A Model-Driven Approach for User Interface - Business Process Alignment

by Kênia Soares Sousa

Thesis submitted in fulfillment of the requirements for the degree of

Doctor in Economics and Management Sciences

of the Université catholique de Louvain

PhD Jury:

Prof. Jean Vanderdonckt, UCL, Belgium, Advisor

Prof. Gaëlle Calvary, UJF - Grenoble INP, France, Examiner

Prof. Stéphane Faulkner, FUNDP, Belgium, Examiner

Prof. Elizabeth Furtado, UNIFOR, Brazil, Examiner

Prof. Manuel Kolp, UCL, Belgium, Examiner

Dr. Noi Sukaviriya, IBM Research, United States, Examiner

Prof. Philippe Chevalier, UCL, Belgium, President of the Jury

With all my love to my grandma Teca,
you live forever in our hearts.
Com todo o meu amor para Teca,
minha vozinha do coração

Acknowledgments

My sincere gratitude to all those special people who are in my life:

First of all, I thank God and Our Lady for blessing and protecting me with health, love, motivation and strength for this great challenge in my life.

To my dear family, my husband Beto, my mom Isabel, my dad Kerginaldo Júnior, my brother Hamilton, my sister Natalia (our designer of the symbols for the task and UI models), my grandparents Teresa, Soares, Teresinha and Kerginaldo, my in-laws Terezinha and Hildeberto and family, my uncles and aunts Jerônimo, Miguel, James, Márcio, Eugênio, Patricia, George, Ily and families. Your love just makes me so happy, I could not ask for more.

A more than special thanks to my love Beto, you support and encourage me every single day of this journey with so much love, patience and creativity. You are my strength, you have made all this possible with your big heart. You are the butter to my bread, the breath to my life!

To Jean, my deepest gratitude, for having encouraged the idea of my PhD since the first day and for supporting us in every way, cultural, personal, and, especially for our research discussions, ideas, and endeavors.

To Elizabeth Furtado, Vasco Furtado and Plácido Pinheiro, who encouraged us to come to Europe to do our PhD and are examples of determination and success.

To the HCI community in Brazil, especially to the ones I have met throughout these years of research and have shared with me their enthusiasm with this area: Simone Barbosa (who was part of my Master dissertation jury), Clarisse de Souza and Jair Leite.

To Brian Henderson-Sellers and César González-Pérez, first for sharing at first-hand their work on the ISO 24744 and also for analyzing our evaluation of the visual notation. We specially thank them for having always been very prompt to discuss with fast replies and attention to our comments.

To the people in the Namahn project that enabled interactions with the Fortis bank in Belgium, especially Joannes Vandermeulen and Els Rogier.

To the people in Belgacom who were extremely open and supportive with this challenging project, especially Didier Audoor, Gert Bamps, Johan De Rop, Benoit Godenir, Odon Kalombo, Daniel Guilliams, Pascal Hendrickx, Upasana Khanna, Tom Leenkneght, Dominique Driessen, Gregory Bos and all others who contributed in different ways for the success of this project.

A special thanks to Amandine Lievyns, who was very enthusiastic and committed to achieving our goals, and has become a great friend.

To the Midas project that allowed meetings with professors, especially Carla Maria Dal Sasso Freitas, Luciana Porcher Nedel and Marcelo Soares Pimenta, who enriched our research with a multi-disciplinary aspect.

To Bart Van Den Meersche, Martine Broekaert, Philippe Spaas, Martine Lewi and Hans Van Mingroot from IBM Belgium, who have been so receptive and attentive with me throughout the PhD Fellowship program.

To Noi Sukaviriya, Rick Hull, John Vergo, David Cohn and their team at the IBM Watson Research Center, including Anil Nigam, Fred Wu, Mark Linehan, Stacy Hobson and Terry Heath for their warm welcoming into their team to collaborate in such an enriching environment.

To the members of my committee, Professors Elizabeth Furtado, Gaëlle Calvary, Manuel Kolp, Noi Sukaviriya, Stéphane Faulkner and Philippe Chevalier for accepting to participate in the jury of this thesis and contributing with their knowledge and attention.

To all professors, administrators, researchers, students and friends at Louvain School of Management, especially Vi, Sodany, Silvia, Priscilla, who have made it more fun; Philippe Chevalier and Dominique Warte, who have supported me throughout this process; Marco Winckler and Silvia Abrahão, who were, from the beginning, a great source of inspiration for me.

To my dearest friend Joana Resende, who has received us in her life as part of her family and who, with such an open heart, has inspired me to be a better person.

This thesis was accomplished thanks to the support of:

The Program Alban, the European Union Program of High Level Scholarships for Latin America, under scholarship number E06D103843BR from September 2006 until September 2009.

The SIMILAR network of excellence, the European research task force supported by the 6th Framework Program of the European Commission, under contract FP6-IST1-2003-507609 (www.similar.cc).

The UsiXML Consortium and the UsiXML project for User Interface eXtensible Markup Language (www.usixml.org).

The IBM PhD Fellowship Awards Program, an intensely competitive worldwide program, which has honored me with a fellowship during the 2009-2010 academic year.

Contents

List of Figures

List of Tables

Abstract

Organizations are adopting processes as an efficient tool to manage and control the business. Naturally, business processes are an important source of requirements for system analysts and developers when designing and implementing systems. These systems cover more and more business activities, gradually automating and improving the overall efficiency of the organization. This automation effort tends to be done rapidly to follow the business dynamism, however the alignment between business and IT becomes overshadowed. The initiative to keep processes and systems aligned came from IT that developed techniques to control the alignment of business processes and their systems, which is the enthusiasm behind the Service Oriented Architecture. However, there lacks a solution that addresses a major aspect of systems: their user interfaces. The negative impact of focusing only on functional aspects is that many changes on business processes that affect user interfaces are not carefully considered. Therefore, our solution is a framework for aligning business processes with user interfaces of systems by adopting a user-centered perspective. This framework establishes an unbroken chain of links between business processes models, task models and abstract representations of user interfaces. Once the models are linked, the framework is able to predict the impact that any change on these models may produce in other models. Such support is targeted at large organizations in order to enable them to be more capable of managing those links with consistency. This approach is an innovative strategy that integrates researches on interaction design and business process management with practical implications. Its main differentials include using simple models; considering light actions; preserving the independence of technology; and adopting a human-oriented approach assuring that every managed information impacts people and not only systems, thus enabling fast adaptation to the business dynamism.

Chapter 1

Introduction

"Architect the lines, not the boxes.", Bruce Robertson, research vice president at Gartner on a new approach for Enterprise Architecture, August, 2009.

This thesis aims to bring a deeper concern on user interaction into the reality of managing information systems that support Business Processes (BP), which are known as enterprise systems.

In this context, the research goal of this work is to define a framework for the alignment of business processes with user interfaces of enterprise systems by adopting a model-driven and user-centered approach. This chapter explains the main research topic of this thesis.

1.1 Research Motivation

Most large organizations represent the way they work in order to teach their professionals 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 (BP), which is motivated by standards, such as ISO [44] and OMG [86].

Davenport [20] defines a *business process* as "a structured, measured set of activities designed to produce a specific output for a particular customer or market. It implies a strong emphasis on how work is done within an organization, in contrast to a product focus' emphasis on what. A process is thus a specific ordering of work activities across time and space, with a beginning and an end, and clearly defined inputs and outputs: a structure for action. [...] 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."

Once deployed and adopted in an organization, business processes receive pressure 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 [108].

Information Systems (IS), more specifically *Enterprise Systems* (ES), "support and integrate a full range of business processes, uniting functional data and making this data visible across the organization in real time" [122]. They largely contribute to process improvement and the IS field has acknowledged the need to design and develop IS to support business, approximately twenty years ago [69]. The IS infrastructure available contributes to reduce costs and time when changing processes, thus creating a new classification for activities, which are automatic and semi-automatic because they can be, respectively, completely or partially executed by computers, supported by Information Technology (IT).

The use of IT has evolved over time from its traditional use as administrative support towards a more strategic role to enforce BP. Today, there is a strong IT support for business processes, in which many solution providers are offering frameworks to develop customized systems, thus allowing organizations to be more flexible; instead of offering generic systems that require organizations to change before adopting them. This support brings more flexibility but, at the same time, contributes to reduce the cycle of process improvement, leading to constant changes in the systems. Therefore, 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. A new initiative has emerged to solve these issues: Business Process Management (BPM) is concerned with managing change to improve business processes [89]. In the 2007 Aberdeen Report we found that 46% of enterprises were using BPM technologies and strategies with an additional 39% planning to use BPM now, out of over 3,600 collected survey responses [1]. To understand how this interest has provided support for large organizations and what aspects are still missing in current proposals, we analyze some works from IT-Business Alignment and Model-Driven UI Development.

1.1.1 IT-Business Alignment

IT-Business Alignment refers to applying IT in harmony with business strategies, goals and needs [67]. It supports this research with sound approaches to link BP with enterprise systems.

It is observed in the IT domain that most researches are focused on specifying the association between models from business and IT to support propagating changes ([7], [134]). Another observation is that many of these researches use software engineering models to address alignment issues [132], others are technology oriented [102], while others are focused on the IT infrastructure [12].

These kinds of strategies lack the consideration of a major aspect of information systems: their User Interfaces (UI). The identification of this important need and the fact that enterprise systems are designed based on BP models have led us to analyze how some works on model-driven UI development find interest on business process modeling.

1.1.2 Model-Driven UI Development

Model-Driven User Interface Development (MD-UID) consists of the development of UIs using models and applying a series of transformations between models in order to generate UIs [88]. MD-UID supports this research with the knowledge on models, which makes the proposed solution independent of the technology adopted in enterprise systems, largely differentiated within organizations. It also supports the user-centered perspective that brings a viewpoint different from the one usually present in solutions for enterprise systems.

It has been agreed by academics and practitioners that the potential of IT in enterprise systems depends on how it is used, which is directly influenced by the UI [127]. Therefore, some researches on model-driven UI design and development have found interest in business processes and enterprise systems, which for a long time have not been considered in such a domain.

This growing interest has been verified in [124], which focuses on designing low-fidelity prototypes based on business process models and on some works that advocate for hybrid approaches, combining task models, UI models and process models [120], [94], [58]. However, there are some important characteristics still lacking for large corporations aiming to align business processes with UIs, such as the need for deeper concern with user interaction and people in general, not only with user interfaces and technology.

1.1.3 UI-Business Alignment

Researches on the alignment of BPs with Information Technology (IT) have increased in the last decade because of the improvements derived from the link between BPs and enterprise systems. Successful Chief Information Officers (CIOs) are seen as those who bring innovation and promote the growth and profitability of the organizations. And BPs have been recognized as enablers of innovation because they are tangible representations of strategic decisions translated into executable operations. The IBM CIO Survey in 2008 [42] analyzed 2,500 CIOs worldwide working in organizations with a high profit growth between 2004 and 2007 (called "High-growth CIOs") and those working in organizations with a low profit growth (called "Low-growth CIOs"). This survey reported that "to innovate, High-growth CIOs actively integrate business and IT across the organization 94 percent more often than Low-growth CIOs".

In cases of lack of integration between business and IT, large organizations have faced severe problems when their employees do not follow certain established policies and laws because they might have been designed just in BP models, but not highlighted nor transferred to enterprise systems. And they also face problems when the evolution of enterprise systems is faster than the evolution of the business processes supporting it. Organizations that face these kinds of issues have not employed any alignment strategy or they apply simplistic strategies that do not cover different organizational contexts. Aiming to solve that, IT-Business alignment has appeared in the CIOs concern top ten list since 2004 [68].

Several recent strategies on IT-Business Alignment and MD-UID address impacts on IT infra-structure and on the functional core with solutions that automate the generation of systems to provide alignment. Such strategies usually link business process elements directly with screens. When the BPs and systems formally exist, the strategies used to create links are also simplistic because they are usually technology dependent. For instance, depending on the architectural decision, one option is that process elements can be linked to screens. This can be done by creating references to screens within the XML of processes [84]. Another option is that screens are linked to process elements. This can be done by using the MVC pattern and indicating in the controller layer which process activity is being performed [106]. In both cases there is a unidirectional strategy, as depicted in Figure 1.1. In the worst case scenario, business processes are in people's minds and it is impossible to create any formal links.

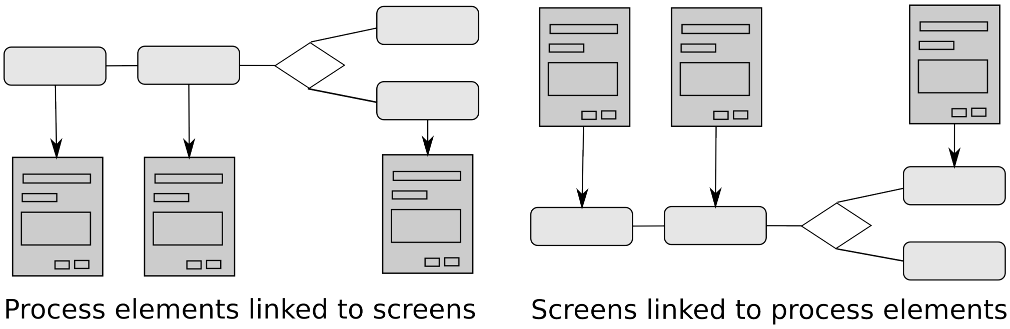

Figure 1.1: Links between process elements and screens

Comprising a larger scope of direct types of links, bidirectional strategies may define different types of cardinality (Figure 1.2). A one-to-one (1-1) cardinality, which links one business activity to one screen, is optimistic and assumes business process automation with UI generation. In a more pragmatic viewpoint, a one-to-many (1-N) cardinality links one process activity to several screens; and a many-to-one (N-1) cardinality links several business activities to one screen. These current approaches easily ignore a more realistic viewpoint, in which a many-to-many (N-N) cardinality links several business activities to several screens, even in different systems, including recent and legacy applications. This framework also covers these kinds of applications because there are several organizations that already have their BPs and systems and cannot afford to re-develop their solutions with new technologies, but need to analyze impact of changes on their BPs and systems.

