rules; each of them has been explored for four different operations, totalizing in 212 rules for forward engineering. For this paper, we focus on the transformation and change management rules that have been written using the Drools [11] rule language.

To demonstrate how the alignment is achieved, the rules are explained using the context of customers requesting insurance contracts, from a case study with a large bank-insurance organization [18]. In this context, when aiming to increase customer satisfaction and optimize the performance of enterprise operations (with a more efficient service through faster responses), business analysts decided to allow customers to follow the status of their requests. This improvement has impacts spread in users' tasks and UIs of different systems in the forward and backward approaches, illustrated in the following sub-sections.

3.1 Forward Approach

In the forward approach, we demonstrate how changes on BPs have impacts on task models and UI components. Note that the IDs used in the rules are abstract illustrations of the elements (e.g. “sF-LogIn-ViewRequest”) to help in their understanding, they are not real representations of the IDs saved on the knowledge base (e.g. UUID = d79d24ec0c6d4ba5b08f9cf911512f78).

From Business to Task. For each element updated in a BP model, there is a rule that transforms it into an element in the task model. These rules are classified in operations that represent actions performed on BP elements that result on actions that must be executed on task models (Table 1). The intent is to consider different types of operations that can change BPs and execute the existing rules at design time to maintain the alignment, exemplified in this paper with one rule for each operation. The first rule is presented both in natural language and in the Drools rule language for demonstration; the other examples are presented only in natural language for clarity.

Table 1. Classification of operations

Operation	Definition	Action on BPMN element	Change on task model	Impact on UI components
Add	To include an element	Create element	Create TM element	Create UI component
Delete	To remove an element	Remove element	Remove TM element	Remove UI component
Modify	To change the attributes of an element	Change type of element Change name of element	Update TM element Change name of TM element	Update UI component Change name of UI component
Order	To arrange elements	Change sequence of activities	Change sequence of tasks	Change sequence of UI component
Merge	To unite elements	Remove element	Remove TM element	Remove UI component
		Change type of element	Update TM element	Update UI component
		Change name of element	Change name of TM element	Change name of UI component
Split	To divide or separate elements	Change type of element	Update TM element	Update UI component
		Change name of element	Change name of TM element	Change name of UI component
		Create element	Create TM element	Create UI component
Decompose	To separate into constituent elements	Create element	Create TM element	Create UI component

In the BPs for insurance contracts, in order to enable customers to follow the status of requests, business analysts have created a new sub-process called ‘Follow request status’. This sub-process is composed of tasks informing that customers can login into the *insurance contract online system* and then view the request status. For each of the newly created tasks and their associations in the sub-process, rules for the operation ‘Add’ are executed to create the equivalent elements in the task model. For the operation **‘Add’**, the following rule is executed, which expressed in natural language means that: “`When a default sequence flow (with a specific id) linking two tasks (e.g. source task ‘Log in’ and target task ‘View request’) is created in a BP, then it results in creating an enabling relationship linking the two equivalent tasks in the task model`”:

```
rule "AddBPMNSequenceFlow"
when
    s: SequenceFlow(id == "sF-LogIn-ViewRequest")
then
    insert(new Enabling(s.getId(),s.getSourceRef(),s.getTargetRef()));
end
```

