User Interface Derivation from Business Processes: A Model-Driven Approach for Organizational Engineering

Kênia Sousa, Hildeberto Mendonça,
Jean Vanderdonckt
Université catholique de Louvain, Louvain School of Management (LSM), Information Systems Unit (ISYS)
Place des Doyens 1, B-1348 Louvain-La-Neuve, Belgium
{sousa, mendonca, vanderdonckt}@isys.ucl.ac.be

Els Rogier, Joannes Vandermeulen
Namahn – Minding the user throughout
Grensstraat / Rue de la Limite 21 2nd B-1210 Brussels, Belgium
+32 2 209 08 85
{jv,er}@namahn.com

ABSTRACT

This paper defines a model-driven approach for organizational engineering in which user interfaces of information systems are derived from business processes. This approach consists of four steps: business process modeling in the context of organizational engineering, task model derivation from the business process model, task refinement, and user interface model derivation from the task model. Each step contributes to specify and refine mappings between the source and the target model. In this way, each model modification could be adequately propagated in the rest of the chain. By applying this model-driven approach, the user interfaces of the information systems are directly meeting the requirements of the business processes and are no longer decoupled from them. This approach has been validated on a case study in a large bank-insurance company.

Categories and Subject Descriptors

D.2.10 [**Design**]: Methodologies -H.5.2 [**User Interfaces**]: Theory and methods, User-centered design -
I.6.5 [**Model Development**]: modeling methodologies

General Terms

Documentation, Design, Reliability, Experimentation, Human Factors, Standardization, Theory.

Keywords

Business process modeling, model driven engineering, model-driven user interface development, usability, user interface extensible markup language.

1. INTRODUCTION

Most organizations today attempt 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. This kind of representation is called Business Processes, which is motivated by standards, such as ISO [10]. Once deployed and adopted, the business processes receive pressures from the market and even from the workers to be improved. Ralph Smith emphasizes the importance of process improvement as a way to identify and correct weaknesses and alerts that this initiative has become more critical and frequent in the twenty first century [21].

Information Systems (IS) largely contribute to process improvement. The infrastructure available for processing and communication contributes to reduce costs and time, creating a new classification for activities, which are automatic and semiautomatic because they can be completely or partially executed by computers respectively. Today, there is a strong Information Technology (IT) support for business processes, and many solution providers are offering frameworks to develop customized applications instead of generic ones that require organizations to be adapted to adopt them. On the other hand, customized applications allow organizations to be more flexible. This support brings more flexibility but, at the same time, contributes to reduce the cycle of process improvement, leading to constant changes in the applications.

A business process model is an important artifact, but not the only one to be considered in the definition of the application's user interaction. The role of people who are responsible for the activities and what is necessary to do in each one, are essential requirements to conceive all User Interfaces (UI) supporting the roles they play in the organization. Therefore, when the process changes, probably, the software should change accordingly. But, to determine what exactly needs to be changed is subject to a precise requirements control and traceability, which becomes more complex proportionally to the software's size and changing frequency.

In this work, we are concerned about the traceability from the business process until the user interface to help business analysts in predicting the impact of process changes on the user interaction and propose changes in the processes when the user interaction is improved. This strategy is according to the traceability definition, which is "the ability to describe and follow the life of a requirement in both a forward and backward direction" [8], providing a critical support for software maintenance.

To support this special kind of traceability, we apply a model-driven approach that derives UIs from business processes. For this purpose, a set of UI models is used that interact with each other according to the Cameleon Reference Framework [5]. These interactions are able to define traceable links between the business process and the final UI, which are used to show precisely the impacts in both directions.

SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.
Copyright 2008 ACM 978-1-59593-753-7/08/0003…$5.00.

This work has matured in the context of a large company subdivided in the banking and insurance businesses. This subdivision resembles having two organizations in one, which is even a more complex situation to be addressed. We could observe a good adoption of process practices in the whole organization and a high degree of granularity distributed in six different levels. Because of the number of processes, around seventy processes for the main products, they encounter difficulties in identifying how changes in their business processes have an impact on the UIs. In their context, changes in the business processes are very common, especially when there are changes in laws under which their processes are based on. Their existing systems have mostly a sequential flow, resulting from strictly following the business processes and therefore lack considering the user perspective.

2. RELATED WORK

Recent works propose different types of technological support for traceability between business processes and the supporting software systems, such as [1] and [23]. Many of them are founded on Organizational Engineering (OE), which focuses "on understanding the relationships and dependencies between business strategy, business processes and the supporting information systems" [24]. Analyzing strategies to align business processes and software systems, a framework was defined to do the traceability between business strategy, business processes and IS using UML [26]. This framework proposes a diagram called Goal/Process/System (GPS) diagram, which represents the association between these three areas and allows identifying dependencies when changes are performed in any of these areas. The IS diagram, which is also in the last layer of the GPS diagram, is composed of system components. They extended UML to define a notation that has components as the core of IS modeling. The GPS diagram has a high level definition that does not demonstrate how it can express dependencies when the requested changes are very specific and within components. For instance, when users request viewing detailed payment values, where can we find the information that specifies which components are affected for a retail store case study? Is it in all of the Purchasing components; including Purchase Order and Invoice component or only in one of those?

