# Supporting Requirements in a Traceability Approach between Business Process and User Interfaces

**Kênia Sousa, Hildeberto Mendonça, Jean Vanderdonckt**
Université catholique de Louvain (UCL)
Louvain-la-Neuve, Belgium
kenia.sousa@student.uclouvain.be,
{hildeberto.mendonca,
jean.vanderdonckt}@uclouvain.be)

**Marcelo S. Pimenta**
Universidade Federal do Rio Grande do Sul (UFRGS)
Instituto de Informática,
Porto Alegre, RS
mpimenta@inf.ufrgs.br

**ABSTRACT**
This paper presents the impact of the traceability from Business Process (BP) to User Interfaces (UI) in software requirements. The application of changes made on BP or on system UIs may have an impact on software requirements, which then requires updating related models in order to coherently enable changes. The goal of mapping software requirements with task models, as an extension to an existing traceability approach, is to guarantee that UIs are aligned with requirements. This paper proposes a traceability framework on how to map requirements with UI models and analyzes the impact of changes on models of business-driven enterprise applications.

**Author Keywords**
Business Process Modeling, Model-Driven Engineering, User-Centered Design, Requirements Engineering.

**ACM Classification Keywords**
H.1 [**Information Systems]**: Models and Principles, H.5.2 [**Information interfaces and presentation (e.g., HCI)**]: User Interfaces, D.2.1 [**Software Engineering**]: Requirements/Specifications, J.1 [**Administrative Data**]: Processing: Business, Financial.

**INTRODUCTION**
Both Software Engineering and Human-Computer Interaction communities are unanimous in recognizing that *usability* is nowadays a quality criteria, at least as important as *utility* for interactive computerized systems.

Some authors converge towards the central idea that to design more useful and usable systems we must better understand the tasks people undertake and better apply our understanding of tasks in the design process. For this goal, activity theory is useful since it interprets "consciousness as the product of an individual's interactions with people and artifacts in the context of everyday practical activity". [7] Considering its emphasis on human intentions, the use of task models addresses the needs of specific users.

Task modeling has been considered in a user-centered perspective as a valuable activity at the early stages of the software life cycle, mainly for eliciting functional requirements of interactive systems, but the actual integration of task models and system requirements (and even with BP modeling) remains an open question. The main identified issue is that the traditional software engineering methodologies and user interface (UI) design approaches do not provide any guidance on how to consider impacts from changes on BP or usability-oriented requirements into the software requirements definition.

Task models usually describe what a user does when s/he carry out tasks and, in some cases, how a user performs tasks. However, they tend to neglect the complementary organizational aspects of usage situations, usually described by means of BP models. Indeed, major system usability problems arise from this lack of integration among BP, functional requirements and UI-oriented usability requirements. For this reason, our goal is to define how software requirements should be considered when tracking changes from BP until UI models by proposing a traceability framework.

The second section of this paper presents definitions of the main concepts and the association of BP models with task models, UI models. The third section introduces requirements and how it is considered in the traceability framework. The fourth section presents an example from a case study. The fifth section presents related works. Finally, we finish with a conclusion and some ideas for future work.

**MODEL-DRIVEN APPROACH FOR BUSINESS PROCESS**
By adopting a model-driven approach for UI design, we take advantage of the importance of models for communicating and exchanging knowledge. Even though business processes are often not representative of the flexibility necessary for user interaction, its structure is suitably similar to the hierarchical structure in task models, which is important to decide how to make the relationship


IHC 2008, October 21–24, 2008, Porto Allegre, Brazil.

between business processes and UI explicit in order to address both traceability and user-centered design, as explained in the following sub-sections.

### Business Process and Task Models

Task analysis has emerged from ergonomics as an important aid to interactive systems design because it is an empirical method to understand how people carry out their tasks. A task analysis produces an explicit model of tasks in a domain, called Task Model. In classical task models, we can use different levels of abstraction to represent the complete task hierarchical structure for an entire application: the highest level contains the tasks; which can be split in subtasks or actions in the intermediary levels until the lowest level, which contains the physical non-decomposable actions directly associated with devices.

Most of user task analysis methods have a common point of view: they emphasize individual aspects in tasks performing. However, unlike procedures that are executed by machines, tasks exist in social organizational settings. Organizations are made up of social actors who cooperate to achieve goals that they are unable to achieve in isolation since actors depend on others for some part of their tasks. Since different social settings can lead to different choices on how to achieve goals and execute specific tasks, the understanding of BP models is very important.

Business process is "a structured, measured set of activities designed to produce a specific output for a particular customer or market." [4] A process is an ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs. Taking a process approach implies adopting the customer's point of view. Processes are the structure by which an organization does what is necessary to produce value for its customers.

Business processes could not be directly associated to UIs because they are often an abstract representation of the business tasks, and there are certain characteristics of business processes that make them a limited representation 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, not exactly how the activity is 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, we have united both models to design UIs for applications that support BPs, called business-driven enterprise applications.

