Journal of Universal Computer Science, vol. 14, no. 19 (2008), 3236-3249
submitted: 23/7/08, accepted: 29/10/08, appeared: 1/11/08 © J.UCS

A Model-Driven Approach to Align Business Processes with User Interfaces

Kenia Sousa, Hildeberto Mendonça, Jean Vanderdonckt
(Université catholique de Louvain, Belgium
kenia.sousa@uclouvain.be, hildeberto.mendonca@uclouvain.be
jean.vanderdonckt@uclouvain.be)

Abstract: Information Technology (IT) has evolved over time from its traditional use as administrative support towards a more strategic role to enforce business processes (BP). But several organizations that adopt BP modeling as a source to implement enterprise systems struggle to maintain such a link. To solve this problem, most researches are focused on software engineering to specify the association between models from business and IT to support propagating changes. But even though there are several techniques to control the alignment of BP and their systems, there lacks a solution that addresses a major aspect of systems: their User Interfaces (UI). The negative impact of focusing only on functional aspects is that many changes on business processes that affect user interfaces are not carefully considered. Our solution proposes a method for the alignment of business processes with user interfaces of systems by adopting a model-driven approach. Such support is targeted at large organizations in order to enable them to be more capable of managing those links. The method along with the tool guarantees that all the models used to develop enterprise systems are internally mapped and that any attempt to make changes in at least one of them is alerted with warnings about the possible impacts. We propose a practical method, adaptable to specic organizational structures, that enables professionals to focus on achieving organizational goals, and still puts forward users' need for a richer user interaction.

Key Words: business process modeling, model-driven engineering, user-centered design

Category: H.1, H.5.2

1 Introduction

Most large organizations use Business Processes (BP) to represent the way they work in order to teach their workers to repeat certain steps every time they need to do certain things, but also to preserve their own corporate knowledge. Davenport [Davenport, 1993] defines a BP as a structured set of activities ordered across time and space to produce a specific result for a particular customer or market. Similarly, the use of Information Technology (IT) has evolved over time from its traditional use as administrative support towards a more strategic role to enforce BP and thus, produce added value for its customers. In addition, there is a growing interest on the alignment of IT with business objectives from different perspectives, such as from business executives, IT managers, and academics.

It is observed in the IT domain that there are several recent researches focused on specifying the association between models from business and IT to support propagating changes [Aversano et al., 2005]. Another observation is that many of these researches use software engineering models to address alignment issues, such as in [Vasconcelos et al., 2001]. However, such strategies lack the consideration of a major aspect of information systems: their User Interfaces (UI). The impact of focusing more on functional aspects is that many changes on business processes that affect UIs are not carefully treated, thus, leaving the decision of how changes impact UIs to be done in an ad-hoc manner. This effect is even more negative when we address large and complex systems in which changes in business process rules are common and may have impact on even hundreds of UIs, thus, leading to the need to define strategies to maintain the traceability between business process and UIs whenever changes are requested.

Certain changes on business processes may have different kinds of impact on UIs, such as ordering of components, ordering of screens, positioning of components on screens, and navigation aspects. But these aspects are only expressed in UI models, not present, for instance in use case, class, and activity diagrams of UML. Aiming at supporting the development of UIs and their alignment with business processes, the overall research goal is to correlate business processes and UIs by 1) defining the association of business process with UI models, and 2) presenting a tool for model transformation that addresses traceability. This proposal is mainly aimed towards organizations that are driven by their business processes, and want their systems to address such processes in a way that as the processes are created, maintained, and evolved, so are its supporting systems.

This paper is structured as follows: section 2 presents related works; section 3 introduces transformations of business processes into task model elements; section 4 contextualizes the organizational scenario; section 5 outlines the traceability solution with method and tool; section 6 closes the gap with the development of UIs; and the last section concludes with results and ideas on future work.

2 Related Works

There are recent works on the domain of model-driven architecture that discuss about traceability in model-driven development and its importance to analyze alignment with requirements, impact and propagation of changes, etc. However, the most common use of traceability in model transformations is between data models and class diagrams, such as in [Vanhooff et al., 2007], which discovers useful trace information from model relations. A work on traceability in model-driven development of business applications [Rummler et al., 2007] links artifacts from the development process that range from software requirements, test cases, design objects and code fragments.