A strategy presented by Aversano to detect misalignment when a change is executed applies two different techniques [2]. First, it considers attributes of objects in order to identify misalignment. Such objects are either process activities or system components (classes and methods). Second, it uses impact analysis to identify objects affected by a performed change. The attributes of objects considered in this work are "technological coverage" and "technological adequacy". These parameters advocate the point of view of the organization's managers and the business process' executors. The impact analysis considers the dependencies between the objects, identifies the types of modifications (modify activity/software component) and the propagation rule that needs to be executed to propagate a change to the connected objects.

This proposal is focused on traceability techniques (e.g. alignment identification and impact analysis) considering the system components, which is commonly found in other approaches. It goes further in details than the previously mentioned work, but it still lacks the support for a user perspective since it is focused on technological and managerial aspects.

There are also approaches that use theories to understand organizations and ISs to help with alignment and communication. The approach applied by Rosenkranz & Holten [20] uses the cybernetic concept of variety as a measure for the complexity of systems (e.g. organizations, ISs), that is, the possible states or patterns of behavior of a system. Following a language-based approach to ISs, they argue that conceptual models can be used to communicate the possible states of a system. They used the MetaMIS approach, an ontology-driven method to bridge communication gaps between the business process and IT departments. It has been found in various researches that such communication gap is very common because one department finds it difficult to understand reports coming from the other department. To deal with this issue, we agree that conceptual models can express the shared understanding of the concepts of the organization.

In a case study, they presented two examples of solutions, such as appointing employees with IT background to work in the business unit, which in our analysis represent short-term solutions that are strongly dependent on the knowledge of specific individuals and, consequently, a misuse of resources, as they have agreed upon in their concluding remarks; such solutions are often applied to address increasing time pressures, common in the banking business.

As we have demonstrated with the previously mentioned research works, many contributions support the alignment of business processes and IS. Results from investigations [7] have demonstrated that for IT to positively affect the organization, ISs must be appropriately used (e.g. use of certain functionality for its intended purpose). This corroborates with the idea that user interaction has 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 researches that already discuss about the enhancement of UI design practice with business process modeling as a starting point, such as experiences reported from the IBM T.J. Watson Research Center [22]. In their work, they point 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. In this work, even though they do not detail how they apply it in their projects, they specify the need to align UI design with business process models. But, it does not mention whether or not UI designers receive business process models as input for their work and the difficulties they face when required to understand process models or to enhance them with human aspects, such as user tasks.

In a further work, they detail their work [23] by explaining their approach to leverage business process models as a starting point of the UI design process. They mention that business process models are not representative enough to communicate with customers or users; therefore they use low-fidelity mock-ups to share the understanding of the process. In their approach, they argue that the information in business processes share some characteristics with the task model and use the business process model as a starting point for UI design. We do not intend to use business process models as communication instruments with customers or users; we agree that they are a good starting point for UI design and can be used by system analysts to conceive the task model.

In general, we believe that many approaches are either more technical or more managerial and lack the concern with the user perspective. The recent approaches that integrate business perspective with UI design are more concerned with specific solutions and devices through the use of rapid prototyping, for instance. On the other hand, we envision a more flexible approach with the use of conceptual models to facilitate communication and knowledge sharing between departments and interoperability of solutions whenever change requests arrive.

In our proposal, we align business process models with task models and explain how the task model is useful to identify how changes on business processes affect the user interaction and vice-versa. In more details, we demonstrate how our proposal is suitable for the context of a large bank-insurance company, which aims at aligning their business processes with IT, but is also concerned with the user experience.

3. ORGANIZATIONAL CONTEXT

The company that we analyzed is decomposed in two main sub-divisions: the insurance and the bank. The insurance is responsible for process engineering, management and implementation of products in the back-end. The bank is responsible for designing screens, usability evaluation, management and implementation of products in the front-end. Their processes are deployed in three main Logical Channels (LC): *home banking*, used by customers at home through the internet; *branches*, used by the bank employees to create and manage contracts; and *back office*, used by the insurance employees to create and manage contracts with the highest permission level.

The main challenge is to make these two sub-divisions communicate clearly and efficiently, especially because they are physically separated. When they contacted us for consulting, their main issue was that the business processes modeled by the re-engineering department were seldom considered by the IS, UI design and human factors departments. As a consequence, they spent a considerable amount of time on meetings to explain the business processes and to make suggestions on UI design, and on quality assurance checks to make sure that the UIs were in accordance with the business processes before sending them to development.