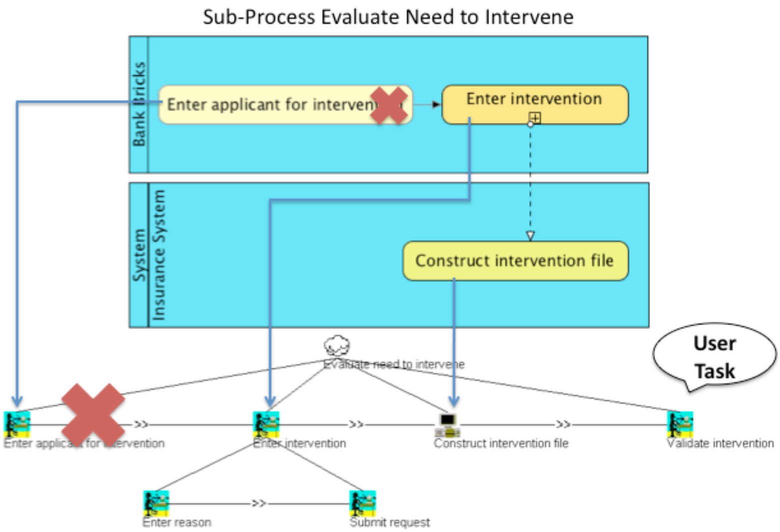

Fig. 2. Task deleted from the sub-process and its equivalent deleted task in the task model

Besides that, business analysts have also decided to improve the internal communication so the customer can acknowledge the status updates more efficiently. They have decided that the bank agents do not need to enter interventions through a different form; it is enough to enter the reasoning within the insurance contract request. The sub-processes "Evaluate need to intervene" and "Consider intervention by inspector" are directly impacted by this strategic decision.

First, the sub-process "Evaluate need to intervene" needs to have the task "Enter applicant intervention" removed since there is no longer a specific form for intervention requests (Fig. 2). To delete this task and its relationships in the task model, rules for the operation 'Delete' are executed. Note that this figure shows the task model evolved with a user task, not relevant for the business process, but important for user interaction. When the BP is changed and the task model is regenerated by executing the rules, this extra user task must remain to guarantee incremental consistency; reason for adopting an incremental transformation approach [12], which updates the changed elements with the newly generated ones, keeping the extra elements and the unchanged elements untouched.

For the operation **'Delete'**, the following rule is executed to delete a task and its relationship:

```
rule "DeleteBPMNTask"
when
     a task (i.e. 'Enter applicant for intervention') and its se-
     quence flow are removed from BP
then
     remove the equivalent task (with the same id of task) and ena-
     bling relationship (with the same id of sequence flow) from the
     task model
end
```

Second, the sub-process "Consider intervention by inspector" has its tasks updated to allow inspectors to change a larger set of applicants' data, not just the value for insurance. This way, they can also change the type of insurance (product) and the charged taxes depending on the applicant's income (Fig. 3). For the operation **'Modify'**, the following rule is executed:

```
rule "ModifyBPMNTaskIntoSubProcess"
when
     a task (i.e. task 'Change applicant data') is modified into a
     sub-process in a BP
then
     change this atomic task in the task model into a task composed
     of the sub-tasks from the equivalent task in the BP
end
```

Still in the sub-process "Consider intervention by inspector", it is updated to allow inspectors to view the applicant name, data file, insurance file and date of request for every new intervention that arrives without the need to previously search for it (Fig. 3). To change the sequence of elements, rules for the operation **'Order'** are executed, such as the execution of the following rule:

```
rule "OrderBPMNSequenceFlow"
when
    the order of two tasks (i.e. 'Search for applicant' and 'View
    applicant data') is changed in a BP
then
     change the order of the equivalent tasks in the task model
end
```

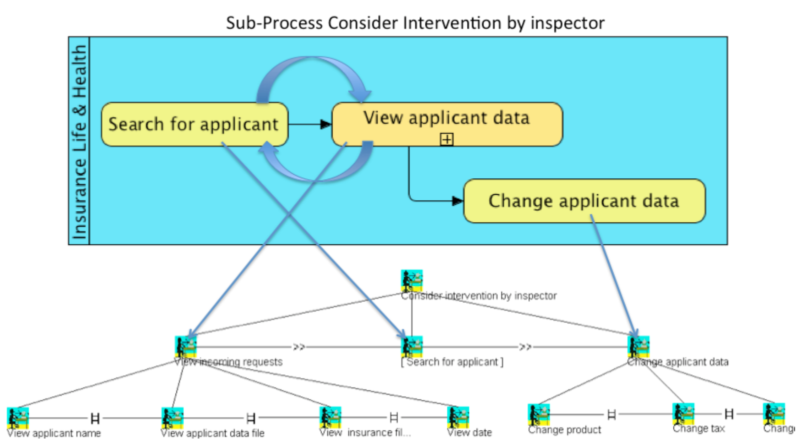

Fig. 3. Tasks updated in the sub-process and its equivalent updated tasks in the task model