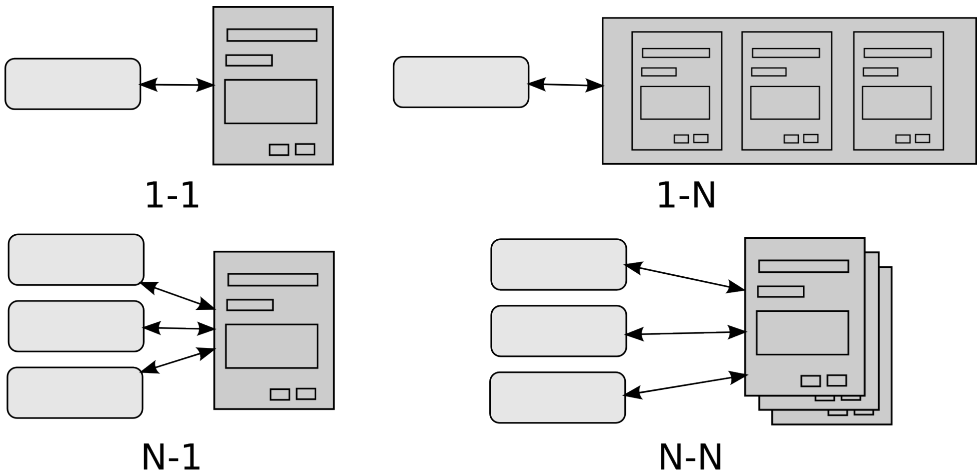

Figure 1.2: Cardinality of links between process elements and screens

Aligning BPs with enterprise systems enables organizations to execute changes with flexibility. However, it is no longer only system processing time that is pivotal for large organizations to change and improve their services for customers. To corroborate this, a research from the Center of Excellence in Process-Efficient Technology has presented results of a new approach to achieve alignment. This new approach is called Process-Centered Design (PCD), a design methodology that proposes a business centric approach for designing user interfaces. Using this approach, they have done a study, which was conducted with 29 companies that had their enterprise systems developed aligning business processes with UIs and 23 companies with systems developed using traditional approaches. This study has shown that the alignment primarily contributed to 100 to 200 percent user productivity gains [41].

Therefore, we have innovated by aligning IT with BPs through UIs by originally calling the term *UI-Business Alignment*. The UI-Business alignment uses task models to bridge these distinct domains in a level that is abstract enough to be technology-independent.

Task models represent users tasks while interacting with systems and they have been extensively used so far for model-driven UI generation, usability tests, UI simulation, as a source to improve UI design through different techniques; but our framework is the first one to use task model as a source of traceability, one of the main differentials in the framework.

The UI-Business Alignment Framework analyzes impact of changes independent of the way BPs are structured and modeled, and how UIs are designed and developed. Here, the concept of *alignment* [39] addresses the fit between the external business arena (i.e. business process models) and the internal structure (i.e. UIs of enterprise systems) and the integration of these structures (i.e. link between BP with UIs).

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

To give an overview of the UI-Business Alignment Framework, we highlight its characteristics:

Vision for both user interfaces and user interaction: It provides a strategy in which we can visually see what is specified in business activities through user interfaces and its components. The use of the task model implies more concern with the user interaction. Therefore, this framework supports the core of interaction design: "put the user first, keep the user in the center and remember the user at the end" [24].

Generic structure: It accepts business activities with different levels of granularity: From an operational perspective, the activities are defined until the smallest action. From a process perspective, the activities specify what is performed and not details of how it is performed [13]. The acceptance of different BP structures is primordial because this granularity may vary depending on the business context. It also accepts UIs designed and developed with different technologies by adopting an abstract representation of UIs.

Concern with people: It opens a new channel for business process improvement that can start with actual users of enterprise systems (bank employees using the system at the bank agency or customers using electronic system on the web at home). Their participation as active agents aids in solving detected issues in business processes through UIs, thus collaborating to address a major business goal to improve productivity within the organization.

Support of a software platform: It is supported by a platform with functionalities that assist in creating a network of links between the models and identifying the impact of changes that serves to provide traceability and maintain consistency between business process, task and UI models.

1.2 Thesis

1.2.1 Thesis Statement

This thesis defines a conceptual and methodological framework and a software platform for a corporate environment that addresses the alignment of business processes with user interfaces of enterprise systems using task models as a cornerstone.

A *framework* is a structure used to solve complex issues. In this context, it is a structure composed of: basic *concepts*, defining the elements of the models involved in the alignment of business processes and user interfaces; a *methodology*, defining the actions to achieve this alignment; and the specification of a *platform*, detailing the main functionalities of a software aimed towards this alignment. A task model represents a *cornerstone* because it occupies a central position in this framework. First, because the creation and changes on this structure can start from any model and the alignment is maintained in reference to this cornerstone. Second, because task models bring the *user perspective* into a solution aimed to bridge the gap between the business and the user interface communities.

1.2.2 Hypotheses

The working hypotheses that guide this investigation are:

Integration between business processes and user interfaces: The main hypothesis of this work is that there is a strong influence from business processes towards how UIs are conceived. We argue that this hypothesis is supported by the influence from business process models on the development of enterprise systems, demonstrated through enterprise architectures as a way to describe the logical organization of business processes and IT infrastructure, originally described as early as in the 80's by Zachman [140].

Adoption of a model-driven engineering paradigm: This framework follows a Model-driven engineering paradigm because it is independent of technology, thus, the use of models makes the framework applicable in a wide range of platform viewpoints [88]. Models are used to direct planning, communicate, exchange knowledge and maintain enterprise systems represented through these models. With an explicit definition of meta-models and structuring rules for model transformation to convert one model into another facilitates instantiation, mapping, maintenance and reuse of these models.

Consideration of business process, task and user interface models: The core models of the framework include three main models: Business process models used to specify which operations are to be performed by the organization to achieve specific goals. Task models used to specify how these operations are performed by the business process performers. UI models to specify how these performers do their work interacting with enterprise systems. The main assumption when adopting models is that they do not bring extra work with their formal manner, but that the time spent on their construction is paid off by their reuse, reduction of complexity of the domain, increased traceability and time to market, and the improved value of the IS, as advocated by Larson [62].

Follows user-centered design approach: User-centered design (UCD) is an approach to design systems that focuses on using information about the people who will use the product, that is, the users [47]. UCD methods focus on users through the planning, design and development of a product. The standard ISO 13407: Human-centered design process is referenced by several user-centered design methods, and its main focus is aligned with existing model-driven approaches (e.g. [16, 124]). These approaches adopt models useful to describe users, their tasks, the physical environment where they are located, and other information directly related with them and their interaction with systems. With a truly user-centered approach, it is possible to organize functionality in a manner that supports and optimizes the work performance of real users in their real work environment [70]. Therefore, the framework adopts UCD techniques to drive the methodology.

Application in the context of large organizations: The framework maps these models in order to provide a stronger relationship between business processes and UIs. With this mapping, the framework is able to provide a solution that enables organizations to adapt efficiently to business challenges in the midst of its extensive business processes, thousands of UIs, hundreds of users and heterogeneous systems, forming an intricate structure. A complex corporate environment is often faced with the need to control these complex structures to bring more results and innovate as the elements of this structure naturally evolve. This has encouraged CIOs to push business and technology integration in order to make innovation real [42].

Adheres to principles of operations strategy: The framework is in accordance to the principle of alignment used in operations strategy that aligns the organization's ensemble of resources, processes and competencies with the competitive position that it seeks over time [133]. This is the case because our alignment approach creates a network of links that enables the configuration of operations towards improving: process performance, end-user interaction and the corporate mindset towards these changes, thus linking the operations strategy support: processes, resources and competencies.

Adheres to the theory of structuration: The framework is aligned with the theory of structuration [31], which states that it is neither only individuals nor only the organizational rules that shape organizational structure and changes. It is therefore wise to avoid extreme positions and consider reality as a mix of stakeholders and structure. For that reason, the framework holds two main characteristics addressing this theory. First, it is human-centered to enable stakeholders to apply their own knowledge when making decisions. Second, it is supported by a methodology and software platform to structure the routines they must follow, thus being both flexible and structured. Giddens expresses this by saying that ordinary day-to-day life involves a link between autonomy of bodily control within predictable routines [17].

Supports traceability: As the models are mapped, a network of links is created that enables the traceability, that is, "the ability to describe and follow the life of a requirement in both a forward and backward direction" [34]. The traceability is described by rules to maintain consistency between the models and followed by navigating in the network of links in any direction to analyze the impact of changes.

1.2.3 Research Scope

The research is focused on the topic of UI-Business alignment, which is a specific research topic that integrates IT-Business alignment and MD-UID.

The main research question is: *How do changes on business processes influence UIs (and vice-versa)?* The main proposition for this research is that business processes do influence UIs of enterprise systems because activities must be represented in UIs to support professionals in performing such activities. Therefore, when business processes are changed, the UIs that support the enactment of these processes must change accordingly. This is possible through the association of BP elements with UI components. The second proposition is that mapping business processes with UIs requires more than direct links, it requires a deeper concern on user interaction. Therefore, we use task models in order to put user interaction at a strategic point, avoiding only a business viewpoint. The third proposition is that the human perspective contributes to improve the user experience because end-users themselves can collaborate by giving ideas of improvements on UIs that directly influence the system usage, navigation, interaction, among other aspects that influence the overall user experience.

The overall research goal is to correlate business with UIs by specifying a framework composed of: 1) a methodology that defines the association of business process with UI models, and 2) a platform that supports model traceability and decision-making on alignment issues. This proposal is mainly aimed towards organizations that want their systems to address business processes in a way that as the processes are created, maintained, and evolved, so are its supporting UIs of their systems and vice-versa.

Framework for UI-Business Alignment: The framework identifies the links from the business process until the user interface to attain alignment and guarantee integration between business viewpoint and UI design. Our goal is aligned with the definition of Enterprise Architectures (EA), which are conceptual and methodological frameworks used to depict and align the organization's business processes, people and IT software [137].

Methodology for UI-Business Alignment: The methodology supports the application of the framework through the specification of actions to be performed by stakeholders involved in working with business process management and UI design.

Platform for UI-Business Alignment: The platform supports the traceability from the business process until the user interface with a set of functionalities that help stakeholders to create the network of links between these models that promotes the analysis of the impact of changes.

Boundaries

It is not in the scope of this research work to address the following issues:

The use of a model-driven approach supports this framework in terms of using models as the core artifacts that are managed throughout the UI life cycle. It does not mean that we provide a model-driven solution to generate the final product (e.g. concrete and final UIs). This kind of solution is provided in several works that adopt model-driven engineering approaches for UI design and development, such as [19] with an MDA approach to generate web user interfaces.

In the literature, the terms process and workflow are often confused or misused. To avoid any misunderstanding, we use the definition from the Workflow Management Coalition Specification (WfMC) [143] that states that workflow is "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."

This definition of workflow is clearly different from the definition of process from the Business Process Modeling Notation (BPMN), V1.2 [87], in which: "a process is any activity performed within a company or organization." Even though we address processes designed in a flow-chart format, we clearly distinguish business process from its automation as a workflow. To complement, one important aspect to point out is that "what distinguishes an arbitrary set of coordinated activities from a business process is the fact that the process must add value to some customer, whether internal or external to the organization" [110]. Therefore, we recognize that **Business Process** is more appropriate for this framework.

Target Audience

The target audience of this thesis is composed of, but not limited to:

- **Research communities**: The Human-Computer Interaction (HCI), Enterprise Engineering and Business Process Management (BPM) communities interested in IT and UI-Business Alignment;
- **Large organizations**: The context of large organizations, for example banks, industries, among others that adopt business process management and use enterprise systems for their routine corporate activities.
- **Educational Institutions**: Universities with an open curriculum that includes teachings on new solutions proposed by the research community with application in real scenarios and a sound scientific foundation.

1.3 Reading Map

This thesis is organized in the following chapters:

Chapter 1 presented the main motivations, intended goals, and the overall structure of the research work.

Chapter 2 presents and compares some existing works on IT-business alignment and MD-UID concerning predefined requirements for the integration of these domains.

Chapter 3 presents the UI-Business alignment framework in which models are at the core of the solution, how they are mapped and managed to provide alignment, and how professionals can understand and use the solution.

Chapter 4 presents the UI-Business alignment platform by explaining the taxonomy used to specify the rules that control the mappings between the models whenever changes are performed as a support for decision making.

Chapter 5 demonstrates the validity of the framework by presenting the theoretical validation, based on the comparison of our proposal with specific requirements; and the empirical validation, based on two case studies used as running examples throughout the work.

Chapter 6 concludes this work by presenting the major contributions and proposing future works.

Chapter 2

State of the Art

"Focus on process alone can lead to design by checklist.", Karel Vredenburg, User Experience Design Program Director, IBM Design.

This chapter presents the state of the art on aligning business processes with IT, including how UIs are handled in this context in order to evaluate their main characteristics. First, works on IT-Business Alignment are presented, followed by works on Model-Driven UI Development that are concerned with UIs of enterprise systems.