We conducted interviews with the re-engineering, IS, UI design and human factors departments to better understand their current context, their difficulties and needs. The re-engineering team has an organized methodology to design their business process models, which are decomposed in six layers: business domain, end-to-end process, end-to-end process per logical channel, sub-process, activity, and task; and documented in two different documents, a global and a detailed description. 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 and correlate business process layers with UIs elements, which can start with screens until basic objects (e.g. fields). The IS, UI design and human factors departments, on the other hand, find it very complicated to handle these different documents, which they argue requires them to have a deep understanding of the company products and extra time to keep all the spreadsheets updated and consistent, which is not their reality since they are under-staffed to attend various demands, especially for UI design and usability evaluations. As a consequence, business processes are not aligned with UIs and it is difficult to identify the impact that changes on a business process have on UIs and vice-versa.

The issues related to scarceness of product-knowledge and human resources in the UI design and human factors departments lead to two possible scenarios: (i) UI designers do their work in an ad-hoc manner, considering previous designs and insights from professionals more experienced with the products; or (ii) the IS department, with more staff and product knowledge, takes the lead and implements before UI design. In the first case, the business process is seldom followed. In the latter, it is difficult for human factors experts to make sure the outcome of usability evaluation is put into practice. In summary, their issues are related to:

- Lack of correlation between business process and UI design;
- Difficulties in doing impact analysis after changes;
- Difficulties to understand, to find, and to keep updated information spread in many different artifacts.

Therefore, they aim at aligning their 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.

4. BUSINESS AND USER PERSPECTIVES

Most of business process notations have a similar structure to represent the sequence of work and the decomposition of the organizational complexity, such as the Business Process Modeling Notation (BPMN) [16]. Activities, linked with each other, and decision points, providing other paths to follow, have a deterministic graph representation. They can be decomposed in other activities and these activities can be decomposed as well, forming a tree representation, where a new level is created when motivated by the necessity to better describe what to do.

Business processes are often not representative of the flexibility necessary for user interaction, but its structure is important to decide how to make the relationship between business processes and UI explicit in order to address the following goals:

- **User-Centered Design**: Provide usability in information systems used by professionals in their every day work;
- **Traceability**: Support organizations in maintaining their systems aligned with constantly evolving business processes.

The importance of conceptual models for OE [20] has been acknowledged in other works. Such outcome is based on theories that value the importance of exchange of knowledge and clear communication between different departments, which has been proved to be an important factor to the success of both short-term and long-term alignment [27]. Considering the importance of user interaction to make IT add value for organizations, we focus on the alignment of business processes and ISs through a UI conceptual model: the task model, which represents the tasks performed by users when interacting with a system. Figure 1 depicts how the re-engineering department has defined their methodology to design business process models, decomposed in these six layers, on the left of the figure, which are associated with the hierarchical levels of a task model, on the right of the figure. The association starts with the end-to-end process layer because the business domain represents the overview of the process architecture.

To address user-centered design, task models foster a strong representation for UI design because they contain decomposition in a hierarchical structure, which provides an overview of the user interaction useful for decision-making on UI design (e.g. which tasks should be grouped in the same screen); and a variety of temporal ordering of tasks, which delivers the flexibility and representational power that users need while interacting with ISs.

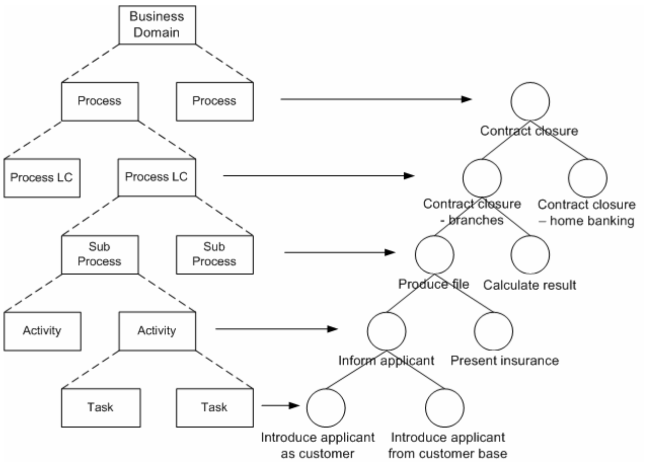


Figure 1. Business Process and Task Model.