Even though many contributions support the alignment of business processes and Information Systems (IS), it still lacks concerns on the UI. Results from investigations [Devaraj & Kohli, 2003] have demonstrated that for IT to positively affect the organization, IS must be appropriately used, which is directly influenced by the UI [Sujitparapitaya et al., 2001]. This corroborates with the idea that user interaction has major influences on the outcome that IT can bring to organizations. But little attention has been devoted to the user interaction of developed systems.

There are few works that consider integration with UI, such as some works done in IBM research centers [Sukaviriya et al., 2007], in which the focus is on designing low-fidelity prototypes based on business process models. Another example is a work [Stolze et al., 2007] that argues that only relying on workflow models may be problematic to represent aspects of the user interaction because it is difficult to consider specific user requirements in such models. Aligned with the idea that information from business processes are not enough for UI design, the work of [Pontico et al., 2006] advocates for a hybrid approach that combines task models and process models.

Many approaches are either more technical or more managerial, but they all lack the concern with the user perspective. Recent approaches that integrate business perspective with UI design are more concerned with specific solutions and devices, for instance using rapid prototyping. There is also a clear absence of generic structures for business and UI models, which inputs constraints in proposals with limited applicability. With a pragmatic point of view, our approach is envisioned as more flexible because of the use of conceptual models with adaptable structures to facilitate communication and knowledge sharing between departments and interoperability of solutions when facing change requests.

3 Aligning Business Process and User Interface Models

In order to associate business processes with UI models, we have made a detailed analysis of the specification on Business Process Modeling Notation (BPMN) [OMG, 2008] and listed some business process elements to be transformed into task model elements. This list was initially based on a first version of this research [Sousa et al., 2008] and it has evolved considering BPMN version 1.1.

Table 1 specifies the association of the relationships in business processes and in task models, with the rationale behind the decisions explained as follows. (1) Sequence Flow and Enabling operator represent the order in which activities are performed. (2) In cases when an activity passes information for the execution of the next activity, a Data Object can be associated to a Sequence Flow. (3) Considering two activities A1 and A2, the use of a Conditional Intermediate

Event in A1 can trigger a rule that momentarily pauses the execution of A1 while A2 is executed, and a Link Intermediate Event in A2 can connect back to the moment where A1 was paused. This is in accordance with the Suspend/Resume operator that momentarily interrupts one activity as another one is executed. (4) Both the exclusive gateway and the deterministic choice represent a point in the process where the first alternative that is chosen determines the flow that will be taken. (5) Many combinations of the alternatives can be selected with Inclusive gateway, but Exclusive gateway expresses that only one of the activities is fully accomplished and its result is passed through the flow, aligned with non-deterministic choice. (6) When two activities are competing, and one is being performed, it may be interrupted by the second one, but the interrupted one is canceled by triggering the Cancel Event as soon as the second one starts, similar to Disabling. (7) Ad-Hoc marker in a sub-process means that its activities can be performed in any order, which addresses the same function as an Order Independence operator. (8) In cases when two or more activities may be executed in parallel, the Parallel Gateway and Concurrency serve the same purpose. (9) When activities are performed in parallel, similar to the previous situation, but they also synchronize data, Data Objects and Information Passing are added to the flow. (10) Message Flow shows the flow of messages between two entities that can be presented by the Enabling With Information Passing operator. (11) A Start Event can represent a source task and the activity that the event will start can be the target task. The event detail can be stored in the temporal precondition, which can hold information such as a message, time, condition, signal, a list of possibilities, etc. (12) An Intermediate Event attached to the boundary of an activity that is linked to a second activity is represented by three tasks in which the first task is an activity, the middle task is the event and the last task in the second activity. A temporal precondition between the first two tasks can store the content of the event (e.g. the message that is received). Other structures for intermediate events are not addressed here. (13) The activity linked to the End Event can be the source and the end event can represent a target task. The event detail can be stored in the temporal precondition that will be triggered when the activity with the event is ended. End event types Cancel, Compensation and Terminate have to be handled differently, but they are not addressed here for space reasons.