2.1 IT-Business Alignment

Recent works propose different types of technological support for traceability between business processes and the supporting software systems, such as [3] and [124]. As Business-driven IT management (BDIM) refers to a research area that focuses on making IT help the business [104], it is evident that the importance of maintaining business process-IT alignment has increased. The three-layer BDIM model to derive IT-business linkage mapping functions [79] are: business layer, which models business entities; business process layer, which consists of processes dependent on IT services; and IT service layer, which considers IT infrastructure, its services and behaviors of users of such services. These layers are linked to investigate cause-and-effect relationships. Similar to this model, we present other strategies to align business processes and enterprise systems.

2.1.1 Strategy, Business Process and Information Systems

A framework was defined to do the traceability between business strategy, business processes and IS using UML stereotypes [134]. 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 first level of the GPS diagram aims at describing the business strategy through goals. The second level is composed of the business processes that must satisfy one or more goals that produce, consume or refine resources, and that are located in one flow activity graph. The third layer is the information systems layer that supports business processes and is composed of system components.

Goal modeling is done based on the notions of Goals, Problems and Contradictions and on the fundamental concepts in the Balanced Scorecard [54] that distinguishes strategic goals (with long-term purposes) and operational goals (with short-term purposes that the company must achieve). Besides, each one can be classified in qualitative or quantitative.

Business process modeling uses the value chain model [95] to analyze business processes by highlighting primary activities (directly related to the organizational products and services) and support activities (the ones that support the core activities).

Information System (IS) Modeling uses a UML extension that they have developed, called Extended Component stereotype, which represents the functional building blocks of the system.

This framework can be observed from two views. The first one is the structure view that presents the three framework layers: 1) the goal diagram describes the goal structure of the company. 2) the process diagram with the dependencies between the business processes. 3) the IS diagram with the system structure and dependencies between components. The GPS diagram puts it all together and depicts the dependencies between goal, process and IS. The second view is the behavior view, which shows the dynamics of the processes and resources with state diagrams, and the interactions between components with UML sequence and collaboration diagrams.

The GPS diagram is used to analyze the dependencies between strategy, business processes and IS. For example, a new business strategy to start selling on-line leads to setting new goals (e.g. have interface adapted to costumers), which requires changes in supporting processes (e.g. arrange product displays) to embrace these new goals. Consequently, to support the new business needs, certain system components (e.g. component for on-line shelf management) must be created or changed.

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. This happens because the GPS diagram does not show the details of the IS diagram. 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 sub-components or only in one of these sub-components?

This work has been extended at the IS level to facilitate system integration [135]. However, the association of the IS level with the business level is still done through the higher level of the IS definition.

2.1.2 The CEO Framework

The Center of Organizational Engineering (CEO) Framework represents an Enterprise Architecture (EA) that aligns the organization's strategy with its business processes, people and Information Technology (IT) software and hardware [110, 137]. This framework comprises five architectural components used to model the organization: Organizational Architecture, Business Architecture, Information Architecture, Application Architecture and Technological Architecture.

The *organizational architecture* specifies the organization in terms of enterprise *mission*, *vision*, *strategy*, *organizational goals* and *organizational units*. It is not related with the specific business conducted by the organization neither with the mechanisms to accomplish the creation of organizational value.

The *business architecture* represents how business strategies are implemented through *business goals* and how *business processes* are defined. A business process is the core concept within this architecture and it represents a set of coordinated, value-adding *activities* that operate over input entities and produce output entities.

The *information architecture* describes the key information elements needed to run the organization's processes and operations as described in the business architecture. It provides a high-level logical representation that is independent of system and technology by describing the business roles necessary to operate activities, which include *actor* that represents a person, machine or device, or an information system; a *resource* that represents input or output for the activity; and an *observable state* that is a specific resource used to observe the status of an activity.

The *application architecture* defines the applications required to support business requirements specified in the business architecture and to manage organization's entities analyzed in the information architecture. These applications are described using *IS blocks* that depict application building blocks and then further defined as a collection of *services*, mechanisms and operations to handle organizations' information.

The *technological architecture* represents the technologies necessary for the implementation and the infrastructure needed for the deployment of the business process support systems. This architecture uses *IT blocks* to represent the infrastructure, application platform and technological components that implement a set of IS blocks. In addition, *IT services* are used to describe a set of technological services provided by the specific application platforms.

They have assessed the alignment between the architecture views and its elements by defining a set of rules for the dependencies between three architectures [110]. First, rules to assess the alignment between the business and information architectures in order to enable business people to have the information they need to run the business. Second, rules to assess the alignment between the business and application architectures in order to guarantee efficiency to conduct and fulfill the business through automated business services. Third, rules to assess the alignment between the information and application architectures in order to ensure that applications have the right data for processing. In addition, there is a set of integrity rules that deal with the relationships between pairs of specific architecture components, for instance, between activity from information architecture and strategy from organizational architecture.

The CEO Framework has evolved with the inclusion of a human perspective in order to capture how human agents establish relationships with activities, resources and concepts of the other perspectives [137]. A set of changes have been made on the CEO Framework, for instance, the concept of service is no longer part of the technological layer because human agents can also provide them; and culture has been included in the organizational perspective. For the human perspective, it distinguishes between *individual subject* and *collective subjects*. Individual subjects relate to organizational activities and resources by specific *personal contexts* performing *actions* and collective subjects by specific *interaction contexts* and its created *interactions* between individuals. Contexts are situations that emerge from interactions among resources and activity-related agents. Since contexts are situated in time, it is possible to know what individuals or collective subjects are doing at particular moments in the project. For example, it is possible to know for a specific individual, which are her/his active contexts in a certain period; which activities she/he has performed during a specific period; which resources she/he used for each context; etc. The CEO Framework has been structured as an EA aimed at having a significant impact on the awareness of organizational members at all levels [69].

2.1.3 Artifact-Centered Operational Modeling

Artifact-Centered Operational Modeling (ACOM) is an approach to capture business operational knowledge that is distinguished from other approaches by identifying one information structure, the artifact, which travels from end to end in a process. It can be understood as an alternative to more familiar approaches, such as BPMN. The goal of an operational model using the ACOM approach is to "represent business operations at a granularity that is sufficient for validating and managing progress toward goals and to clarify the dependencies between goals." [13]

They have defined a framework for IT-enabled business transformation using ACOM as the central foundation. Figure 2.1 depicts the framework that is divided in four abstraction layers, each one dedicated to model the enterprise behavior from a different perspective. The ACOM is used in this framework to provide a direct link between the strategic and development layers.

The *strategy model* lists the business objectives that can be specified in terms of the Balanced Scorecard [54], similarly to the GPS diagram (explained in sub-section 2.1.1). The *operation model* specifies the structural and behavioral aspects of business using the ACOM approach. The *solution composition model* is a platform-independent design of the solution to be implemented. The *implementation model* is a platform-specific version of the solution composition model that specifies systems by using a service-oriented architecture (SOA) platform.

The ACOM approach is exemplified by a supplier and customer transaction processing in a health insurance company. This project aimed to redesign their provider data management processes in order to keep the database of physicians in its provider networks up-to-date and allow faster processing of claims. For this purpose, they decided to use operational modeling to help define their processes in a standard manner.

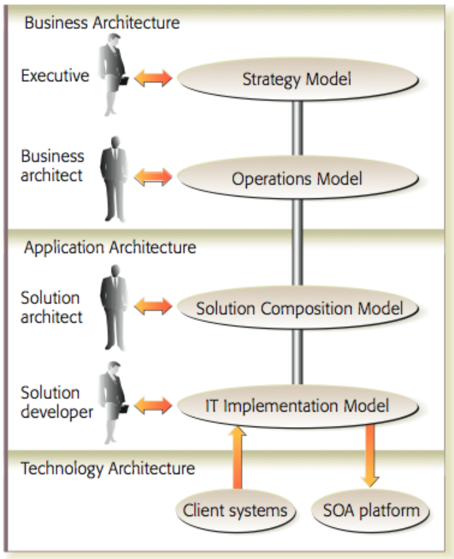

Figure 2.1: The Framework for business transformation (from [13])

The ACOM approach was used by company representatives and generated operational models for the processing of the update provider artifact. They delivered standardized processes for the different offices throughout the geographic service areas of the company, by considering the artifact-centric view and removing focus from detailed activities.

The ACOM's focus on operational goals and how they satisfy business goals provides a traceable link to strategy. This results in the ability to map organizational goals with effective IT implementations.

2.1.4 Alignment Strategy

This strategy detects misalignment when a change is executed by applying two different techniques [7]. 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.

2.1.5 MetaMIS approach

There are also approaches that use theories to understand organizations and ISs to help with alignment and communication. The approach applied by Rosenkranz & Holten [101] 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.

Aiming at solving communication issues, one of their examples is related to appointing employees with IT background to work in the business unit. However, in our analysis, it represents a short-term solution that is 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.

2.1.6 End-User Business Process Modeling

The End-User Business Process Modeling [119] is an approach in which business users are active participants in business process modeling using the Collaborative Task Manager (CTM), which provides end-user development capabilities. In their approach, formal process flows are created based on tasks users are performing that are tracked by web services on a central server. CTM tracks personal to-do lists of different process participants and integrates them in Task Delegation Graphs (TDGs), which provide a workflow-like overview of collaborative tasks. These tasks are considered as input for the definition of Task Patterns, considered as reusable hierarchical task structures. CTM formalizes the process using the JBoss Business Process Management (jBPM) solution. The deployed jBPM flows can be executed using the jBPM engine. This approach allows the formal workflow to be deviated so users can do ad-hoc tasks, which facilitates redefining the workflow.

Their solution is useful for situations where the organization does not have their processes formally modeled, neither enterprise systems that automate their processes and need to create a workflow system to guarantee that the sequence of activities is done by the respective participants, thus saving the effort to distribute tasks by e-mail.

Similarly, Zacarias et al. [138, 139] enrich enterprise modeling with a methodological approach to capture and model work practices by acknowledging the complexity and dynamic behavior of people at work. Their framework defines an architecture and ontology of organizational agents that includes five essential concepts: *resources*, entities required to perform activities; *agents*, resources with acting, coordination and adaptation capabilities; *activities*, abstractions of what agents do; *roles*, the observable behavior of agents; and *context*, determines the specific role played by agents. Actions of agents are captured to identify personal contexts, which are linked together to find inter-personal context-based networks. From there, action flows are identified, forming a business process of way activities are actually performed, using the Microsoft EM Clustering algorithm. This resulting process can be used to check the alignment of the actual execution with the current models and correct deviations or to detect innovations and update the current models.

2.1.7 Model-driven development approaches

The Model Information Flow used at SAP [12] follows a model-driven approach that uses models and transformations between them to define, deploy and manage SAP applications. It is noted that the models in the chain, after the business process model is customized for a project, are mostly related to IT infrastructure, for instance models to describe internal structure of the software, demand for resources by components, resource infrastructure configurations, etc.

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 [132], which discovers useful trace information from model relations. In the work of [102] on the traceability in model-driven development of business applications, they link artifacts from the development process that range from software requirements, test cases, design objects and code fragments. It does not mention any artifact for UI and the artifacts are not necessarily models, but sometimes text documents, for instance. On the other hand, it focuses on how software development techniques aid in traceability, in their case, using aspect-orientation.

2.2 Model-Driven UI Development

Models are abstractions over a (part of a) software aiming at simplifying the specification (design, development and other stages) of complex software [36]. A model-driven approach "provides a means for using models to direct the course of understanding, design, construction, deployment, operation, maintenance and modification" [88].

Model-Driven UID (MD-UID) provides a more systematic approach to UI construction, by relating the interface developed to the underlying abstract models and it comes as a solution to improve efficiency by reusing user interface models [10]. As a result, it enables the possibility to reduce development efforts and to change the mindset from technology towards user interaction. MD-UID also enables organizations to develop IS that provide the necessary support for current user interaction challenges, by considering the user profile, developing UI for multiple devices, etc.

Many efforts have focused on model-driven design of UIs, creating the UI from user tasks [15, 19, 25, 91, 145] with a focus on automatic generation of UIs. Paterno & Santoro [92] specify the relationships between task model and abstract UIs, and between abstract UIs and its implementation. Vanderdonckt [131] defines a mapping model that contains the mappings between the models and elements of models to generate UIs. There are Model-Driven User Interface Development Environments that map concepts from different models to guide in designing UIs, such as TEALLACH [35], TERESA [78], and UI Pilot [98]. However, the focus of the investigation in this domain is on the current works that address model-driven and user-centered design and development of enterprise systems using business processes models.

2.2.1 User-Centered Design and BP modeling

One of the few researches on the enhancement of the UI design practice with business process modeling is the experience reported by the IBM T.J. Watson Research Center [123]. 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, 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.

In a further work, they detail their work [124] by explaining their approach to leverage business process models as a starting point of the UI design life cycle. 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. 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 to conceive the task model.

The model-driven UI environment called Model-Driven Human Interaction (MDHI) enables designers to connect UI design with business domain model elements [124]. There are four views in the human interaction perspective: (i) "User role - Artifacts" - lists the artifacts needed by each user role; (ii) "Artifacts - User role" - for each artifact, it lists the user roles that can access it; (iii) "User role - Task" - lists the business tasks needed by each user role; (iv) "User role - Design" - lists the UI design pages defined for each user role. These different views aids to design UIs considering tasks performed, artifacts used and the users involved.

In [125], they use data group concepts to link design elements to business process models. A data group is a subset of data elements that appear together on a UI. In [126], they present an analysis of their proposed Business Model and UI Modeling Methodology, which resulted in faster iterations with more results within the same amount of time in comparison with a traditional development method.