Concerning the other operations, we consider that operations to merge, split and decompose elements are resulting from a collection of the previously mentioned operations. For instance, to ***merge*** two sub-processes in a BP, one of the sub-processes is deleted (e.g. call Rule *DeleteBPMNSubProcess*) and the other one is modified, such as changing its name (e.g. call Rule *ModifyBPMNSubProcessName*). An example to ***split*** one sub-process in two sub-processes, the existing sub-process is modified (e.g. call Rule *ModifyBPMNSubProcessName*), a new one is created (e.g. call Rule *AddBPMNSubProcess*) and a link between them is created (e.g. call Rule *AddBPMNSequenceFlow*). To ***decompose*** one sub-process in tasks, for each new task created within the sub-process, a new sub-task is created within the equivalent task in the task model (e.g. call Rule *ModifyBPMNSubProcessWithSubTasks*). When any of these operations are performed on BP elements, the consequent actions are performed on the task model elements and UI components that have been linked to the updated BP elements. After the rules are executed and task models are aligned with the updated business processes, a new set of rules are executed in order to maintain alignment with UIs, explained as follows.

From Task to UI. Since the task model is mapped to UI components; for each element changed in a task model, there is a rule that indicates which UI components should be updated accordingly. These rules are classified in operations (Table 1) that represent actions performed on task model elements that result on advices, which could be manually done on UIs by UI designers/usability experts.

To exemplify that, still in the intervention of insurance contracts (as a continuation of the rules related to Fig. 3), inspectors view all incoming requests then decide to either search for a specific applicant or directly change the data of an applicant among the incoming requests. In this scenario, the order of tasks has been changed and the type of one task has been updated to a sub-process (Fig. 4). First, for the operation 'Order', the following rule is executed:

```
rule "OrderTMEnabling"
when
     the order of two tasks (i.e. 'Viewing incoming requests' and
     'Search for applicant') is changed in the task model
then
     change the order of the UI components that are linked to these
     tasks (i.e. two screen fragments 'View incoming requests' and
     'Search for applicant')
end
```

Second, for the operation **'Modify'**, the following rule is executed:

```
rule "ModifyTMAtomicUITaskIntoSubTasks"
when
     an atomic task (i.e. 'Change of applicant data') is modified
     into a task composed of sub-tasks in the task model
then
     add screen elements for each of the added sub-tasks (e.g.
     'Change product', 'Change tax') in the screen fragment that is
     linked to this updated task (i.e. screen fragment 'Financial
     data' of the screen 'Change data')
end
```

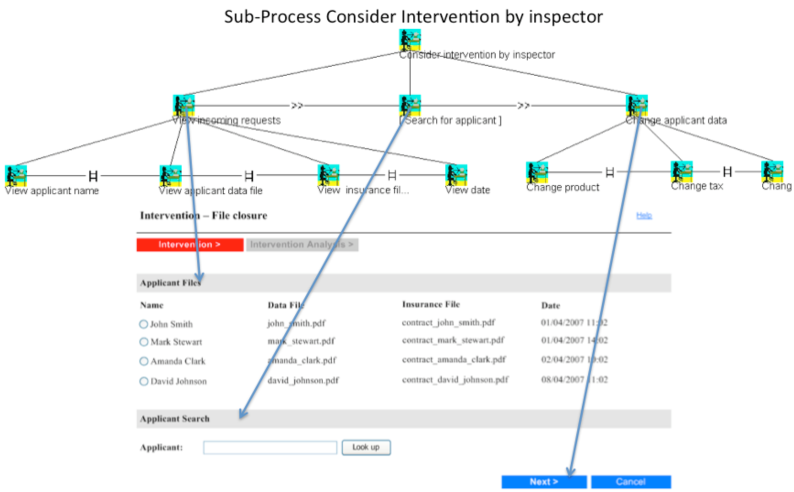

Fig. 4. Tasks updated in the task model and its equivalent updated UI

Another example for the operation 'Modify' is when two sub-tasks are first linked with 'Concurrency' relationship, then this operator is changed to an 'Enabling' relationship in the task model; the consequent change in the UI is that the updated UI should reflect the sequence by organizing the related UI components in two screens or in two screen fragments that are navigable as a wizard.