Table 2 associates activity attributes from business processes with task properties from task models. (1) The conditional flow can have expressions that determine whether or not the task in this flow will be used, which can express that a task is optional, depending on a certain condition. (2) Standard LoopType determines that an activity is performed repeated times, similarly to the Iteration. In addition it has an expression (its attribute LoopCondition) that is evaluated as a condition for each cycle of the loop. (3) Multi-instance loop has a numeric

Table 1: Association of business and task elements

No.	Business Process	Task Model
1	Sequence Flow	Enabling
2	Sequence Flow + Data Object	Enabling + Info Passing
3	Conditional + Link Intermediate Event	Suspend/resume
4	Exclusive gateway	Deterministic choice
5	Inclusive gateway + Exclusive gateway	Non-Deterministic choice
6	Cancel Intermediate Event	Disabling
7	Ad-Hoc marker in sub-process	Order Independence
8	Parallel gateway	Concurrency
9	Parallel gateway + Data Object	Concurrency + Info Passing
10	Message Flow	Enabling + Info Passing
11	Start Event	Enabling+temporal precondition
12	Intermediate Event	Enabling+temporal precondition
13	End Event	Enabling+temporal precondition

Table 2: Association of activity attributes and task properties

No.	Activity Attribute	Task Property
1	Conditional Flow	Optional
2	Standard Loop + LoopCondition	Iteration
3	Multi-Instance Loop + MI Condition	Finite Iteration
4	Standard Loop + Loop Maximum	Finite Iteration

expression (its attribute MI Condition) that is evaluated once before the activity is performed and the result of this evaluation specifies the number of times that the activity will be repeated. This is in alignment with Finite Iteration, in which a task can be iterated n times. (4) Standard Loop has the attribute Loop Maximum that informs the limit to the number of loops, in alignment with Finite Iteration, in which a task can be iterated n times.

Table 3 associates task types of business process and of task models. Task types provide detailed information that helps in designing UIs. In business processes, tasks are the most atomic activities. In task models, tasks are decomposed in sub-tasks until reaching the most atomic level. (1) User and Interaction types

Table 3: Association of process activities and task types

No.	Process Task Type	Task Type
1	User	Interaction
2	Service	Application
3	Manual	Single User
4	None	Abstract
5	User + Attribute Performers	Multiple Users
6	Receive	Application
7	Send	Application
8	Script	Application

represent tasks performed by a human with the assistance of a system. (2) Service and application represent an automated task performed by the system. (3) Manual and Single User are related to tasks that are performed by a human without any aid from a system. (4) Abstract is used to group tasks, it has no specific meaning, which can mean that the type None is suitable for this case. (5) User type can also be appropriate to represent multiple users when the attribute Performers specifies if it is performed by a group or an organization. (6) Receive type is similar to an automated task that waits for an incoming message pattern (web service) from an external participant. (7) Send type is like an automated task that sends an outgoing message pattern (web service) to an external participant. (8) Script type is a script defined in a language that a business process engine can interpret and execute, similar to an automated task.

This proposed approach closes the gap in UID lifecycle for enterprise systems, which most commonly starts with task models, leaving aside business processes. The transformation of business process models into task models represents a first version of task models that needs refinement. Such refinement can be done by human factors experts and UI designers, who update tasks and their relationships with new aspects of user interaction. The task model refinement can certainly be done with the participation of business analysts to allow consistency with business processes.

4 Organizational Context

We have analyzed the context of a large organization in order to propose solutions for their issues concerning the alignment of business processes with UIs, as we have presented in [Sousa et al., 2007]. This organization is decomposed in bank and insurance sub-divisions. The bank has a business department composed of around fifty business analysts, responsible for analyzing and evolving the

business processes that drive the functioning of the bank and insurance. The insurance sub-division designs and develops systems that support the business processes.

Their methodology specifies business processes with flows of sub-processes and activities until an atomic level in which tasks are associated to business rules and data. In their business process models, for each process there are 60 elements ranging from sub-processes, activities, and tasks, not even mentioning their business rules, which would increase this number by nearly 100 more elements. We have also considered that some organizations do not detail their business processes to such a level, instead, they maintain business process models in a high level description and detail them when necessary for system development using software engineering artifacts (e.g. use cases).