2.2.2 Diamodl

In one of the most recent works of Traetteberg [130] on model-based UI design, he takes an industrial perspective and advocates to adapt BP modeling to include aspects of task modeling, arguing that it is not realistic to introduce another modeling notation in contexts where UI design is not as valued as system architecture. Therefore, the proposed method uses BPMN for both BP and task modeling.

The approach relates BP models with task models since both of them capture the behavior of people. The BP execution is related to the dialog model since both describe software behavior. The BP execution deployed in the server-side is related to the client-side UI runtime, as depicted in Figure 2.2.

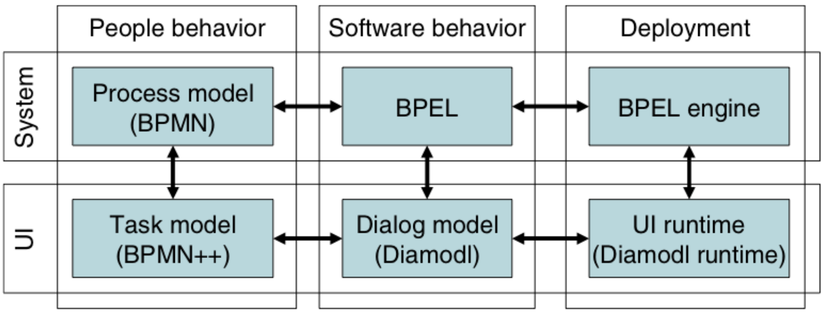

Figure 2.2: Relationship between system and user interface domains (from [130])

The relationship between system and UI domains in Figure 2.2 depicts BPMN inside the system domain. According to a renowned definition by Davenport [20], business process is a structured, measured set of activities designed to produce value for customers. From this definition, we can defend the fact that BP captures the behavior of people, but it is independent of the system.

The proposed modeling method starts with BP modeling using BPMN combined with domain modeling using class diagrams. Since this version of the BP model is considered as system centric, the second step is to separate users' tasks from the system in a refactored process model using BPMN.

This refactored process model is similar to a task model in the fact that it includes a new swimlane for the user role to make explicit what the user does and how it interacts with the other roles and processes. In addition, it includes pre and post conditions to emphasize how domain data is operated, which may result in a refined domain model. The third step is to create the dialog model using the dialog modeling language Diamodl. The forth step is to generate the graphical UI prototype from the dialog model. The last step is the deployment of the system.

In the refactored business process, where users' tasks are separated from system tasks, there is only one user role. However, we argue that users play different roles in processes. For instance, the customer can interact with the system to send a request or receive a review through the system; and the expert could also interact with the system to make an expert review or to return a review, thus the both the customer and expert may play user roles with different purposes.

2.2.3 Hybrid approaches

In the work of Stolze et al. [120], they call attention 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. In this work, the main goal is to support occasional developers to solve business problems by developing workflow applications in an agile way. For that, custom UI components can be reused to modify template pages and describe business logic in a XML-based workflow task model. In addition, custom UI components use information from the task model to specify which operations certain roles are allowed to perform on them, such as browse, read, edit, add, and delete.

Aligned with the idea that information from business processes are not enough for UI design, the work of [94] advocates for a hybrid approach for modeling combining task models and process models.

Similarly, [58] proposes a solution to use both workflow and task models to design role-centered UIs. It associates tasks in BPMN model with the highest level of task models. However, there is a limitation in this approach because even though the company in their case study decomposed their business process models in tasks; not all business process models have only one level of decomposition. It is possible to have processes composed of at least three different levels: sub-processes, activities, and tasks. When modeling task models, they adopt a hierarchical sequence-oriented style, but fixed sequence is not as broad as the variety of possibilities provided by different operators that enable a richer user interaction.

2.3 Summary

2.3.1 Shortcomings

From the detailed analysis of related works on IT-Business Alignment and Model-Driven UI Development, we have identified a list of shortcomings that must be addressed in this domain:

1. **No vision for user interfaces**: The impact of focusing 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 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.

2. **Vision for user interfaces, but not for user interaction**: Some recent works in this area propose aligning business processes with UIs with direct links between activities in processes to elements on the UIs. However, such strategies leave aside the consideration of the user interaction, which is much richer than what is specified in business processes. "Users interact in an environment that includes physical artifacts and networks of diverse computers and whose physical, social, organizational and cultural characteristics affect their actions" [11]. In this context, when designing interaction, user interfaces are the means, not the end.

3. **Absence of generic structure**: The decomposition of a business process in layers may vary depending on the organization and that is a crucial aspect, which makes one proposal applicable or not on certain organizations. However, there are some proposals that have been defined based on specific case studies, but some constraints in a proposal may lead to a limited applicability.

4. **Not enough concern with people**: Works that focus much more on how the artifacts are linked, handled and maintained than on the

people who actually conceptualize and those who use the enterprise systems, fail to consider that it is actually these people the ones that are key to making the organization work.

5. **Scarcity in tool support for traceability**: Most works propose a framework with principles to be applied, but it is not often that such works present a tool to help professionals in making decisions when changes are made to business processes or in other artifacts that are related to it. Other works have tools to support the development life cycle, but no support for traceability is offered. This fact brings less efficiency when searching for information spread in different sources.

2.3.2 Requirements

The aimed result is to provide a framework that promotes the following benefits for UI-Business alignment, which we consider as requirements to be fulfilled in our research:

1. **User-centered**: Support user interaction by designing systems focused on users and their tasks.

2. **Adaptable**: The ability to be adapted and applied in different organizations, independent of their adopted business process structure, UI design or IT solution.

3. **Human-centered**: Provide a framework that considers the viewpoints of the business process performers as active agents in the proposed methodology.

4. **Traceable**: Support organizations in maintaining their systems aligned with the models, where it is possible to identify the impact that changes on a particular model have on other models and on the system as a whole.

5. **Software Support**: Supply a software platform that integrates the different models in one source in order to provide traceability.

Table 2.1 depicts the association of the requirements with the shortcomings identified in existing works as explained in section 1.1 Research Motivation.

Table 2.1: Relating shortcomings with requirements

Shortcomings	**Requirements**
No vision for user interaction Vision for user interfaces, but not for user interaction	User-Centered
Absence of generic structure	Adaptable
Not enough concern with people	Human-Centered
Scarcity in tool support for traceability	Traceable Software Support

The list of shortcomings and requirements cover three major aspects of this research: conceptual, methodological and automation.

The User-Centered Requirement addresses: the conceptual aspect by adding the task model in the meta-model of an alignment approach; the methodological aspect through the analysis of tasks derived from business processes; and the automation aspect by transforming business process models into task models and vice-versa.

The Adaptability requirement addresses: the conceptual aspect by creating meta-models that accept different structures of models; the methodological aspect by letting stakeholders free to model using the techniques they are most familiar with; and the automation aspect by providing rules that manage the models and accepting that the business model can be imported from different tools.

The Human-Centered requirement addresses: the conceptual aspect by adding the end-user as an entity in the meta-model; the methodological aspect by opening a channel where end-users can give their opinion about the UIs; and the automation aspect by enabling stakeholders to see who are the end-users impacted from changes on the models.

The Traceability requirement addresses: the conceptual aspect by creating the connection between the main meta-models; the methodological aspect by allowing stakeholders to create a network of links between the models; and the automation aspect by presenting an impact analysis with a detailed report of the proposed changes on impacted models.

The Software support requirement addresses the automation aspect by providing a platform to support applying the methodology.

2.3.3 Comparison of the approaches

As we have demonstrated with the previously mentioned research works, many contributions support the alignment of business processes and IS. Results from investigations [23] 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.

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.

We have compared these approaches according to the requirements, defined previously. Other requirements could have been considered, such as adherence to the principles of agility. However, this and other prospective requirements are currently out of the scope of this comparative analysis because they currently considered as aimed goals for an extension of this solution as future work.

Table 2.2: Global comparison of approaches for all requirements

	GPS	CEO	ACOM	CTM	MDHI	Diamodl
User-centered				x	x	x
Adaptable		x				
Human-centered		x		x		
Traceable	x	x	x	x	x	x
Software Support			x	x	x	x

Table 2.2 depicts how the investigated approaches were evaluated considering the requirements, which represents an overview of this analysis. More specifically, we explain how these approaches adhere or not to these requirements.

User-Centered

Table 2.3 depicts how each approach was evaluated according to the User-Centered requirement, which is explained as follows.

Table 2.3: Comparison of approaches for the User-Centered requirement

Approach	Characteristic
GPS	*not user-centered* because the focus is on information system architecture, considering information system and technological services, IT infrastructure, application platform and technological/software components.
CEO	*not user-centered* because the framework has no concept related to user interfaces nor tasks for user interaction. Activities performed by people using a system are represented as automated services that are directly associated with technological components.
ACOM	*not user-centered* because it is focused on system implementations and how IT components (i.e. service components) are linked to operational tasks and artifacts.
CTM	*user-centered* because formal process flows are created based on tasks users are performing tracked from personal to-do lists of different process participants.
MDHI	*user-centered* because it adopts a dual model approach that uses business process models and human interaction modeling to produce systems with UI and business logic connected.
Diamodl	*user-centered* because it adopts the concepts of task analysis (not specifically the task model as we usually see in MD-UID).

Adaptable

Table 2.4 depicts how each approach was evaluated according to the Adaptability requirement, which is explained as follows.

Table 2.4: Comparison of approaches for the Adaptability requirement

Approach	Characteristic
GPS	*not adaptable* because its components are fixed to three types, namely business goals, business processes and information systems, represented in its main diagram.
CEO	*adaptable* with the possibility to redefine the rules that associate enterprise concepts.
ACOM	*not adaptable* to any organizational context because it assumes the adoption of service-oriented architecture for IT solutions.
CTM	*not adaptable* to any organizational context because it assumes the organization does not have their processes formally modeled, neither enterprise systems that automate their processes and need to create enterprise systems from scratch from the use of a Microsoft Outlook add-in to extract users' to-do lists.
MDHI	*not adaptable* to any organizational context because it assumes the organization already adopts or intends to adopt artifact-oriented business process modeling.
Diamodl	*not adaptable* to any organizational context because it assumes the adoption of service-oriented architecture for IT solutions.

Human-Centered

Table 2.5 depicts how each approach was evaluated according to the Human-Centered requirement, which is explained as follows.

Table 2.5: Comparison of approaches for the Human-Centered requirement

Approach	Characteristic
GPS	*not human-centered* because this work has focused on goals, processes and systems that represented one of the first versions of the CEO framework, without the human perspective.
CEO	*human-centered* since it includes a human perspective by relating actions of individuals and interactions of collective subjects with organizational activities and resources.
ACOM	*not human-centered* because it adopts a model-driven business transformation methodology with automatic generation from business to technology.
CTM	*human-centered* because business users are active participants in business process modeling. However, they participate as creators of process models, which do not necessarily get the most useful information from them using their existing know-how since it requires them to learn certain techniques in order to be able to give any suggestion.
MDHI	*not human-centered* because the key people who participate in this framework are business analysts, data modeler, user experience designer and UI modeler.
Diamodl	*not human-centered* because the proposed method is oriented to UI and IT professionals.

Traceable

Table 2.6 depicts how each approach was evaluated according to the Traceability requirement, which is explained as follows.

Table 2.6: Comparison of approaches for the Traceability requirement

Approach	Characteristic
GPS	*traceable* because the framework describes, links and traces organizational concepts at multiple levels of detail using three separate areas of concerns: strategy and goals, business processes and information systems that are represented in the GPS diagram.
CEO	*traceable* because it aligns the organization's strategy with its business processes, people and IT.
ACOM	*traceable* because it uses a model-driven business transformation (MDBT) approach that maps organizational goals with IT implementations. But they only consider that changes are driven by the business stakeholder and performed first in the operational model.
CTM	*traceable* because local tasks from personal to-do lists are re-structured in task patterns that can be enacted, updated and evolved and such evolution is traced through task instance-based ancestor/descendant relationships.
MDHI	*traceable* because it makes the link between BP and UI through the business information model.
Diamodl	*traceable* because it makes the link between BP and UI through the dialog and domain model.

Software Support

Table 2.7 depicts how each approach was evaluated according to the Software Support requirement, which is explained as follows.

Table 2.7: Comparison of approaches for the Software Support requirement

Approach	Characteristic
GPS	*not supported by tool* because it does not provide any support for traceability by using a UML modeling tool, since the framework is presented as an extension to the UML as a standard UML profile.
CEO	*not supported by tool* because there is not an automated support for traceability between processes and systems; even though there is automated support for context-related operations (e.g. context-based analysis, context visualization) and support to relate strategies with tasks, and showing resource flows among individuals.
ACOM	*supported by tool* since they have developed a MDBT toolkit to support modeling and semi-automatic model transformations of operational models into running applications.
CTM	*supported by tool* since it is implemented and validated through a process-enhanced groupware system - the Collaborative Task Manager (CTM).
MDHI	*supported by tool* with a model-driven UI environment called Model-Driven Human Interaction (MDHI) that enables connecting UI design with business processes.
Diamodl	*supported by tool* through a set of connected tools, including tools for BP modeling, dialog modeling, domain modeling and BP execution.

2.4 Conclusion

This chapter analyzed a set of works on IT-Business Alignment and on Model-Driven UI Development that are concerned with business process modeling and UIs of enterprise systems. This analysis enabled identifying the major shortcomings of the existing approaches, which are useful source of information to specify the requirements of the framework we propose.