With its representativeness for UI design, task models can be used as an efficient conduit between business processes and UI design. Business processes are often an abstract representation of the business tasks, different from task models, which are concrete set of tasks that help UI designers visualize what could happen during user interaction. There are certain characteristics of business processes that make them a limited representation in terms of information needed for UI design: 1) the business process concepts do not consider automation in itself. The one responsible for applying the process decides what should be automated. Therefore, business processes ignore *how* the activity is accomplished, focusing only on *what* should be accomplished. 2) Business processes do not encompass all tasks that are intrinsic of user interaction, such as cancel, save temporarily, undo, etc. 3) In most cases, the business process is not detailed enough to describe individual behavior and even when it is present, the sequence of activities may not represent the user behavior, which has strong influence from the context of use. As a result, business process models can be used as requirements, not directly to UI design, but, in our point of view, for the creation of task models, before UI design. These two models represent two different schools of thought; one is business-oriented, while the other is user-oriented. It is important to recognize that we need to unite both worlds to design UIs of ISs.

As demonstrated by comparing business processes and task models, we argue that using task models to bridge business processes and UI design can help address usability issues. Task models have been used to support requirement analysis, design of ISs and usability evaluation [14], and even to model user roles with different responsibilities and rights in large applications [18], but it has not yet been used as a source for traceability. Furthermore, we demonstrate how our approach can address traceability issues. We propose to correlate business process elements (e.g. sub-process, activity, task) with tasks; and tasks with UI elements (e.g. screens, objects) to identify the impact of changes in any of these models.

Changing from a business perspective to a user-centered perspective, it is necessary to focus on certain aspects of user interaction, such as naming of tasks in the task model must follow a consistent taxonomy. The types of actions used to name tasks are based on a taxonomy introduced by [6] to qualify a UI in terms of abstract actions it supports. The names of tasks are to be created as an expression composed of a verb and the item on which the action is operated. The main types of actions are: start, stop, select, create, delete, modify, move, duplicate, perform, toggle, and view.

Considering the relationships between tasks as an important source for understanding user behavior, it is important to associate relationships in task models and in business processes since the task model is created based on the business process. For instance, *enabling* relationship means that a task T1 has to be finished in order to initiate a task T2; *deterministic choice* means that once one task is initiated the other cannot be accomplished anymore; *suspend/resume* means that a task T2 may interrupt a task T1 before the termination of T1. Once T2 is finished, T1 may be resumed; and *disabling* means that a task T2 may interrupt T1 before the termination of T1, but T1 cannot be resumed after T2 has terminated. When creating a task model based on a business process, the system analyst can substitute enabling for *sequence* relationships; deterministic choice for *decision points*; suspend/resume for *information and warning* relationships; and disabling for *stop and process intervention* relationships.

5. MODEL-DRIVEN APPROACH FROM BUSINESS PROCESS

According to the importance of models for communication and exchange of knowledge, we adopt a model-driven approach for UI design. Some of these approaches use User Interface Description Language (UIDL) to specify models and exchange these specifications between IT professionals; which can be exemplified by UsiXML [11], UIML [1], etc. But a UIDL needs to be grounded on a methodology in order to specify the steps to handle the models according to certain requirements, such as possess a systematic approach and following well-founded guidelines in a repeatable manner [4]. The Cameleon Reference Framework is a flexible approach towards model-driven UI design, in which models are created and mapped using UsiXML. This framework 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).

There are works that focus on mapping these models to generate UIs. Paternò & Santoro [17] specify the relationships between task model and AUI, and between the AUI and its implementation. Vanderdonckt [25] defines a mapping model that contains the mappings between the models and elements of models. There are Model-Driven User Interface Development Environments that map concepts from different models to guide in designing UIs, such as TEALLACH [9], TERESA [15], and UI Pilot [19]. It is not in the scope of this work to detail the different techniques, nor to compare and assess them, but we consider the contributions above as a support for model mapping and the basis for the traceability between the models.

Adapting the Cameleon Reference Framework and using UsiXML models in the context of large organizations, we present a business–driven approach organized in three phases, as depicted in Figure 3. The models are mapped among each other in the proposed sequence as presented in Figure 2, which preserves continuity and supports traceability.