Since we can come across organizations that follow divergent methodologies, our proposal is also applicable in the example of high level business process description. In such cases, task models are also created from business processes, but the refinement is more fine-tuned. This scenario is not addressed in this paper, which focuses on the context of the studied organization with a great set of business process models specified into details.

We have conducted interviews with three business analysts, two system analysts and developers, and two UI designers. During the meetings we worked with examples of business processes and screens for insurance contracts. We have learned that: The business analyst team has a business process modeling methodology decomposed in six layers. They expect the other departments to follow a UI design method that maintains alignment with the business processes. Since that is not currently the case, they prepared different spreadsheets in the attempt to align business process layers with UIs elements, which can start with screens until basic objects (e.g. fields). The system analysts and UI designers, on the other hand, find it very complicated to handle these different documents, which they argue requires a deep understanding of the company products. They also need extra time to keep the spreadsheets updated and consistent, which is not their reality since they are under-staffed to address various demands, especially for UI design and usability evaluations.

The main detected issues are: lack of correlation between business process and UI design; difficulties in doing impact analysis after changes; and difficulties to find, to understand and to keep updated information spread in several artifacts. To address these issues, we propose an approach to align business processes with UI design, trying to maintain both an organizational point of view, to be compliant with decisions from top-management; and an end-user perspective, to develop systems that are usable. In the next section, we, then, present how these issues are addressed through method specification and tool support.

5 Model-Driven Approach from Business Process

Using the strategy proposed in section 3 to transform business process models into task models; their association is automatically created as the transformations are executed. There is also the scenario in which models are manually created, thus, making it necessary to make manual mappings between them. Our proposal is supported by UsiXML [Limbourg & Vanderdonckt, 2004] models and founded on the Cameleon Reference Framework [Calvary et al., 2003], which is composed of four development steps: create conceptual models (e.g. task model, data model, user model), create Abstract UI (AUI), create Concrete UI (CUI), and create Final UI (FUI).

We have defined a method for traceability between BP and UIs (Figure 1) applying a method engineering strategy [Sousa et al., 2007] that considers organizational goals, thus, making it more feasible to be applied it in the industry. The method activities are: 1) Business analysts design business process models in a business modeling tool; 2) Business analysts export the XML from a BP tool and import it into a traceability tool; 3) System analysts create the domain model; 4) Business analysts request to generate task model based on the BP XML; 5) UI designers and human factors experts refine the generated task models by updating tasks and relationships; 6) UI designers list screen components based on task models (screen group, screen, screen fragment, and screen element); 7) UI designers create CUI models from task models; 8) Programmers develop FUIs from CUI models.

This method can be applied in association with software development process techniques according to what the organization applies in its projects. In a scenario in which the organization adopts the Rational Unified Process (RUP) [Kruchten, 2000], the core disciplines are performed according to the organizations' customization and the traceability method activities 1 and 2 are related to the Business Modeling discipline, activities 3 to 7 are related to the Analysis & Design discipline, and activity 8 is related to the Implementation discipline.

In an agile context [Ambler, 2002] that adopts the Agile Model Driven Development, the core activities are performed according to the organizations' customization and the traceability method activities 1 and 2 generate outputs that serve as input to the Initial Requirements Envisioning, activities 3 to 7 are related to the Iteration Modeling, and activity 8 is related to the Test Driven Development.