In our proposal, 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. Using the framework, it is possible to 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 large organizations, which aims at aligning their business processes with IT, but is also concerned with the user experience, as presented in the next chapter.

Chapter 3

UI-Business Alignment Framework

"Design of the total user experience affects sales, service cost, productivity, customer loyalty - most aspects of doing business.", Karel Vredenburg, User Experience Design Program Director, IBM Design.

This chapter presents the conceptual and methodological framework for UI-Business alignment with the definition of the concepts used to present the solution and the full description of the methodology. The framework is presented with characteristics that fulfill the requirements presented in Chapter 2 and is supported by a platform, subject of the next chapter.

3.1 Introduction

The UI-Business alignment framework is composed of a methodology and a platform. The methodology is composed of core actions and the platform supports the methodology and it can be extended with new functionality by adding plug-ins to supply additional functionality to align BP with UIs, as depicted in Figure 3.1.

The UI-Business Alignment framework addresses the following requirements:

User-Centered: The task model is recognized as the best model to precisely represent how users interact with systems and that piece of information is primordial to be linked with UI components and monitor users' interaction aiming at optimizing the processes that they execute and consequently

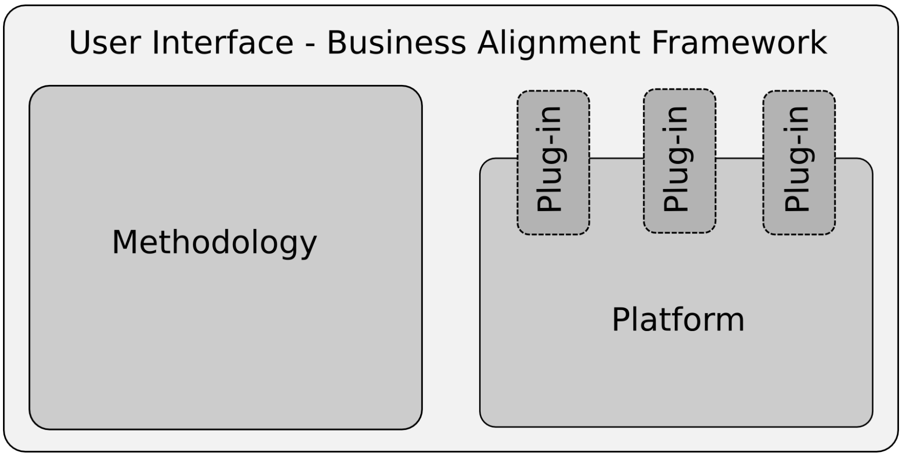

Figure 3.1: UI-Business Alignment Framework

the user experience. Without the task model, linking processes directly with UIs is not optimal because processes with high-level description do not express how activities are performed; and processes with detailed description still do not express tasks that users perform when interacting with enterprise systems. The detailed specification of interactive tasks represents a strong source for analytical data concerning process optimization.

Adaptable: This framework is adaptable to changes in the organizational context, it accepts that the way processes are modeled and structured in different layers may change and that the philosophy of user interaction may be evolved, thus leading to the need to update how these models are managed. For this purpose, we have selected an expert system approach based on production rules, where these rules specify how these models are managed to provide more flexibility. When there are changes in the structure of the models that impact the form of managing them, the rules can be directly updated, without the need for maintenance of the traceability tool.

Human-Centered: One of the core aspects of our framework is considering users as active agents who open a new channel for business improvement by informing any sort of issues they face with systems. The identified issues may lead to changes on the processes, on the UIs or on both. This is in accordance with the theory of structuration [31], which considers the organizational structure, but positions structure within human practice, recognizing that the agents have and apply their knowledge when acting, which may then change the organizational structure.

Traceable: The traceability from the business process until the UI aids in identifying the impact of process changes on the user interaction and in the proposal of changes in the processes when the user interaction is improved. The main elements of the models serve as foundation for the definition of rules that make it possible to trace the models in forward, middle-out and backward approaches.

Software Support: A software platform for the methodology aims to enhance insight for decision makers by: (i) giving complementary control over corporate operations, in addition to what is already offered by the market, (ii) widening their viewpoint beyond common IT-business perspectives, (iii) simulating impact of changes and (iv) enabling innovations that can influence both process improvement and user experience. This emerging software is envisioned to be harmonized with existing tools adopted in organizations, such as those used for business process modeling, orchestration of services, UI design, system development, etc.

The UI-Business Alignment framework is presented with two perspectives: conceptual (which presents the models used and how the models are mapped) and methodological (which specifies how to apply the methodology in an organizational scenario), specified in the following sections.

3.2 Conceptual Structure of the Framework

Adopting a model-driven perspective, the main concepts used in the framework are models, founded on an established framework and language from the domain of MD-UID, explained through a theoretical foundation and meta models.

3.2.1 Theoretical Foundation

The framework follows a model-driven approach to align BP models with UI models. At the center of the theoretical foundation is the Cameleon Reference Framework [15] that supports model mapping, UsiXML [64] that supports the formal definition of UIs and Business Process Modeling Notation (BPMN) [87] that provides the standard for business process modeling.

The Cameleon Reference Framework [15] structures a set of models in different levels of abstraction that provide a support for the current user interaction challenges, considering different contexts of use (comprising user, environment and platform). The Cameleon Framework is composed of four levels of abstractions (see Figure 3.2): tasks and conceptual models (e.g. task model, domain model, context of use model), Abstract UI (AUI), Concrete

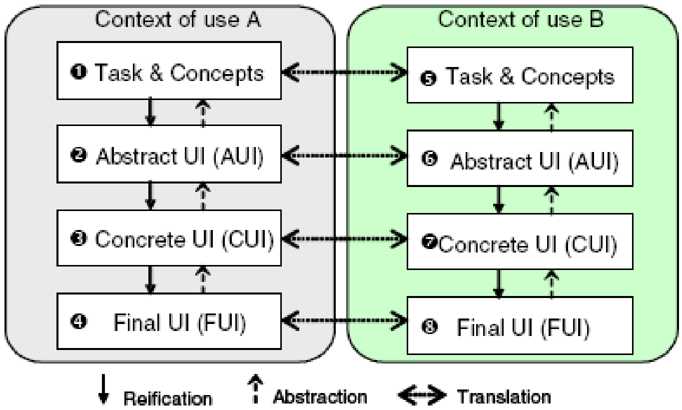

Figure 3.2: Cameleon Reference Framework (from [15])

UI (CUI), and Final UI (FUI). These levels of abstraction are aligned with the model-driven approach, which aims to reduce both the amount of developer effort and the complexity of the models used [36].

The models in the Cameleon Framework have specific characteristics to contribute to MD-UID: *Task models* describe the various users' tasks to be carried out while interacting with the interactive system under development. *Domain models* specify the concepts required by users' tasks. *Context of use models* specify characteristics of users, of platforms that will be used, and of physical environments where users will be located when interacting with the system. *AUI models* represent user interface components independent of any modality (e.g. graphical, vocal). *CUI models* provide a specification of user interface components independent of any programming language. *FUI* is the operational UI of the developed system.

Some of the existing model-driven approaches for UI design use a User Interface Description Language (UIDL) to specify models and exchange these specifications between IT professionals; which can be exemplified by UsiXML [64], UIML [3], XIML [97], among other UIDLs, which are compared with a significant selection of XML-compliant user interface description languages in [109].

Based on the Cameleon Reference Framework, UsiXML [64] is a XML-compliant markup language to describe UIs for multiple contexts of use, such as graphical, auditory and vocal user interfaces, virtual reality, and multimodal user interfaces. It follows a multi-path model-driven approach, composed of steps for multi-context interactive applications in a simplified manner. UsiXML main models are task model, domain model, context of use model, AUI model, and CUI model.

The Cameleon Framework and the UsiXML language provide a set of models that are mapped with each other and that support transformation of one model into another, which supports model traceability. Transformations can be done mainly in three different directions: (i) reification, from a high-level model into a lower-level model (to develop the system); (ii) abstraction, from a low-level model into a higher-level model (to create conceptual models of existing systems); and (iii) translation, adapting a model to the constraints imposed by a new context of use (to adapt the same model for different platforms; e.g. from CUI for desktop into CUI for web).

Choice of UI standard: The UI-Business Alignment Framework is founded on the Cameleon Reference Framework and, more precisely on UsiXML for the definition of the task and UI meta-models because it:

- provides the necessary support to represent models in a formal and structured manner;
- supports the flexibility necessary for model-driven UI development;
- represents a reliable standard that is subject to the W3C standardization process;
- is compliant with Model-Driven Architecture (MDA), which prescribes certain kinds of models to be used, how those models may be prepared and the relationships of the different kinds of models [88];
- is in accordance with UIDL requirements: processability, expressiveness, standardization, human readability, and concision; reasons that are exploited in [131];
- covers a wide range of UI interaction modalities (e.g. vocal, 3D, multimodal), which creates more possibilities of integration of this framework with different types of systems;
- follows a transformational approach that allows flexibility in the definition of traceability links;
- has been in constant evolution, being the subject of exploitation by several companies and universities under the umbrella of a long-term European project.

These standards are used to define how a set of these models are mapped with each other. The main difference in this research is the addition of an external model, in which there is an interest on business process modeling.

Choice of BP standard: Business Process Modeling is based on BPMN, which is a standard that provides diagrams to be used by all business users, from business analysts who design business processes, to technical developers who implement the technology that performs these processes and to the business people who manage these processes [87]. BPMN has been selected as a BP notation for this framework because it:

- is a worldwide OMG recognized standard;
- is readily understandable by all business users with its standard visualization mechanisms, following the tradition of flowcharting notations for readability;
- handles advanced business process concepts (e.g. choreographies, compensation);
- consolidates the best ideas from divergent notations (i.e. UML activity diagram, IDEF, etc.);
- covers a wide range of usage by being mappable with XML languages for the execution of business processes (e.g. BPEL4WS); among other benefits that make it the next generation notation that combines readability, flexibility, and expandability;
- is implemented in several proprietary or open source tools.

Options of BP standards: There are other business process notations that could be used with the UI-Business Alignment Framework. The main requirement to apply a business process notation in this framework is that it is able to decompose activities in smaller parts, such as a sub-process in BPMN that can be decomposed in as many levels as needed to specify a complex activity. This framework accepts different approaches to BP modeling, such as the ones described as follows.

In the case of BP notations that are dependent of the technology used to build enterprise systems, such as ebXML [84] that is a business process model specification schema for e-business software, it represents a notation that is specific and not complete enough to be applied in a technology-independent context. However, it is still applicable with our framework because of its decomposable activities. It is composed of binary business collaborations with activities that execute other collaborations, which is the key to recursive compositions and re-use of collaborations.

BPMN adopts an activity-centric process modeling, which consists of business activities and connectors that describe the sequence of these activities [59]; while ArtiFact^{TM}-based Business Operations Model [83] is a type of information-centric BP modeling, which has its process scope as a set of connected business entities, that are information entities associated to a data model and behavior model [59].

Aiming towards a flexible way for actual work-task support, a new approach was introduced, which is termed subject-oriented business process management [29], as it focuses on the communication among process participants while they accomplish their work-task; but it does not represent composable activities.

To help stakeholders intuitively understand, create and change business processes, a language-oriented BP modeling [81] defines simple constructs that can be embedded in others, e.g. activity composed of steps.

Petri Nets is a modeling language used to describe the world of information processing. It is composed of local states, local actions and their mutual relationships [99]. An extension of Petri nets is Object Petri Nets [61] that supports multiple levels of actions in the net, where a Petri Net can contain Petri nets as its tokens inducing a hierarchy of nested Petri Nets.

The XML Process Definition Language (XPDL) is a common format to interchange BP definitions between different tools to analyze, model, describe and document BP [144]. It is able to represent all aspects of a BPMN diagram and it contains the concept of activity that can be represented as atomic tasks and as a sub-process, which contains its own definition of activities, thus reinforcing the hierarchical nature of BP.

There are also standards that are related to business process, such as Web Service Business Process Execution Language (WS-BPEL) [85], but they are considered out of the scope because we focus on standards that specify BP, not on standards that execute BP.

3.2.2 Meta Models

The main models used in the UI-Business Alignment Framework are: Business Process Model, Task Model, User Interface Model and other core entities related with this domain, as depicted in Figure 3.3.

Figure 3.3 depicts an overview of the connection between the four packages: Core, Process, Task and UI. The meta model of the UI-Business Alignment Framework is founded on the state-of-the-art on business process modeling and model-driven user interfaces, using BPMN, Cameleon Reference Framework and UsiXML. Each one of these standards is complete to achieve their aim to formally specify a domain. However, the specification of this meta model is delimited to contain only the elements that are primordial to be linked with the other model elements with the primary goal of UI-Business Alignment. The connection between these packages is explained in the following sub-sections, as each concept is described.