To summarize with a general overview of changes, the process 'Closure' was updated to provide transparent information for customers and more productivity with internal communication. This change has resulted in updating four sub-processes for a core product in the bank (Fig. 5). Changes on these sub-processes were processed in the task models that were adequately evolved to consider the user interaction. The updated task models were then processed in order to indicate exactly which UIs components (linked to users' tasks) were impacted from such changes. Before changes are made, the traceability enables impact analysis by allowing stakeholders to navigate in the chain of links and visualize which UIs could be impacted if any change is made on specific BPs as suggested by business analysts.

The impact analysis resulted in suggesting the following changes: 1) The screen 'File Closure' has now a new screen fragment 'intervention' composed of two screen elements to allow bank agents to inform the reason and to do the request in the same screen, therefore; 2) There is no need for the screen 'Enter Intervention' with a specific form that added burocracy; 3) In the screen 'Consider Intervention', its screen fragments were updated by changing their order; 4) The screen 'Change data' has more screen elements in screen fragment 'applicant data' that gave more flexibility to inspectors; and 5) The screen 'Follow Request' was created in the insurance contract online system to allow customers to follow the status of their requests. Now, we examine when changes start on UIs through the backward approach.

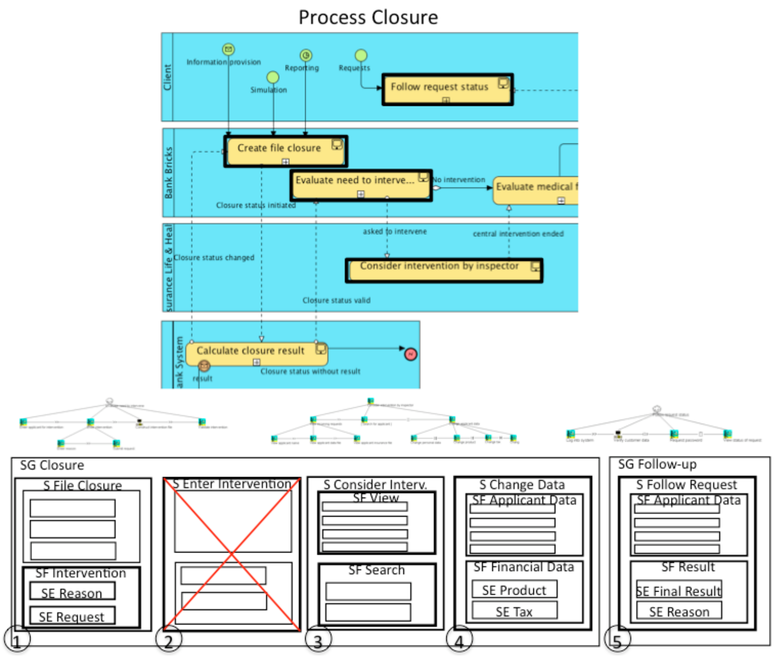

Fig. 5. Overview of the impact analysis considering BPs, task models and UI components

3.2 Backward Approach

The UI-Business alignment framework also adheres to the property of *bidirectionality*, which means that the transformation can be applied not only from source to target but also backwards from target to source [12]. The backward approach is applied when users of enterprise systems suggest improvements on UIs or point out issues that interfere with the progress of their work. This opens a new channel for business process improvement that can start with actual users (bank employees using the system at the bank agency or customers using electronic system on the web at home) who aid to increase productivity within the organization.