Following CMMi for Development [University, 2006], the core activities are performed according to the organizations' customization and the traceability method activities 1 and 2 are related to the Process Management process areas, such as Organizational Process Definition; activities 3 to 8 are related to the Engineering process areas, such as Technical Solution. Such associations between our method and standard software engineering approaches make professionals

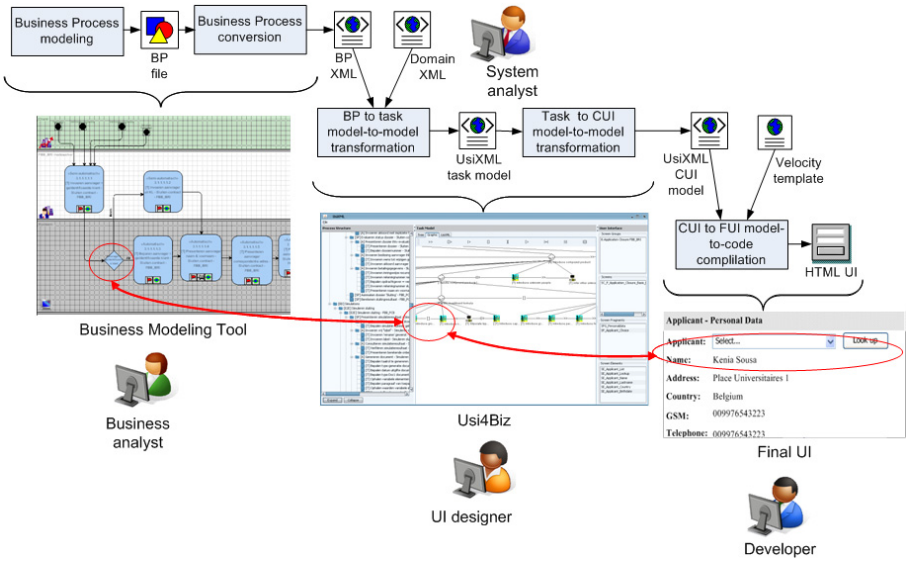

Figure 1: Outline of the method with roles, artifacts and tools

aware of the inclusion of the traceability method within their projects.

To apply this method, a tool called Usi4Biz (User Interface for Business) is being developed to manage models using XML, which enables communication with other tools, such as modeling business processes in commercial tools that provide XML schemas that can be exported. Usi4Biz is aimed at identifying the impact of changes whenever the business processes are updated, this tool can provide information of which other models are impacted.

Both the method and the tool provide resources for the software process being applied in the organization. For the RUP context, for instance, the business process created in the Business Modeling discipline using any business modeling tool (1) is input for Usi4Biz (2), which in turn gives as output the task model (4), that serves as input for the Analysis & Design discipline during domain analysis using any domain modeling tool (3), for task analysis using any task modeling tool (5), for listing screen components using Usi4Biz (6) and for UI design using any UI design tool (7). The result is a set of designed UIs that are used as input for the Implementation discipline during UI development (8). After mapping business processes and user interfaces using Usi4Biz, which demands user interaction, our solution helps the user to keep the consistence of these mappings, which is made automatically by Usi4Biz.

The association of business process, task model and UIs is depicted in Figure 2. On the left, there is the business process model decomposed in six layers,