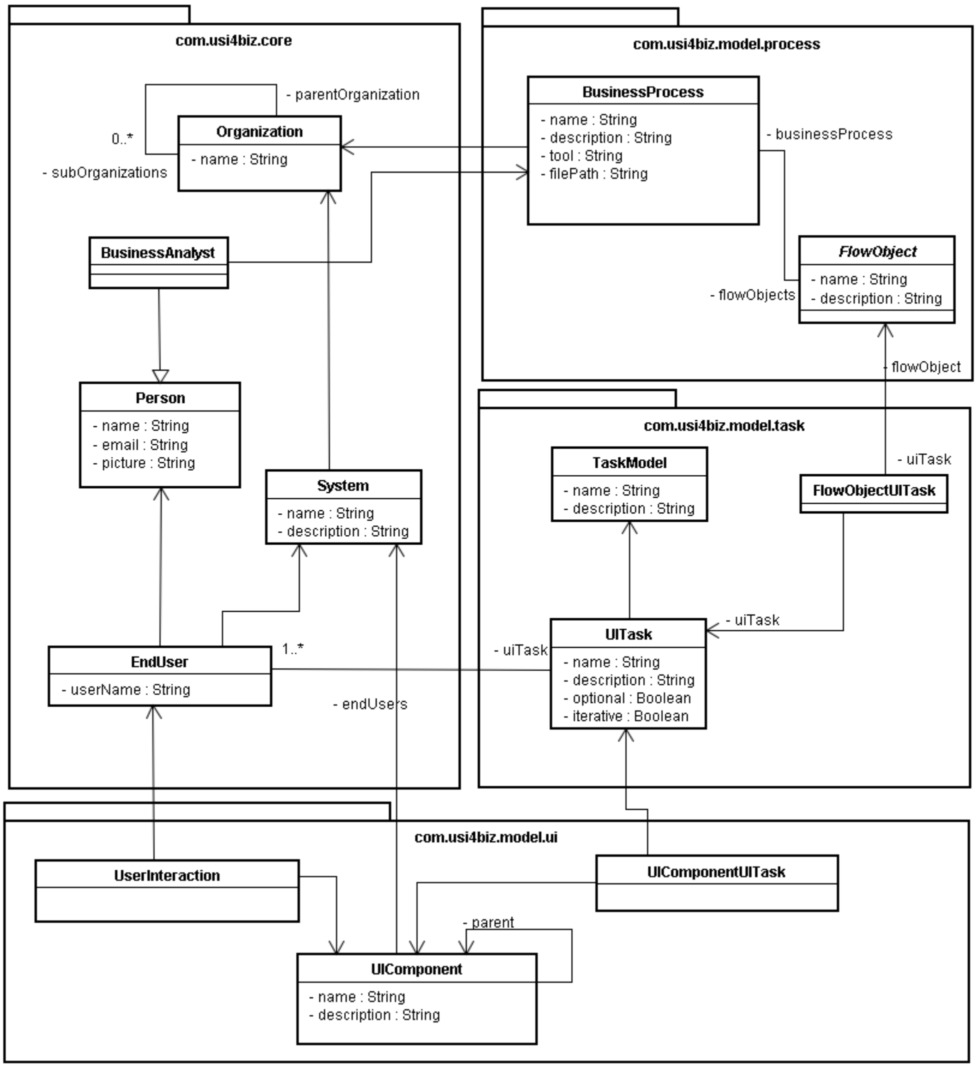

Figure 3.3: Overview of the Meta Model of the Framework

Business Process

BPMN [87] is useful to represent the sequence of work and the decomposition of the organizational complexity, as well as most business process notations. According to BPMN [87], a *business process* is defined as a set of activities that are performed within an organization or across organizations. The elements that define the behavior of a business process are called *Flow Objects*, which can be of three types: activities, events and gateways.

An *activity* is characterized as any work performed within a business process. The types of activities that can be part of a business process are: process, sub-process and task. A *process* has a hierarchical nature because it can be defined at a high level for the entire organization until lower levels of details for specific professionals within this organization. A process is depicted as a set of activities organized in a flow and it can be sub-divided in sub-processes and tasks. A *sub-process* is a compound activity in the sense that it is defined as a flow of activities, it can be as simple as a set of activities or as complex as a call for another process further decomposed. A *task* is an atomic activity, generally performed by and end-user and/or an application. A business process task is called a *BP task* hereafter in order to avoid any confusion with tasks from task models, which are called *UI tasks*.

An *event* is a behavior that happens in the business process that affects the flow of the process and it usually has a cause or an impact. The types of events are: start, intermediate and end. A *start event* indicates where a particular process starts and the mechanism that triggers the instantiation of that process. An *intermediate event* indicates where an event happens somewhere between the start and end of a process and the mechanisms to respond to or to set off an event trigger. An *end event* indicates where a process ends and the consequences of reaching it.

A *gateway* controls the flow of a process as they converge or diverge that either allows or disallows passage through the gateway.

These Flow Objects can be connected to each other and to other information through three types of *Connecting Objects*: sequence flow, message flow and association. A *Sequence Flow* shows order of activities performed in a process. A *Message Flow* shows the flow of messages between two elements in a process that are prepared for that. An *Association* links artifacts with flow objects.

These primary elements can be grouped in *swimlanes*, which help partition and organize activities and other elements and they indicate the participant of a process who performs such group of activities.

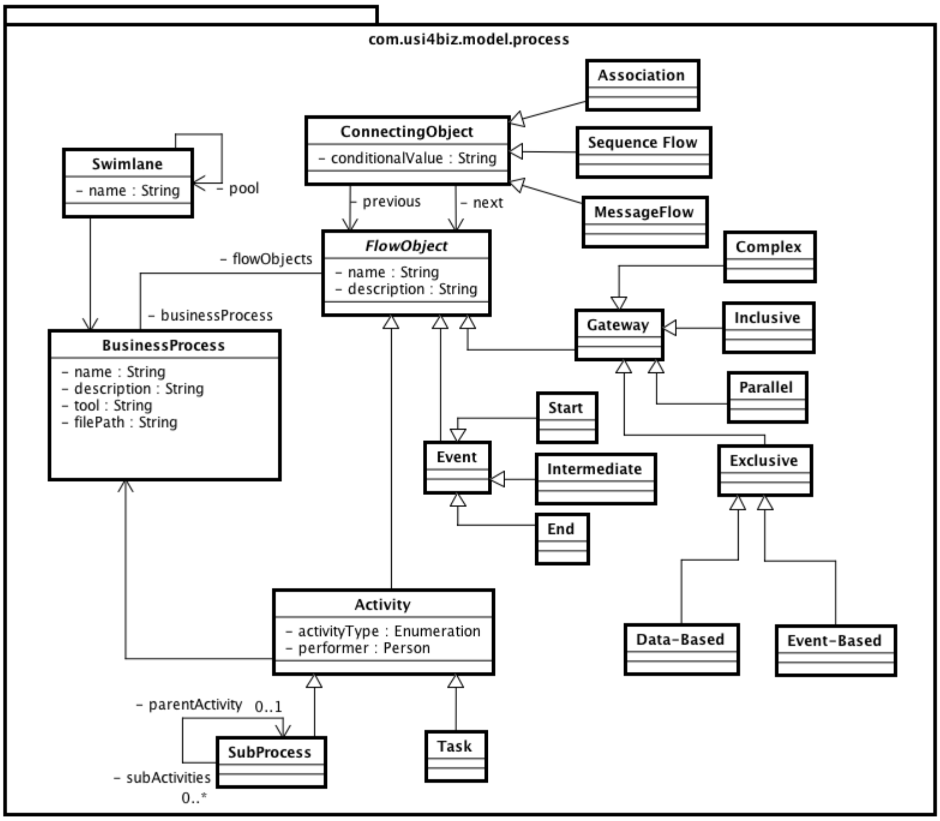

Figure 3.4: Meta Model of the business process

Figure 3.4 depicts the primary business process elements that are linked with the other model elements. The names of the business process elements follow the BPMN specification because it represents one of the most well-known and accepted standards for business processes. Even though the naming standard from BPMN is used, this model is applicable to other standards. A recent comparison of business process meta models [73] has gathered a set of high-level concepts that are included in various specification and standardization efforts. These concepts are directly compatible with BPMN elements. These concepts include: Task (BPMN Activity); Control Flow (BPMN Connecting Object, Gateway); Events (BPMN Event); Roles (BPMN Swimlane); and other concepts related to the interchange of information during the execution of the process, which is out of the scope of this meta model.

In addition to business process modeling, different types of conceptual models have been valued on the 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

[142]. In the cycle of improving processes and updating applications accordingly, 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. Business processes are often not representative of the necessary flexibility for user interaction, however its structure is important to decide how to make the relationship between business processes and UI explicit.

Considering the importance of user interaction to make IT add value for organizations, we propose aligning business processes with enterprise systems through a UI conceptual model: the task model.

Task Model

Task Model is used to *"describe how activities can be performed to reach users' goals when interacting with the application considered"* [90]. With this perspective towards users who interact with UIs, we call these tasks, *UI Tasks*, differentiating them from BP tasks. Task models are designed in a task hierarchical structure that contains different levels of abstraction, which starts with tasks in the highest level that are further decomposed in sub-tasks in the intermediary levels until the lowest level. In this structure, a new level is created whenever there is the need for a better description of what to do. UI Tasks have basic properties, such as name, description, optionality and iterative mode. With the support of the cooperative part of the task model that indicates the relationships among tasks performed by different users, the property user is fundamental, thus adding a representation for user roles in cooperative works [77].

The *types of UI Tasks* in the task model are: (i) abstract, grouping of related tasks; (ii) user, task performed manually by a human being; (iii) interactive, task performed by the user interacting with the system; and (iv) system, task executed by the system itself.

For the purpose of mapping business process activities that are to be performed by users when interacting with enterprise systems, we have delimited the scope to consider only interactive tasks in the traceability chain. Because we deal only with UIs, we do not need manual (user) tasks and automatic (system) tasks. Manual and automatic tasks do not require user interaction, and we have restricted our needs to interactive tasks only. Consequently, we simplify the task model with only one kind of task, which also contributes to increase the task model acceptance by providing simplicity.

Task relationships are one of the main characteristics in task models that have an impact on UI design. The task relationships follow the LOTOS temporal operators. LOTOS [45] is an international standard notation used in various industrial and research centers to specify applications which have, as an important feature, temporal ordering among possible actions. As follows, we present a new graphical representation for the LOTOS temporal operators as our contribution for the task model followed by the definitions for each *relationship type* [65].

The element *FlowObjectUITask* directly associates UI Tasks with the BP element FlowObject, which represents activities, events and gateways. This new element is created to enable any kind of BP Flow Object to be associated to UI Tasks in a many-to-many relationship. For instance, a BP activity can be associated to several UI Tasks, when users perform several minor tasks using the system to complete a BP activity; and a UI Task can be associated to several BP activities, when people perform activities from different parts of the process by doing the same UI Task using the system.

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 UI tasks should be grouped in the same screen); and a variety of temporal ordering of UI tasks, which delivers the flexibility and representational power that users need while interacting with ISs. Therefore, these UI tasks are essential requirements to conceive UIs.

Figure 3.5 depicts the primary task model elements that are primordial to be linked with the other model elements. This task model is based on UsiXML, the main difference in the task model for the UI-Business alignment framework is the addition of FlowObjectUITask to associate the task model with the business process model.

Table 3.1: New Graphical representation for Task Relationships

<table>
<tr><th>Relationship</th><th>Lotos symbol</th><th>New symbol</th><th>Description</th></tr>
<tr><td>Enabling</td><td>T1 >> T2</td><td></td><td>T1 has to be finished in order to initiate T2.</td></tr>
<tr><td>Enabling with information passing</td><td>T1 []> T2</td><td></td><td>T1 has to be finished in order to initiate T2 and T2 is synchronized with T1 on some piece of data.</td></tr>
<tr><td>Suspend / resume</td><td>T1 |> T2</td><td></td><td>T2 may interrupt T1 before the termination of T1. Once T2 is finished, T1 may be resumed.</td></tr>
<tr><td>Deterministic choice</td><td>T1 [] T2</td><td></td><td>Once one task is initiated the other cannot be accomplished anymore.</td></tr>
<tr><td>Disabling</td><td>T1 [> T2</td><td></td><td>T2 may interrupt T1 before the termination of T1, but T1 cannot be resumed after T2 has terminated.</td></tr>
<tr><td>Order Independence</td><td>T1 |=| T2</td><td></td><td>T1 and T2 can be performed in any order.</td></tr>
<tr><td>Concurrency</td><td>T1 ||| T2</td><td></td><td>T1 is interleaved with T2.</td></tr>
<tr><td>Concurrency with information passing</td><td>T1 |[]| T2</td><td></td><td>T1 is interleaved with T2 while they synchronize on some data.</td></tr>
</table>

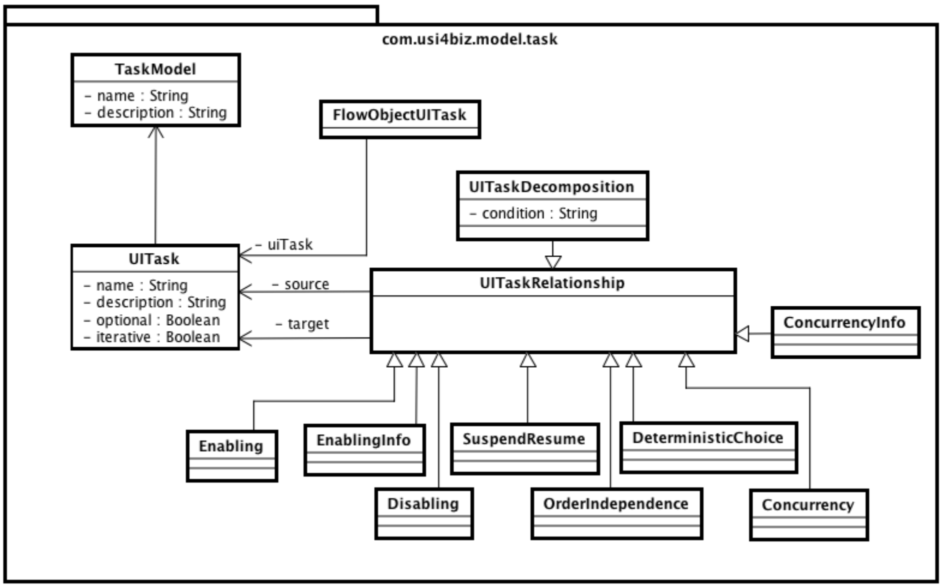

Figure 3.5: Meta Model of the task model

Figure 3.6 depicts the UsiXML task model and how the concepts of the task model used here are based on UsiXML. The elements that are circled in the UsiXML task model are considered in our model.