### Task models and UI models

To define the user interaction, a task model is then used to describe how tasks can be performed to reach the users' goals when interacting with the system. These activities contain essential information to conceive all 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 [8] specify the relationships between task model and AUI, and between the AUI and its implementation. Vanderdonckt [11] defines a mapping model that contains the mappings between the models and their elements. It is not in the scope of this paper to detail or compare different techniques, but we consider them as a support for model mapping and the basis for the traceability between the models.

### BP models and UI models

The different levels of the business process models are associated with UI models through the hierarchical levels of task models, depicted in **Figure 1** (for space reasons, it depicts only the mappings between these models, not detailed information on the content).

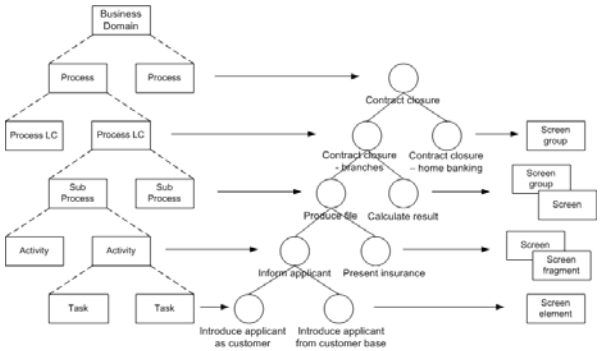


Figure 1. Association of business process with user interfaces.

The association is done by mapping each level of the BP with levels in the task models, then the task model is associated with screens components, which are: *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 [10]. The association starts with the second layer of the BP model because the business domain represents the overview of the process architecture. The association from task models starts in the second level of because the first level is often an abstract task useful for grouping, not representative for screen organization.

## MODELS AND REQUIREMENTS

A usability requirements specification defines measures and criterion levels for effectiveness, efficiency and satisfaction, which must be reached during the software development [6]. In practice, usability metrics are measures of the support that the system provides for users' tasks. Thus, an understanding of these tasks is a necessary condition for the statement of usability requirements. Therefore, we directly associate requirements with task models, taking advantage of the mappings between models in order to guarantee that UIs are also aligned with requirements and facilitate

tracking changes in business-driven enterprise applications.

Since usability requirements can measure software quality assurance, we consider them as part of software requirements. Such approach is preferable for testing usability since early in the development and it demands starting from a clear understanding of requirements. Even though usability requirements are intrinsically related to UIs, our approach is not restricted to them.

In this solution (**Figure 2**), when changes are made on BP models, they may impact on requirements or on task models, or even in both simultaneously. Whenever there are impacts in one of them, the other one is alerted with a warning of possible needs for update. In addition, changes on task models have impacts on UIs. On another scenario, when changes are made first on UIs, the impact is on task models, which in their turn impact BP models and also alerts requirements for possible updates.

The direct alignment of task models with requirements brings forward the importance of those requirements related with user interaction. This is primordial for large organizations (e.g. industries, banks) that have complex business processes, which must be supported by thousands of user interfaces in their applications. In addition, such traceability can prevent requirements from being forgotten in documents and, thus, reinforcing its consideration during the development cycle. And such formalization and alignment avoids the mistaken description of tasks and UI inside requirements specifications.

Our framework is aligned with a definition of requirements traceability found in [9], which refers to the ability to define, capture and follow the traces left by requirements on other elements of the software and the traces left by those elements on requirements.

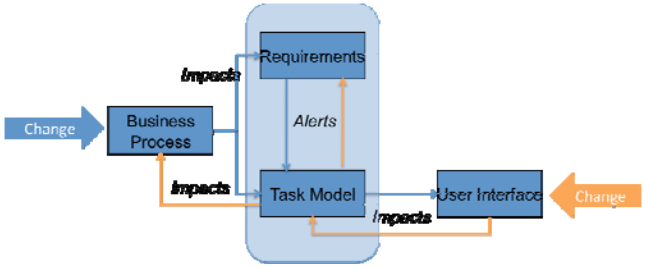


Figure 2. Requirements traceability with models

## REQUIREMENTS IN A MODEL-DRIVEN APPROACH

To better explain our strategy, we present an example of the impact of supporting requirements in a model-driven approach. This example was extracted from a case study with a large bank-insurance organization facing difficulties in relating their business processes with UI design.

After reviews in the business processes resulting from new policies, the process for insurance contract, and more specifically the 'produce file' sub-process was updated by creating new activities for medical acceptance. This change in the business process impacts the related task model with the need to create new tasks. Consequently, the changes in the task model impact the UIs with the need to create new UIs for the newly created tasks.