To exemplify the context of the backward approach, we present a set of change scenarios: after some weeks using the updated systems, some requests arrived from users at the bank agencies. For the screen 'File Closure', agents requested the screen fragment 'Intervention' to be retracted by default so they can expand it just when necessary. For the screen 'Consider Intervention', inspectors requested to add the screen fragment 'search' also on the top of the screen because there are usually too many incoming requests on this screen. For the insurance contract online system, customers asked to be able to request a second review and send extra documentation (e.g. work contract, bank statement, salary statement) to prove specific financial data in order to improve the type of insurance they want to receive.

To address this request, a new screen was created, named 'Request Second Opinion' and the task model was updated. With the operation **'Add'**, the following rule is executed:

```
rule "AddUIScreen"
when
   a new screen (i.e. Screen 'Request Second Opinion') is created
   with screen elements (i.e. 'Send-Work-Contract', 'Send-Bank-
   Statement', 'Send-Salary-Statement')
then
    create a task for the new screen and a set of sub-tasks for each
    of the created screen elements within the related screen
end
```

Once the task model has been updated, the next step is to update the BPs accordingly. For each added task in the task model, rules were also executed; such as for the operation **'Add'**, the appropriate rules (e.g. Rule *AddBPMNTask*) added the equivalent activities in the related BPs. There are though some situations in which one task model element is mapped to several BP elements because of the many-to-many nature of the mappings. For instance, if two sub-tasks are first linked with 'Concurrency' relationship, then this operator is changed to an 'Enabling' relationship in the task model; the consequent change in the BP can be: sequence flow, default sequence flow, message flow, association and directional association. One of them can be selected as default to execute the transformation, and business analysts may review the resulting BP to make any necessary updates.

4 Related Works

Recent works propose different types of technological support for traceability between business processes and the supporting software systems, such as [2]. Among them, the Center of Organizational Engineering (CEO) Framework represents an enterprise architecture that aligns the organization's strategy with its business processes, people and IT software and hardware [17], with a differential on adding people to this strategy.

Artifact-Centered Operational Modeling (ACOM) is an approach to capture business operational knowledge that is distinguished from other approaches by identifying one information structure, the artifact that travels from end to end in a process. It can be understood as an alternative to more familiar approaches, such as process modeling, differentiated by managing progress toward goals [4].

The Model Information Flow used at SAP [3] follows a model-driven approach that uses models and transformations between them for defining, deploying and managing SAP applications. It is noted that the models in the chain, after the business process model is customized for a project, are mostly related to IT infrastructure, for instance models to describe internal structure of the software, demand for resources by components, resource infrastructure configurations, etc.

One of the researches on the enhancement of the UI design practice with business process modeling [19] points out the importance of model-based UI design automation in scenarios with intensive business process that give rise to systems with lots of data entering and display, and role-specific UIs.

The researched strategies that align business processes and systems focus on information system architecture, considering information system and technological services, IT infrastructure, application platform and technological/software components. Even though there are recent works on the alignment of business processes with user-centered design and development, they tend to focus on specific technologies, such as web-based applications [15], even though they advocate for a hybrid approach combining task models and process models; while others define structures for associating business processes and UIs with direct links, such as [20]. On the other hand, we envision a more flexible approach with the use of conceptual models to facilitate interoperability of solutions; and considering users' tasks that are primordial for designing UIs that are easy to use and contribute to increase the productivity of professionals in organizations when doing their daily work constantly interacting with enterprise systems.

5 Conclusion

This paper presented a model-driven approach to link BP and UI models. With this approach, models are mapped in order to more efficiently propagate changes when needed. In addition, the user perspective is considered in alignment with business needs. In more details, we demonstrated how our proposal is suitable for the context of a large bank/insurance organization. This research has proposed a framework for UI-Business Alignment that is practical, user-centered, and adaptable to specific organizational structures, human-centered and provides model traceability.