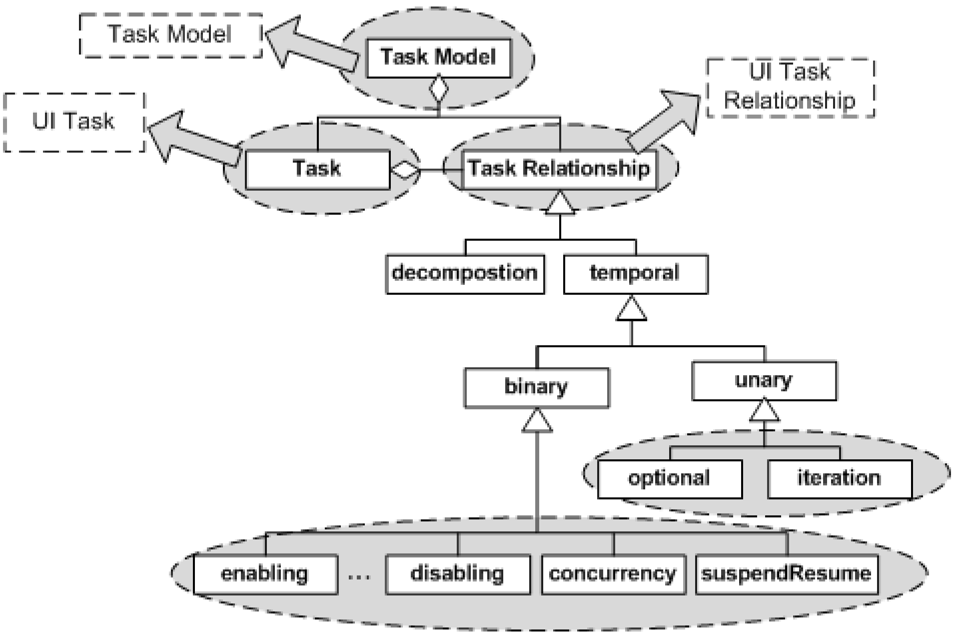

Figure 3.6: Compatibility with UsiXML task model

User Interface Model

UI is a means by which users interact with the system. One of the main challenges of models aiming to formalize user interaction is to follow the fast technological evolution of devices and new kinds of interactions. This problematic has been our major motivation to investigate which model is the most appropriate to cover so many possibilities. In a survey comparing existing UI modeling languages [109], the abstract UI model seems to be the one that covers the broadest possible range of targets for the broadest possible range of delivery contexts. Therefore, we came up with an extension of the abstract UI model based on UsiXML. The difference is that the terms are not used literally because this UI model needs simplicity on the names and quantities of UI components that should be easily understandable in order to be used by end-users when delineating the UI model. This different representation of the UI model does not substitute the language, which is used to guarantee interoperability.

The UI model structure is comprised of *UI components*, which are components used for user interaction that start in an atomic level and are further composed in broader levels.

The following UI components are explained in natural language and with informal predicate logic [37], also called first-order predicate calculus or simply first-order logic [103], which is expressive enough to represent commonsense knowledge, useful to solidify our contribution for UI models. Within first order logic, two definitions are presented since they are used throughout the UI components' formal definitions: universal instantiation and binary relation.

The rule of *Universal Instantiation* [103] is an inference of a sentence obtained by substitution of a ground term (a term without variables, such as a screen element) for the variable (e.g. the type of a screen element). This reference rule is written using the notion of *substitution*, where $\mathcal{SUBST}(\Theta, \alpha)$ denotes the substitution of Θ to the sentence α.

A *binary relation* is a subset of a *Cartesian product*, which is represented by ordered pairs whose first component is a member of a specific set (e.g. a set of screen elements) and whose second component is a member of another set (e.g. a set of screen fragments).

Screen Element (Definition 3.1) is:

- any UI widget that serves as input and/or output for a UI task, it is the most atomic component;

- an *instance* of a type belonging to the set of types of screen elements (e.g. button, combo box, list box, label, radio button);
- instantiated through the rule of *Universal Instantiation* [103] by substitution of a screen element for a type of screen element.

For a screen element e:

$$\begin{aligned} e \in \mathcal{E} = \{e_1, ..., e_n\} \wedge \\ \mathcal{TE} = \{textbox, checkbox, label, ...\} : \\ \frac{\forall e \ \alpha}{\mathcal{SUBST}(\{e/\mathcal{TE}\}, \alpha)} \end{aligned} \tag{3.1}$$

Screen Fragment (Definition 3.2) is:

- a portion of the user interface that is semantically related;
- an *instance* of a type belonging to the set of types of screen fragments (e.g. tab, menu);
- instantiated through the rule of *Universal Instantiation* by substitution of a screen fragment for a type of screen fragment;
- a container of a set of UI components, which are either in the set of screen elements, or in the set of screen fragments, or in both;
- where screen elements are a *proper subset* of screen fragments - i.e. E is a subset of, but not equal to, F;
- there is a *binary relation* between the set of screen elements and screen fragments.

For a screen fragment f:

$$\begin{aligned} f \in \mathcal{F} = \{f_1, ..., f_n\} \wedge \\ \mathcal{TF} = \{panel, menu, tab, ...\} \wedge \\ \frac{\forall f \ \alpha}{\mathcal{SUBST}(\{f/\mathcal{TF}\}, \alpha)} \wedge \\ f = container(x_1, ..., x_n) : \\ \forall x \ x \in (\mathcal{E} \cup \mathcal{F}) \wedge \\ \mathcal{E} \subseteq \mathcal{F} \wedge \\ \exists \ relationship \ \mathcal{R} \subseteq \mathcal{E} \times \mathcal{F} \end{aligned} \tag{3.2}$$

Screen (Definition 3.3) is:

- a state of the user interface when executing a UI task or part of a UI task;
- an *instance* of a type belonging to the set of types of screens (e.g. page, dialog box);
- instantiated through the rule of *Universal Instantiation* by substitution of a screen for a type of screen;
- a container of a set of UI components, which are either in the set of screen elements, or in the set of screen fragments, or in both;
- where screen fragments are a *proper subset* of screens;
- there is a *binary relation* between the set of screen fragments and screens.

For a screen s:

$$\begin{aligned} s \in \mathcal{S} = \{s_1, ..., s_n\} \wedge \\ \mathcal{TS} = \{page, pop-up, ...\} \wedge \\ \frac{\forall s\ \alpha}{\mathcal{SUBST}(\{s/\mathcal{TS}\}, \alpha)} \wedge \\ s = container(x_1, ..., x_n) : \\ \forall x\ x \in (\mathcal{E} \cup \mathcal{F}) \wedge \\ \mathcal{F} \subseteq \mathcal{S} \wedge \\ \exists\ relationship\ \mathcal{R} \subseteq \mathcal{F} \times \mathcal{S} \end{aligned} \tag{3.3}$$

Screen Group (Definition 3.4) is:

- a group of closely related screens;
- an *instance* of a type belonging to the set of types of screens groups (e.g. group);
- instantiated through the rule of *Universal Instantiation* by substitution of a screen group for a type of screen group;

- a container of a set of UI components, which are either in the set of screens, or in the set of screen groups, or in both;
- where screens are a *proper subset* of screen groups;
- there is a *binary relation* between the set of screens and screen groups.

For a screen group g:

$$\begin{aligned}
g \in \mathcal{G} = \{g_1, ..., g_n\} \wedge \\
\mathcal{TG} = \{group\} \wedge \\
\frac{\forall g \ \alpha}{\mathcal{SUBST}(\{g/\mathcal{TG}\}, \alpha)} \wedge \\
g = container(y_1, ..., y_n) : \\
\forall y \ y \in (\mathcal{S} \cup \mathcal{G}) \wedge \\
\mathcal{S} \subseteq \mathcal{G} \wedge \\
\exists \ relationship \mathcal{R} \subseteq \mathcal{S} \times \mathcal{G}
\end{aligned} \tag{3.4}$$

The broadest components are composable in nature, unlike the most atomic component. Table 3.2 depicts the composition and constraints of the UI components. Screen Elements are mandatory atomic components. Screen Fragments are optional components, they may contain screen fragments and it must contain at least one screen element. Screens are mandatory components, they may contain screen fragments and it must contain at least one screen element. Screen Groups are optional components, they may contain other screen groups and it must contain at least one screen.

Table 3.2: Decomposition of UI Components

UI Component	**Symbol**	**Composed of**	**Constraint**
Screen Element		-	Mandatory
Screen Fragment		Screen Fragment Screen Element	Optional At least one Screen Element
Screen		Screen Fragment Screen Element	Mandatory At least one Screen Element
Screen Group		Screen Group Screen	Optional At least one Screen

The element *UIComponentUITask* directly associates UI Components with UI Tasks. This new element is created to enable any kind of UI Task to be associated to UI Components in a many-to-many relationship. In this kind of relationship, a UI Task can be associated to several UI components (i.e. when a UI task is completed after users fill-out several screen elements); and a UI Component can be associated to several UI Tasks (i.e. when a specific screen fragment requires users to perform several minor UI tasks using the system).

Screens, Screen Fragments and Screen Elements have their specific UI Component Types. Screens can be presented as pop-ups, dialog boxes, pages, among others. Screen Fragments can be represented as panels, menus, tabs, etc. Screen Elements can be depicted as fields, labels, buttons, links, etc. The organization of the UI model in these UI components makes it comprehensive enough to consider most types of UI, even those with elements poorly specified or even ignored by UI representations in general, such as video, music player, 3D rendering, etc. For any new type of UI element, it is enough just to identify it as a new type in the existing set.

The organization of UIs in these components are closely aligned with the UI elements used for essential UI prototyping by Ambler [5], which focuses on users and their usage of the system. An essential UI prototype has major UI elements (a large-grained item, potentially a screen) and minor UI elements (a small-grained item, such as input fields, labels, etc.). Going further in details with our UI components, screen groups, screens and screen fragments fall under the category of major UI elements; and screen element under the category of minor UI elements.

The element *UserInteraction* directly associates UI Components with End-Users, for instance, an end-user can be associated to a screen group and this link is replicated to its sub-components. This new element is created to identify who are the end-users impacted when a change is done in certain UIs.

Figure 3.7 depicts the primary user interface model elements that are primordial to be linked with the other model elements.

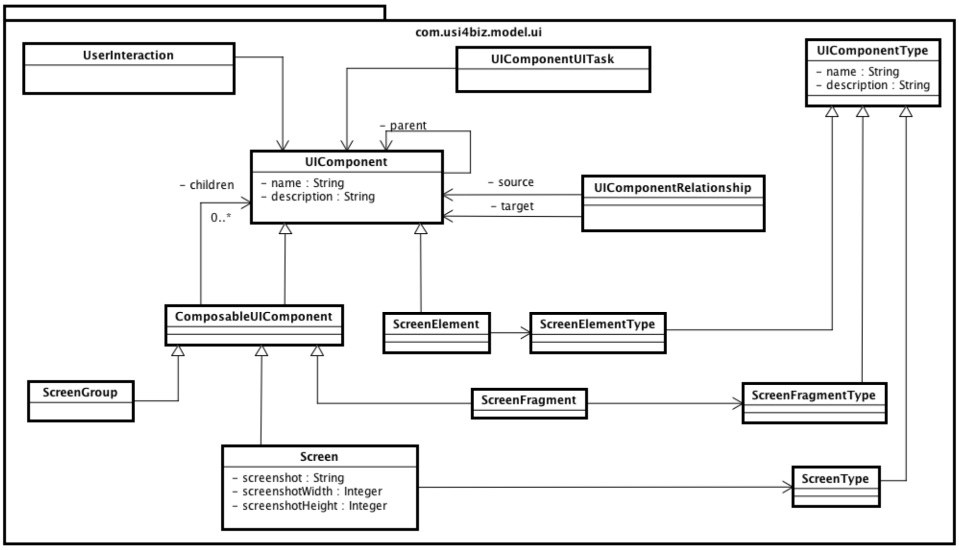

Figure 3.7: Meta Model of the user interface model

Figure 3.8 depicts the UsiXML abstract user interface model and how the concepts of the UI model used here are based on UsiXML, with the addition of UserInteraction to associate the UI model with the end-user and the UIComponentUITask to associate the UI model with the task model.