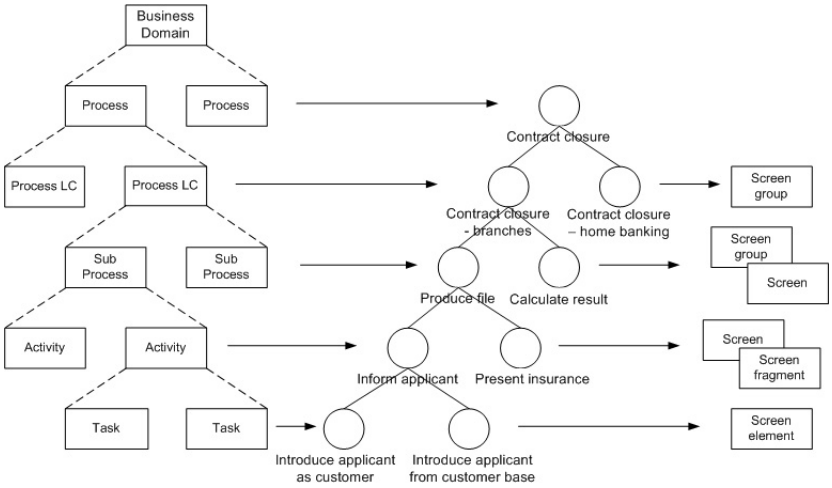

Figure 2: Association of business processes with user interfaces

as defined by the organization under study. In our proposal, these levels of the business process are associated with the hierarchical levels of a task model, on the middle of the figure. The association starts with the process layer because the business domain represents the overview of the process architecture. On the right of the figure, the second level of the task model is associated with screens components numbered in Figure 3: *screen group*, a group of closely related screens; *screen*, a state of the user interface when executing a task or part of a task; *screen fragment*, a container of related elements; and *screen element*, the most atomic component. This association starts in the second level of task models because the first level is often an abstract task useful for grouping, not representative for screen organization. Both the second and third levels of task models can be represented through screen groups. In cases when these two levels are screen groups, the fourth level is represented by a screen, not a fragment. However, the last level of task models is always represented by screen elements; therefore, screen fragments are optional when there are many levels of screen groups. These are examples of verification rules, useful to verify when system analysts associate tasks with screens manually.

Figure 3 depicts Usi4Biz, in which we present the structure of the tool, not the contents of the models. On the left of the tool there is the BP extracted from the business process XML of a commercial tool, shown in a tree structure. On the middle, there is the task model generated from this business process. On the right, there are the UI components. This tool is being developed in JavaTM, using a) the Swing [Microsystems, 2008c] library to implement the graphical UI thanks to its standard look and feel presentation in different platforms; b) StAX

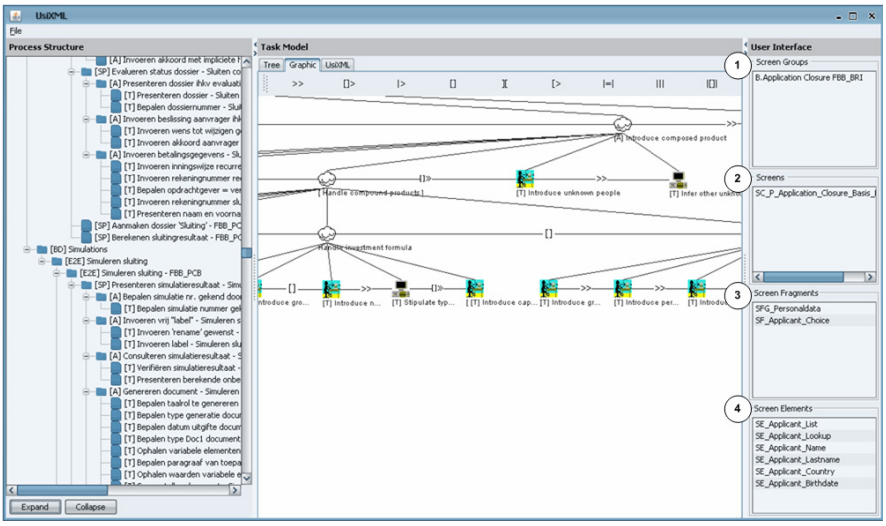

Figure 3: Usi4Biz with business process, task model and UI components

(Streaming API for XML) [Microsystems, 2008b] to read and write XML content because of its high performance to manipulate a huge amount of XML data, such as external models and UsiXML models; c) the OWLAPI [of Manchester, 2008] to process OWL ontologies; d) SWRL Bridge [Protégé, 2008] to transform SWRL rules in OWL language to a specific rule language supported by a rule engine; e) the Drools [JBoss, 2008] rule engine to give the necessary flexibility to manage models; f) the Apache Derby [Foundation, 2008a] embedded database to store local and temporary data, simplifying the installation and distribution process; g) the Netbeans Visual Library [NetBeans, 2008] to render visual diagrams, such as task models and navigation models; and h) Jersey [Microsystems, 2008a], the REST web services library to synchronize local data with a centralized repository, accessible by other people in the organization.

6 Closing the Gap with UI Design and Development

Most IT organizations that already adopt specific techniques for designing and developing their system UIs can also profit from adopting this method and tool. To contribute with UI design of enterprise applications, it is important to detail how the work of UI designers can maintain alignment with BP and still be useful for programmers. Besides performing the activities of the method, we depict a scenario with template-based design and development of UIs and demonstrate how it can be linked to Usi4Biz.