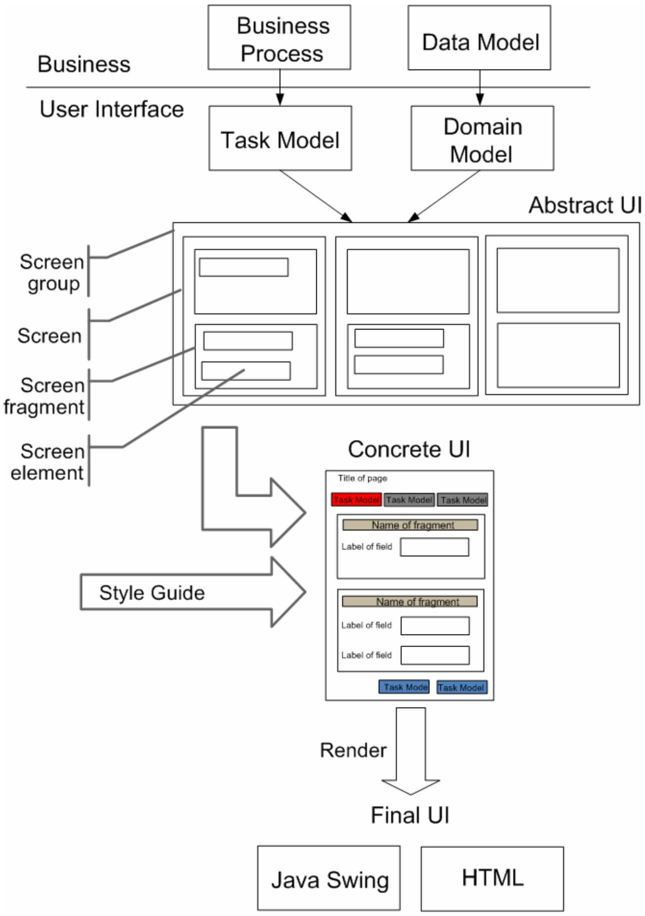


Figure 2. Model-driven approach.

5.1 Conception phase

This phase is dedicated to business analysts modeling business processes that serve as requirements for UI design. It is currently considered a model external to UsiXML (Figure 3 – ①) and it can be created using any available process modelling tool. These tools are able to export their models into XML format, which is appropriate to interchange information with other tools or systems that communicate with UI models (Figure 3 – ②).

5.2 Management phase

The management consists of three modelling levels necessary for UI design (Figure 3 – ③). These models can be managed and traced by specialized tools in order to provide forward engineering to develop the system; reverse engineering to create conceptual models of existing systems; and translation of the same model for a different platform (Figure 3 – ④).

This phase starts when the focus changes from business orientation to user orientation. Business process helps system analysts to conceive the task model. Their work can be aided by tools that use the exported XML and transform it into a UsiXML representation of the task model. Task models should be reviewed by human factors experts in order to make sure that the user perspective was taken into consideration. This revision should be emphasized when a transformation is made.

When the task model is ready along with other conceptual models such as the data model, human factors experts consider them to conceive the AUI. The AUI model visually specifies which screens are needed and which elements (e.g. text, field, command) are within each of them to execute a set of tasks. In the context being studied, the company uses a standard that well resembles the AUI structure. It is comprised of broad components that are detailed until an atomic level, which are: (i) *screen group*, a group of closely related screens; (ii) s*creen,* a state of the user interface when executing a task or part of a task; (iii) *screen fragment,* a container of related elements; *and* (iv) *screen element,* the most atomic component. UI designers use the AUI, the company style guide and platform restrictions to create the CUI model. Each identified screen on the AUI model will be better described in a CUI, which visually resembles the real user experience.

5.3 Application phase

The previous phase conceived the models that are represented in UsiXML. As a mark-up language, it can be processed by a tool (Figure 3 – ⑤) to transform these models into a FUI (Figure 3 – ⑥). Because of the CUI representativeness, each one will be derived into a FUI, when aspects such as architecture, programming languages and infra-structure are taken into account.

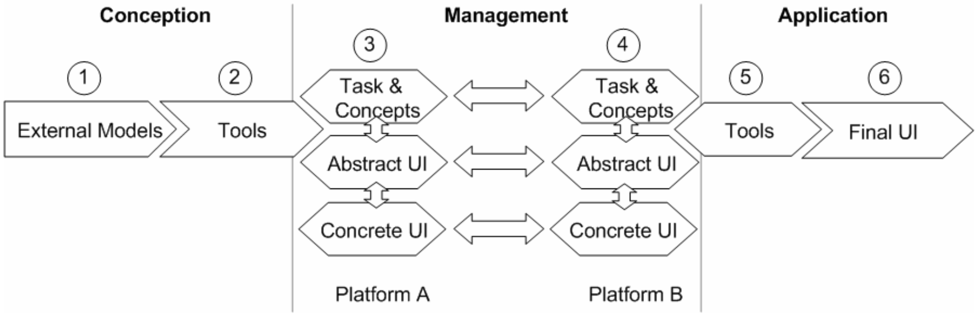


Figure 3. Phases of UI design.

6. TRACEABILITY

The company has a special interest in the traceability goal because of the size of the process and, consequently, the size of their software. The business analyst, concerned with the consistence between them, needs to know which screens should be changed or, at least, reviewed when the business process changes. The human factors expert can also propose changes in the process according to changes made in a UI model to improve the user experience.

The model-driven development provides the traceability opportunity facilitating the creation and maintenance of relationships between models [3]. According to the Query/Views/Transformations from the Object Management Group (OMG), traceability relationships between model elements involved in a transformation are created implicitly, hence, changes to a source model may be propagated to a target model by re-executing the transformation [13].