For requirements traceability, we have analyzed some requirements that could be impacted by this change in the business process. To exemplify, we have considered one requirement, which is associated to the insurance contract business process: "If the customer doesn't have a bank account, the user in the branch can open a temporary account." This impacted requirement alerts the associated task model with a warning of possible needs for update. Therefore, the newly created tasks, and consequently, the related screens have also to accept insurance applicants logging in to control their medical information, even if they are not customers in the bank, without an account.

The mappings of the models is done according to a previously defined method [10] that proposes: 1) the generation of task models based on business processes, thus, automatically mapping these models; 2) listing of screen components based on task models and associate the different screen components with tasks. Extending these mappings, requirements are directly mapped to task models, thus inheriting the association with business processes.

Mapping the UI models is supported by UsiXML [11], a UI definition language, which provides support to represent models in a structured form and supports the flexibility necessary for defining model-driven UIs. Mappings are also founded on the Cameleon Reference Framework [2], extended with the addition of business process modelling and communication between models with tools, with mappings described in [10]. BP models are external to UsiXML and they 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.

As gained benefits, without the consideration of requirements, the change in the business process would generate changes on the UIs related solely to the creation of new screens. The inclusion of requirements traceability in this strategy facilitates the application of requirements throughout the system development, maintenance and evolution, thus avoiding them to be applied only partially or even being entirely forgotten in the myriad of several change requests.

## RELATED WORK

Requirements traceability, as proposed in [5], presents a structure that associates requirements with human sources responsible for these requirements and also for the application design and code. It differentiates artifact-based (it makes use of relations between requirements artifacts and its high level design artifacts) and personnel-based requirements traceability (it makes uses of relations

between agents and requirements artifacts). Since we will include human sources in our traceability framework in a future and more detailed work, we have decided to focus this paper on the conventional form of artifact-based requirements traceability.

In [9], the author advocates that using formal specification languages to describe the system may increase traceability efficiency since it facilitates identifying which traces to follow for a particular element. This kind of formalization is well aligned with our proposal since we adopt UsiXML. His traceability work goes beyond and addresses the informal needs of requirements tracing since he argues that the search for important information related to requirements is guided by necessity, not by pre-defined structures. We agree to some extent to this statement, however we aim at considering informal aspects of traceability in future works.

There is work in user-centered design that addresses tracing requirements of a system to the conceptual architecture of that system [3], which is concerned with user interaction. While this is common in IT initiatives, on the other hand, aligning requirements with business applications is closer to our proposal, such as the integrated approach presented in [1]. Even though this and many other contributions support the alignment of business processes and IS; they still lack concerns on the UI.

With a pragmatic point of view, our approach is envisioned as more flexible because of the use of conceptual models to facilitate communication and knowledge sharing between departments; and, most of all, to provide a traceability framework with the ability of being used by several organizations that adopt heterogeneous solutions for organizing and handling their models when facing change requests.

## CONCLUSION

This paper presented a model-driven approach to link business processes, UI models and requirements. 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.

The experience in a large bank/insurance company enabled us to propose a solution for aligning business processes and UIs of their supporting systems, a major issue in this company as well as in many others in the competitive business word. This reality encourages us to validate this new approach considering requirements in large organizations aiming at analyzing their interest on this strategy and their openness to change.

The next step is to analyze requirements engineering modelling approaches in order to select a suitable approach. This choice depends on the kind of analysis and the reasoning offered by the approach in order to facilitate the link between requirements and the other models. Another important aspect to be analyzed is the acceptability from business analysts towards changes proposed in UIs that may have impacts on their business processes, considering organizational policies, hierarchy, power, marketing strategies, product knowledge, etc.


## ACKNOWLEDGMENTS

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


## REFERENCES


1. 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, pp. 1338–1343.
2. 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, 2003, pp. 289–308.
3. Campos, P. Task-Driven Tools for Requirements Engineering. In 13th International Requirements Engineering Conference, Doctoral Consortium, 2005.
4. Davenport, T. Process Innovation: Reengineering work through information technology, Harvard Business School Press, Boston, 1993.
5. Gotel, O., Filkenstein, A. Extending Requirements Traceability Through Contribution Structures. Technical Report, Department of Computer Science, City University, London, UK, 1997.
6. ISO 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability. Switzerland, 2008.
7. Kaptelinin, V., Nardi, B. A. Acting with technology: Activity theory and interaction design. First Monday 12(4), 2007.
8. Paternò, F., Santoro, C. A unified method for designing interactive systems adaptable to mobile and stationary platforms. Interacting with Computers 15, 3, 2003, pp. 349–366.
9. Pinheiro, F. A. C. Formal and Informal Aspects of Requirements Tracing. WER, 2000, pp. 1-21.
10. Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., Vandermeulen, J. User Interface Derivation from Business Processes: A Model-Driven Approach for Organization engineering. In Proc. of SAC'2008, ACM Press, 2008, pp. 553-560.
11. Vanderdonckt, J. A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In Proc. of CAiSE'05, Springer-Verlag, Berlin, 2005, pp. 16–31.