1. UI designers list screen components based on task models using Usi4Biz:
   1.1. Analyze the complexity of task models and decide what parts of them are represented as screen components;
   1.2. Associate tasks with the listed screen components;
2. UI designers create CUIs:
   2.1. Design templates using the defined screen components;
   2.2. Associate templates with screen components using Usi4Biz;
3. Programmers develop FUIs:
   3.1. Use the UI templates as input to develop code for the behavior of CUIs;
   3.2. Associate templates with the coded UIs (e.g. Java classes) using Usi4Biz.

Considering a development technique, UI designers can connect CUIs components with UI templates to generate FUIs productivily. Velocity is an open source template engine for template-based transformations [Foundation, 2008b]. Velocity templates are commonly used to generate code that enables language portability. With the same source model (e.g. data model) and a template engine, just changing the template, it is possible to generate code in different languages. Using Velocity to design UIs, we use the following models: 1) CUI: an input model, organized in a structured manner that undergoes transformation; 2) Template: formats the information in the CUI into the target model, here a FUI, using references to entities in the CUI; 3) Template engine: an application that produces an output by using the template structure and replacing reference data with real data from the CUI; 4) Target: the resulting model after executing the transformations, here a FUI.

The example on Listing 1 depicts an extract of a CUI model and Listing 2 shows an extract of a template that makes references to values in the CUI model. In this example, to show the label Applicant and a text field, the page is created by substituting the value SCE1 for the value of the id corresponding to SCE1, which is Applicant; then the value SCE2 for the value of the id corresponding to SCE2, which is Applicant value; and the same substitution for the other values.

As previously explained in Figure 2, BPs are linked with task models, which are then linked with screen components. The association of UI designed templates with the screen components listed in Usi4Biz and the association of templates with code enable covering the entire lifecycle, from BPs until the implemented UIs. Therefore, the organization can select the adopted software development process and associate it with the model-driven traceability method and Usi4Biz. The goal is to let organizations lead their projects as they constantly do and do minor customizations to adopt our method and tool. The aimed result is to provide techniques and a tool that supports linking different models to aid with impact analysis whenever changes are needed in these models. This strategy has been analyzed by top managers from the business department, who have given positive feedback considering cost analysis, feasibility of tool support

Listing 1 Extract of CUI model

```
<cuiModel id="CUI1" name="CUI_Insurance_Basic_Data">
  <window id="SCR1" name="screen_1" width="800" height="600">
    <gridBagBox id="SCF1" name="screen_fragment_1">
     <outputText id="SCE1" name="Applicant" isBold="true"
          textColor="#000000" textFont="Dialog"/>
     <inputText id="SCE2" name="Applicant_value" isEnabled="true"
           textColor="#000000" maxLength="100" isPassword="false"/>
     <button id="SC3" name="Look Up" isEnabled="true" fgColor="#000000"
           bgColor="#ece9d8" textColor="#000000"/>
     <outputText id="SCE4" name="Name" isBold="true" textColor="#000000"
            textFont="Dialog"/>
      <inputText id="SCE5" name="Name_value" isEnabled="true" textColor="#000000"
           maxLength="100" isPassword="false"/>
      . . .
    </gridBagBox>
  </window>
</cuiModel>
```

Listing 2 Extract of Velocity template

```
<table id=$SF1 border=0>
        <tr>
             <td> $SCE1 </td>
             <td> <input type=text name=$SCE2> </td>
             <td> <input type=button name=$SCE3> </td>
             <td> $SCE4</td>
             <td> <input type=text name=$SCE5> </td>
          . . .
    </tr>
</table>
```

and acceptance of change for the examples in which it was applied.

7 Conclusion

This research has resulted in a model-driven approach to link BPs with UIs and explored the advantages of this link to propose a solution for organizations with extensiveBPs. This approach started to become concrete with the implementation of the tool. Inspired by the case study and consolidated by recent researches, the tool will be validated and improved in future case studies with different business profiles to analyze their interest on the strategy and openness to change. The next steps are to improve transformation rules from BP into task model and rules for tracking changes using web services in order to enable interoperability. In addition, metrics can be specified to quantitatively demonstrate the impact of mapping models and keeping track of changes.