To illustrate the importance of traceability, we will present an example with one of the main processes in the company, the simulation of insurance contracts. They have decided to make a marketing change: create a new product for insurance contract. Now, they have two options, which are investment and savings. For the process *closure of insurance contracts*, the sub-process *produce contract* (Figure 4) and for its activity *introduce product*, the associated task *inform product* (Figure 5) in the task model is related with the screen fragment *contract data* (Figure 7). With further mapping of model components, it is possible to identify that this screen fragment is part of the screen *insurance contract* and of the screen group *closure of insurance contract* (Figure 6).

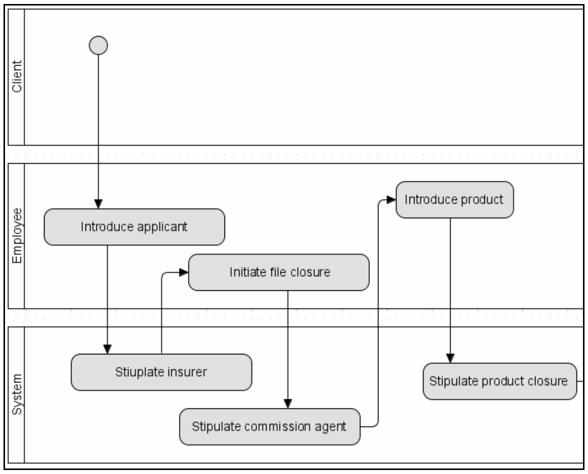


Figure 4. Business process.

The sub-process *produce contract* is composed of seventeen activities, which are partially demonstrated in Figure 4 that contains the activity *introduce product*, detailed in a task model. The task model to *inform product* is partially demonstrated in Figure 5. It specifies that the user introduces the start date, the payment date, the product and the premium formula, shown on the screen fragment *contract data*. The other tasks are performed by the system and they are not shown on the UI.

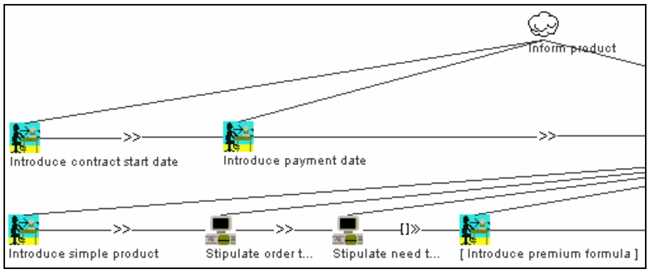


Figure 5. Task model of "inform product".

The screen fragment *contract data* is located on the screen *insurance contract*, which is organized within the screen group *closure of insurance contract*, shown in Figure 6.

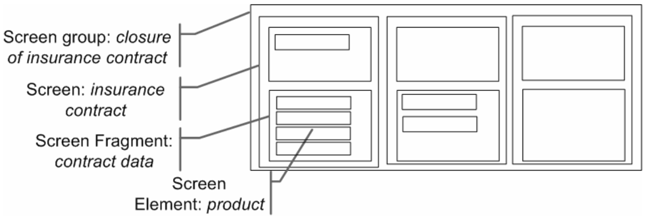
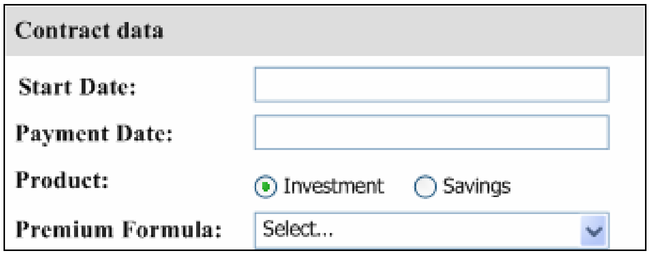

Screen group: *closure of insurance contract*
Screen: *insurance contract*
Screen Fragment: *contract data*
Screen Element: *product*

Figure 6. AUI for closure of insurance contract.

With the AUI, it is already possible to know which UIs are impacted by changes made in the process. From this point on, UI designers are concerned with improvements on the user experience, such as platform characteristics and style guide, considered on the CUI as depicted in Figure 7.



Contract data

Start Date:

Payment Date:

Product: ● Investment ○ Savings

Premium Formula: Select...

Figure 7. Screen fragment for the task "inform product".

Considering a future automation, we propose that when a change is requested, a UI modeling tool (yet to be developed) can be used to find out what kind of change is necessary on screens based on comparing the business process and task models considering their association as depicted in Figure 1.