As future work, in order to assess the application of this framework in a different business domain, we are applying this framework in a telecommunications large company in which we will be able to evaluate the integration of the UI-Business Alignment strategy with their software development process, the execution of the rules in a project related to one of the products with good visibility in the organization, and the acceptance of the stakeholders involved with the solution.

Acknowledgments. We gratefully acknowledge the support of the Program Alban, the European Union Program of High Level Scholarships for Latin America, under scholarship number E06D103843BR.

References

1. Aberdeen Group. Aligning IT to Business Process: How BPM is Complementing ERP and Custom Applications. Research Brief. Abardeen Group (2007)
2. Ali, M.F., Pérez-Quiñones, M.A., Abrams, M.: Building Multi-Platform User Interfaces with UIML. In: Multiple User Interfaces: Engineering and Application Framework. John Wiley and Sons, New York (2003)
3. Belrose, G., Brand, K., Edwards, N., Graupner, S.: Business-driven IT for SAP – The Model Information Flow. In: Proceedings of the Second IEEE/IFIP International Workshop on Business-Driven IT Management (2007)
4. Bhattacharya, K., Caswell, N., Kumaran, S., Nigam, A., Wu, F.: Artifact-centric Operational Modeling: Lessons learned from customer engagements. IBM Systems Journal 46, 4 (2007)
5. Brachman, R.J., Levesque, H.J.: Knowledge Representation and Reasoning. Elsevier, Amsterdam (2004)
6. Brown, J., Graham, T.C.N., Wright, T.: The Vista Environment for the Coevolutionary Design of User Interfaces. In: CHI 1998, pp. 376–383 (1998)

7. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers 15(3), 289–308 (2003)
8. Davenport, T.: Process Innovation: Reengineering work through information technology. Harvard Business School Press, Boston (1993)
9. Henderson, J.C., Venkatraman, N.: Strategic Alignment: Leveraging information technology for transforming organizations. IBM Systems Journal 38(2&3) (1999)
10. Henry, P.: Process-User Interface Alignment: New Value From a New Level of Alignment. Align Journal (2007)
11. JBoss Drools, `http://www.jboss.org/drools`
12. Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Professional, Reading (2003)
13. OMG Business Process Modeling Notation Specification v1.1, `http://www.omg.org/spec/BPMN/1.1/PDF`
14. Paternò, F., Santoro, C.: A unified method for designing interactive systems adaptable to mobile and stationary platforms. Interacting with Computers 15(3), 349–366 (2003)
15. Pontico, F., Farenc, C., Winckler, M.: Model-Based support for specifying eService eGovernment Applications. In: Coninx, K., Luyten, K., Schneider, K.A. (eds.) TAMODIA 2006. LNCS, vol. 4385, pp. 54–67. Springer, Heidelberg (2007)
16. Russel, S., Norvig, P.: Artificial Intelligence – A Modern Approach. Prentice-Hall, Englewood Cliffs (2003)
17. Sousa, P., Caetano, A., Vasconcelos, A., Pereira, C., Tribolet, J.: Enterprise Architecture Modelling with the Unified Modelling Language 2.0. In: Ritten, P. (ed.) Enterprise Modelling and Computing with UML. IRM Press, Hershey (2006)
18. Sousa, K., Mendonça, H., Vanderdonckt, J.: A Model-Driven Approach to Align Business Processes with User Interfaces. International Journal of Universal Computer Science, Special issue on Human-Computer Interaction14(19), 3236–3249 (2008); Gallud, J.A., Lozano, M. (eds.)
19. Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S., Stolze, M.: User-centered Design and Business Process Modeling: Cross Road in Rapid Prototyping Tools. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4662, pp. 165–178. Springer, Heidelberg (2007)
20. Traetteberg, H.: UI Design without a Task Modeling Language – Using BPMN and Diamodl for Task Modeling and Dialog Design. In: Forbrig, P., Paternò, F. (eds.) HCSE/TAMODIA 2008. LNCS, vol. 5247, pp. 110–117. Springer, Heidelberg (2008)
21. Vanderdonckt, J.: A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 16–31. Springer, Heidelberg (2005)