Acknowledgements

Our special thanks to the company for allowing us to share information of our joint project. 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

[Ambler, 2002] Ambler, S. W. (2002). Agile Modeling. John Wiley and Sons.
[Aversano et al., 2005] Aversano, L., Bodhuin, T. & Tortorella, M. (2005). In Proc. of SAC2005 pp. 1338–1343, ACM Transactions on Computer Systems.
[Calvary et al., 2003] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L. & Vanderdonckt, J. (2003). Interacting with Computers 15, 289–308.
[Davenport, 1993] Davenport, T. (1993). Process Innovation: Reengineering work through information technology. Harvard Business School Press.
[Devaraj & Kohli, 2003] Devaraj, S. & Kohli, R. (2003). Management Science 49, 273–289.
[Foundation, 2008a] Foundation, A. (2008a). http://db.apache.org/derby.
[Foundation, 2008b] Foundation, A. (2008b). http://velocity.apache.org.
[JBoss, 2008] JBoss (2008). http://www.jboss.org/drools.
[Kruchten, 2000] Kruchten, P. (2000). The Rational Unified Process. Addison-Wesley.
[Limbourg & Vanderdonckt, 2004] Limbourg, Q. & Vanderdonckt, J. (2004). In Engineering Advanced Web Applications pp. 325–338, Rinton Press.
[Microsystems, 2008a] Microsystems, S. (2008a). https://jersey.dev.java.net.
[Microsystems, 2008b] Microsystems, S. (2008b). http://jcp.org/en/jsr/detail?id=173.
[Microsystems, 2008c] Microsystems, S. (2008c). http://java.sun.com/javase/6/swing.
[NetBeans, 2008] NetBeans (2008). http://graph.netbeans.org.
[of Manchester, 2008] of Manchester, U. (2008). http://owlapi.sourceforge.net.
[OMG, 2008] OMG (2008). http://www.omg.org/spec/BPMN/1.1/PDF.
[Pontico et al., 2006] Pontico, F., Farenc, C. & Winckler, M. (2006). In Proc. of TAMODIA 2006 pp. 43–50, Springer-Verlag.
[Protégé, 2008] Protégé (2008). http://protege.cim3.net/cgi-bin/wiki.pl?SWRLTab.
[Rummler et al., 2007] Rummler, A., Grammel, B. & Pohl, C. (2007). In ECMDA Traceability Workshop EEMCS EPrints Service.
[Sousa et al., 2007] Sousa, K., Mendonca, H. & Vanderdonckt, J. (2007). In Proc. of TAMODIA2007 pp. 112–125, Springer-Verlag.
[Sousa et al., 2008] Sousa, K., Mendonca, H. & Vanderdonckt, J. (2008). In 7th International Conference on Computer-Aided Design of User Interfaces CADUI08 Springer-Verlag.
[Sousa et al., 2007] Sousa, K., Mendona, H., Vanderdonckt, J., Rogier, E. & Vandermeulen, J. (2007). In Proc. of ACM SAC2008 pp. 553–560, ACM Press.
[Stolze et al., 2007] Stolze, M., Riand, P., Wallace, M. & Heath, T. (2007). In Proc. of TAMODIA 2007 pp. 2–14, Springer-Verlag.
[Sujitparapitaya et al., 2001] Sujitparapitaya, S., Janz, B. D., Wetherbe, J. C. & Sammet, D. (2001). In Proc. of 34th Hawaii International Conference on System Science IEEE Computer Society.
[Sukaviriya et al., 2007] Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S. & Stolze, M. (2007). In Proc. of Interact07 pp. 165–178, Springer-Verlag.
[University, 2006] University, C. M. (2006). Technical report Software Engineering Institute.
[Vanhooff et al., 2007] Vanhooff, B., Van Baelen, S., Joosen, W. & Berbers, Y. (2007). In Proc. of ECMDA Traceability Workshop EEMCS EPrints Service.
[Vasconcelos et al., 2001] Vasconcelos, A., Caetano, A., Neves, J., Sinogas, P., Mendes, R. & Tribolet, J. (2001). In Proc. of IEEE EDOC IEEE Computer Society Press.