Depending on the result of comparing business processes with task models, the change on the screen components may vary, as listed below: (i) Add screen / screen fragment – when there is a new sub-process or activity in the business process that is not yet in the task model; (ii) Delete screen / screen fragment – when there is a task in the task model, but a sub-process or activity was taken out of the business process; (iii) Add screen element – when there is a new task in the business process that is not yet in the task model; (iv) Delete screen element – when there is a task in the task model, but it was taken out of the business process; (v) Change order of screen fragments – when the order of activities in the business process and their correlated tasks in the task model were changed; (vi) Change order of screen elements – when the order of tasks in the business process and their correlated tasks in the task model were changed; and (vii) Simple review – when the changes were related to description, or rules, not on the business process or task model structure.

With an existing work on model transformations using UsiXML [12], it is possible to execute transformation in different perspectives of UI design: (i) *reification*, transformation of a high-level requirement into low-level analysis or design, including code generation; (ii) *abstraction*, an extraction of high-level requirement from a set of low-level requirements or from code; (iii) *translation*, a transformation a UI in consequence of a context of use change; and (iv) *reflection,* a transformation of the artifacts of any level onto artifacts of the same level of abstraction, but with different contents. These transformations are possible because mappings are established from any source model to any target model or from any model element to any other using the same UIDL to specify the models.

There are already some tools that allow editing various models in UsiXML, producing (semi-)automatically models and code from and to the four different levels of UI abstraction [11]. But the existing tools do not cover the conception phase, as depicted in Figure 3.

7. ASSESSMENT

The new approach we proposed to the company was largely based upon the methodology they currently use. The way of working is largely the same, except that instead of working with a lot of different artifacts, often not even aligned with each other and with information duplicities, they will work with mapped conceptual models. New models with well delimited scope help them to consider the user perspective and allow them to communicate easier with other departments. The company was concerned with four criteria to assess the model-driven approach for UIs, which are: cost, feasibility, maintainability and acceptance.

Under cost, we first have the cost needed to implement tools to manage the various models. Existing systems for business process modeling and task modeling can be used, but a specific tool to map the models and track the changes still needs to be implemented. The implementation of this tool is crucial for allowing the correct alignment between the business process and the UI models, and also to ensure traceability and synchronization. The cost to apply the new approach includes the training of professionals and the cost with time to actually create and refine the task models.

Within feasibility we look at tool support, flexibility and traceability. Tool support is already present in different commercial and non-commercial tools that allow business process modeling and task modeling, but they might need some extra functionalities that existing tools do not offer yet, as previously mentioned. Flexibility is provided for UI designers since the task model represents a structure with vast space for decision making of what represents screen groups, screens, screen fragments and screen elements. Traceability comes inherent with the adoption of model mapping and thus, makes the application of this approach more appealing for a company with extensive business processes.

Maintainability is simplified when synchronization is built into a tool that tracks model changes. Mapping between the different models allows an easy impact analysis when changes must occur in one of the models.

The acceptance of this approach was conditioned by the ease of professionals learning the task model, which was not common in their working abilities. But once faced with the benefits brought by reducing the number of artifacts, improved communication, better consideration of user perspective and traceability; it made the concern on the learning process not a main issue for acceptance.

8. CONCLUSION

This work presented a model-driven approach for UI design based on business processes. With this approach, models are derived from each other and aligned in order to more efficiently propagate changes when needed. In addition, the user experience is considered in alignment with business needs.

In order to apply this approach, the company needs to train the professionals in task modeling and an external tool needs to be created for task modeling and mapping among the various models. Once the approach starts to be applied, it needs to be monitored to understand the professionals' feedback and make any adjustments when necessary.

Among future works there is the need to develop a tool that transforms business process models in task models in UsiXML and integrate it with existing tools; and the study of the return of investment using this approach.

9. ACKNOWLEDGMENTS

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 and the Similar network of excellence (http://www.similar.cc), the European research task force creating HCI similar to human-human communication of the European Sixth Framework Program.

10. REFERENCES

[1] Ali, M.F., Pérez-Quiñones, M.A. and Abrams M., Building Multi-Platform User Interfaces with UIML. in Multiple User Interfaces: Engineering and Application Framework. John Wiley and Sons, New York, (2003)

[2] Aversano, L. Bodhuin, T., and Tortorella, M. Assessment and impact analysis for aligning business processes and software systems. In *Proc. of SAC'2005,* ACM Press, 2005, 1338–1343.

[3] Aizenbud-Reshef, N., Nolan, B. T., Rubin, J., and Shaham-Gafni, Y. Model traceability. *IBM Systems Journal 45*,3), (2006), 515–526.

[4] Braun, C., Wortmann, F., Hafner, M., and Winter, R. Method construction - a core approach to organizational engineering. In *Proc. of SAC'2005*, ACM Press, new York,2005, 1295–1299.

[5] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers 15*, 3 (June 2003), 289–308.

[6] Constantine, L.L.: Canonical Abstract Prototypes for Ab-

stract Visual and Interaction Design.In: Proc. of 10th Int. Workshop on Design, Specification, and Verification of Interactive Systems DSV-IS’2003 (Funchal, June 11-13, 2003). Lecture Notes in Computer Science, Vol. 2844. Springer-Verlag, Berlin (2003) 1–15. Accessible at http://www.foruse.com/articles/abstract.pdf

[7] Devaraj, S. and Kohli, R. Performance Impacts of Information Technology: Is Actual Usage the Missing Link? *Management Science 49*, 3 (2003), 273–289.

[8] Gotel, O.C.Z. and Finkelstein, A.C.W. An Analysis of the Requirements Traceability Problem. In *Proc. of ICRE’94*, IEEE Computer Society Press, Los Alamitos (1994), 94–101.

[9] Griffiths T., Barclay, P.J., Paton, N.W., McKirdy, J., Kennedy, J.B., Gray, P.D., Cooper, R., Goble, C.A., and Silva, P.P. Teallach: A Model-based user interface development environment for object databases. *Interacting with Computers 14*, 1 (2001), 31–68.

[10] ISO, International Organization for Standardization. http://www.iso.org.

[11] Limbourg, Q., Vanderdonckt, J.: UsiXML: A User Interface Description Language Sup-Porting Multiple Levels of Independence. In: Matera, M., Comai, S. (eds.): Engineering Advanced Web Applications. Rinton Press, Paramus (2004), 325–338.

[12] Limbourg, Q. and Vanderdonckt, J., Multi-Path Transformational Development of User Interfaces with Graph Transformations, Chapter 8, In Human-Centered Software Engineering: Software Architectures and Model-Driven Integration”, Vol. II, Springer HCI Series, Springer-Verlag (2007).

[13] MOF 2.0 Query/Views/Transformations RFP, OMG Document ptc/2005-11-01, Object Management Group, Inc.,revised on November 01, 2005.

[14] Mori, G., Paternò, F., and Santoro, C. CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. *IEEE Transactions on Software Engineering 28*, 8 (2002), 797-813

[15] Mori G., Paternò F., Santoro C. Design and development of multidevice user interfaces through multiple logical descriptions. *IEEE Transactions on Software Engineering 30*, 8 (2004), 507–520.

[16] OMG, Business Process Modeling Notation Specification, 1.0, February, 2006.

[17] Paternò, F., Santoro, C. A unified method for designing interactive systems adaptable to mobile and stationary platforms. *Interacting with Computers 15*, 3 (2003) 349–366.

[18] Penichet, V. M. R., Paternò, F., Gallud, J.A., Lozano, M.D. Collaborative Social Structures and Task Modeling Integration. In *Proc. of* DSVIS' 2006, Springer-Verlag, Berlin (2007), 67–80.

[19] Puerta, A., Micheletti, M., and Mak, A. The UI Pilot: A Model-Based Tool to Guide Early Interface Design. In Proc. of IUI'2005, ACM Press, New York, (2005), 215–222.

[20] Rosenkranz, C. and Holten, R. Combining Cybernetics and Conceptual Modeling – The Concept of Variety in Organizational Engineering. In *Proc. of SAC’2007*, ACM Press, New York (2007), 1228–1233.

[21] Smith, Ralph. Business Process Management and the Balanced Scorecard. John Wiley & Sons, New Jersey, 2007.

[22] Sukaviriya, N., Kumaran, S., Nandi, P., and Heath, T. Integrate Model-driven UI with Business Transformations: Shifting Focus of Model-driven UI. In: *Proc. of MDDAUI’05* (Montego Bay, Jamaica), CEUR Workshop Series, Vol. 159, 2005.

[23] Sukaviriya, N., VibhaSinha, Ramachandra, T., Mani, S., and Stolze, M. User-centered Design & Business Process Modeling: Cross Road in Rapid Prototyping Tools. In Proc. of INTERACT’07, Springer-Verlag, berlin (2007), 165–178.

[24] Tribolet, J., Sousa, P. Editorial Message: Special Track on Organizational Engineering. In *Proc. of SAC’2004*, ACM Press, new York (2004) 1337–1338.

[25] Vanderdonckt, J. A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In *Proc. of CAiSE'05*, Springer-Verlag, Berlin (2005), 16–31.

[26] Vasconcelos, A., Caetano, A., Neves, J., Sinogas, P., Mendes, R., and Tribolet, J. A Framework for Modeling Strategy, Business Processes and Information Systems. In Proc. of IEEE International Enterprise Distributed Object Conference, IEEE Computer Society Press, Los Alamitos (2001).

[27] Wagner, H-T., Beimborn, D., Franke, J., and Weitzel, T. IT Business Alignment and IT Usage in Operational Processes: A Retail Banking Case. In Proc of HICSS’2006, IEEE Computer Society Press, Los Alamitos (2006), 172–173.