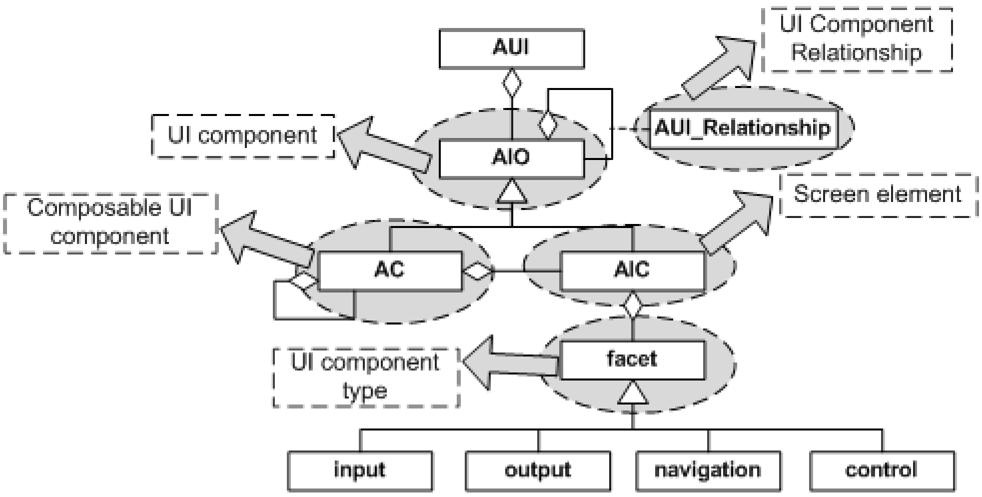

Figure 3.8: Compatibility with UsiXML abstract user interface model

Core Model

When dealing with UIs, one must consider other important concepts: Organization, System and End-User. *Organization* is a company that performs business processes using enterprise systems. *System* is an application composed of a set of UI components. *End-User* is a *Person* within an organization that can use systems (i.e. End-user) and/or design and manage processes (i.e. Business Analyst).

Figure 3.9 depicts the core elements that are primordial to be linked with the other model elements.

Large organizations normally have extensive enterprise systems (e.g. a bank may have a system for each product: billing, loans, payments, investments, insurance) that are used either by professionals within the organizations or directly by customers. In such a context, when the process changes, probably, the system should change accordingly. But, to determine what exactly needs to be changed is subject to a precise traceability control, which becomes more complex proportionally to the size and changing frequency of processes and of its systems. Therefore, we have specified how these models are mapped, explained in the following sub-section.

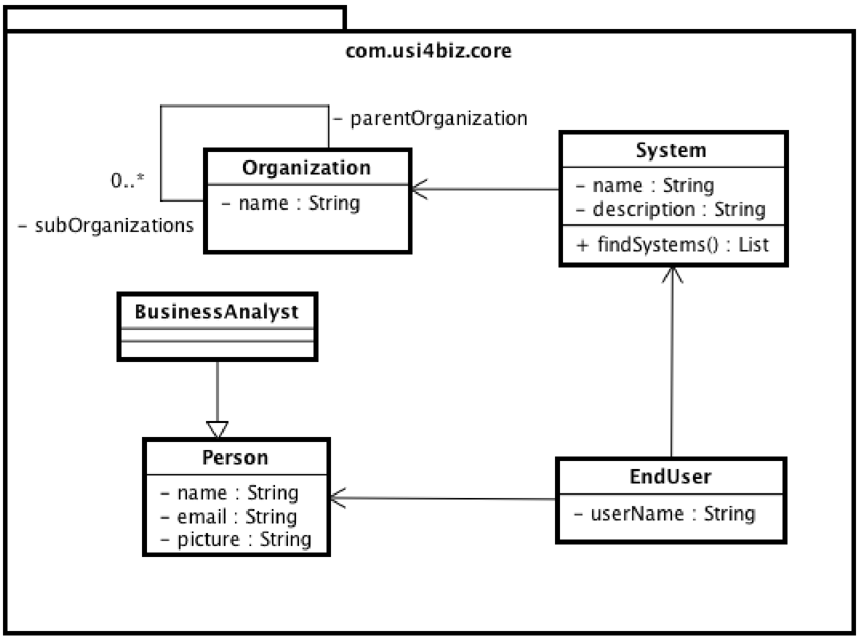

Figure 3.9: Meta Model of the core of the framework

3.2.3 Model Mapping

The association of business process, task model and UIs, depicted in Figure 3.10, is aimed to support organizations in maintaining their enterprise systems aligned with constantly evolving business processes taking a user-centered perspective. On the left of the figure, there is the business process model decomposed in layers, according to BPMN. These levels of the business process are associated with the hierarchical levels of a task model, on the middle of the figure. On the right of the figure, the different levels of the task model are associated with different UI components.

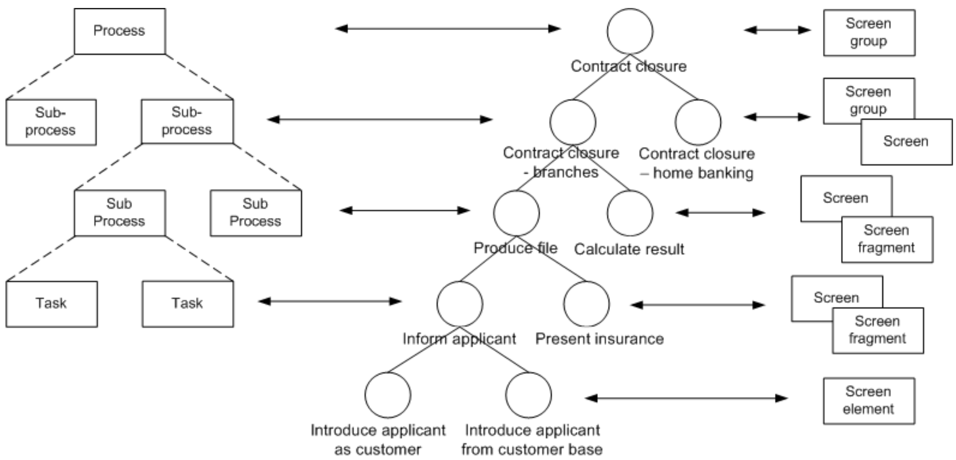

Figure 3.10: Association of business processes with user interfaces

Some organizations do not detail their business processes to a fine-grained 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), as proposed in [52]. Others do detail their business process until a detailed description of activities, until the smallest action.

Since we can come across organizations that follow these divergent modeling techniques, it is aimed that our approach addresses both realities. Therefore, our solution is also applicable in models with operational (detailed) and process (high-level) perspectives. In both cases, task models are created from business processes; the main difference is that the refinement is more fine-tuned when the BP is in a process perspective.

Task models are used as an efficient conduit between business processes and UIs not only because of its representativeness for UI design, but also because of its user-perspective. Unlike other UI design models, such as the dialog model [71] or the interaction model [9] that captures the dynamic behavior of UIs, task models are further from the implementation of UI components and closer to how users interact with those components.

Some existing solutions link business processes directly with systems or UIs; but business processes are often an abstract representation of the business tasks. 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. They are 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 the task model that makes it a strong representation in terms of information needed for alignment with UIs of enterprise systems:

- Task models encompass tasks that are intrinsic to user interaction, such as cancel, save temporarily, undo, etc.
- In most cases, task models are detailed enough to describe individual behavior, which has strong influence from the context of use.
- Task models have a hierarchical structure that has the same purpose of the business process structure: from high level to detailed description of the work to be done.
- Task models help identify issues in the user interaction that have a direct impact on process performance.

As a result, the symmetry of both structures represents a clear bridge between business processes and UIs. Business process models and task 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 map business processes with UIs of enterprise systems.

In other researches, task models have been used to support requirement analysis, design of ISs and usability evaluation [77], to automatically generate UIs [131], and even to model user roles with different responsibilities and rights in large applications [93], but it had not yet been used as a source for traceability, as we have proposed in [111]. Therefore, we emphasize that the framework proposed here does not aim to automatically generate UIs because we argue that the work of UI designers is relevant to achieve user satisfaction. UI designers with the human interpretation over user experience, user needs, and design ideas can create UIs that are suitable to the constraints imposed by specific projects. The framework is focused on aligning business processes with user interfaces, therefore it is not in the scope of this work to detail, compare, or assess the different techniques for UI generation, but we consider the existing contributions as a support for model mapping.

Mapping Characteristics: The three core models of this framework have been mapped, thus creating a traceability chain that is useful, among other things, for performing impact analysis and providing specifications for transformations between these models. These mappings have the following characteristics [74]:

- They allow expressing *one-to-one, one-to-many, many-to-one and many-to-many mappings* between source and target models;
- They include *endogenous mappings*, which are between source and target models of the same domain (e.g. between models for UI design) and *exogenous mappings*, which are between source and target models of different domains (e.g. between models for UI design and BP modeling);
- They specify *horizontal mappings*, which are done between models in the same level of specification (e.g. between BP and task models that are independent of the computation viewpoint) and *vertical mappings*, which are done between models in different levels of specification (e.g. between task models, with a computation-independent viewpoint, and UI models, with a platform specific viewpoint).
- They specify *syntactic mappings*, which considers the structure in which the model elements are constructed (e.g. mapping between UI task and UI models); and *semantic mappings*, which considers the meanings of the model elements (e.g. mapping between BP and task models).

Following, we present how BP, task model and UI components can be mapped.

Mapping BP Model and Task Model

After a detailed analysis of the specification on BPMN, it was detected that business process elements can be transformed into elements in a task model [114]. The mapping between the business processes with task models is organized in four basic categories of elements with their specific subdivisions, as proposed in BPMN:

1. Flow Objects, sub-divided in: Events, Activities, Gateways.
2. Connecting Objects, sub-divided in: Sequence Flow, Message Flow, Association.
3. Swimlanes, sub-divided in: Pool and Lane.
4. Artifacts, sub-divided in: Data Group, Group, Annotation.

Following the order of this classification, we will describe in more details how each element of the business process is related with the task model.

Flow Objects:

Events

Table 3.3 specifies the association of the mappings between Events in business processes and elements in task models, with the rationale behind the decisions explained as follows.

Table 3.3: Association of BPMN events and UI task elements

No.	Business Process	Task Model
1	None Start Event	- -
2	Message Start Event	UI Task (+ Enabling w/ info passing)
3	Timer Start Event	UI Task (+ Enabling w/ info passing)
4	Conditional Start Event	UI Task (+ Enabling w/ info passing)
5	Signal Start Event	UI Task (+ Enabling w/ info passing)
6	Multiple Start Event	UI Task (+ Enabling w/ info passing)
7	None End Event	- -
8	Message End Event	(Enabling +) UI Task
9	Error End Event	Disabling + UI Task

Continued on next page

No.	**Business Process**	**Task Model**
10	Cancel End Event	Disabling + UI Task
11	Compensation End Event	Enabling w/ info passing + UI Task
12	Signal End Event	Enabling + UI Task
13	Terminate End Event	- -
14	Terminate End Event	Stop operator
15	Multiple End Event	*Check event detail attribute*
16	None Intermediate Event	UI Task
17	Message catch Intermediate Event	Enabling + UI Task
18	Message throw Intermediate Event	UI Task + Enabling w/ info passing
19	Timer Intermediate Event	Enabling + UI Task
20	Error Intermediate Event	Enabling w/ info passing + Subtasks + Suspend/resume
21	Cancel Intermediate Event	Enabling w/ info passing + Subtasks + Disabling
22	Compensation catch Interm. Event	Enabling w/ info passing + Subtasks + Disabling
23	Compensation throw Interm. Event	Enabling w/ info passing + UI Task
24	Conditional Intermediate Event	Enabling w/ info passing + Subtasks + Deterministic choice
25	Conditional Intermediate Event	Suspend/resume + UI Task
26	Link Intermediate Event	- -
27	Signal catch Interm. Event	Enabling w/ info passing + Subtasks + Disabling
28	Signal catch Interm. Event	Disabling + UI Task
29	Signal throw Interm. Event	UI Task + Enabling w/ info passing
30	Multiple Intermediate Event	*Check event detail attribute*

Start Events

Start Events have six types in BPMN: None, Message, Timer, Conditional, Signal, and Multiple.

(1) None start event just starts the flow, so it does not need to be substituted by any element in the task model, which starts directly with the first UI Task.

(2) Message start event can be represented by a UI Task to receive a message (e.g. manager acceptance arrived) followed by the task relationship "enabling with information passing" to pass the message.

(3) Timer start event can be represented by a UI Task to verify the time-date or specific cycle (e.g. every Monday at 9am) followed by the task relationship "enabling with information passing" to pass the time.

(4) Conditional start event can be represented by a UI Task that verifies a certain condition (e.g. loan acceptance rate is lower than 35%) followed by the task relationship "enabling with information passing" to pass the result of the condition.

(5) Signal start event can be represented by a UI Task that verifies if a signal arrived (e.g. detection of fraudulent information) followed by the task relationship "enabling with information passing" to pass the detection of the signal.

(6) Multiple start event means that there are multiple ways of triggering the Process. The attributes of the Start Event will define which of the other types of Triggers apply. When the start event has multiple triggers, it means it can be one of the other event details, so the multiple start event can be represented similarly to the other start event details.

End Events

There are eight types of End Events in BPMN: None, Message, Error, Cancel, Compensation, Signal, Terminate, and Multiple.

(7) None end event just finishes the flow, so it does not need to be substituted by any element in the task model since it finishes directly with the last UI Task.

(8) Message end event can be represented by the task relationship "enabling" followed by a UI Task to send a message (e.g. warning about loan rejected).

(9) Error end event can be represented by the task relationship "disabling" that interrupts the previous UI Task and terminates the process, followed by a UI Task to generate the error (e.g. information missing for loan request).

(10) Cancel end event can be represented by the task relationship "disabling" that interrupts the previous UI Task followed by a UI Task to cancel the transaction (e.g. system unavailability to complete the request).

(11) Compensation end event can be represented by the task relationship "enabling with information passing" passing the outcome of the previous UI Task that is determined to be undesirable by some criteria (e.g. loan request being canceled) followed by a UI Task to undo the effects, know as compensation (e.g. delete request from pending list).

(12) Signal end event can be represented by the task relationship "enabling" followed by a UI Task to broadcast the signal.

(13) Terminate end event indicates that all activities in the Process should be immediately ended. This is the same as just finishing the last UI Task of the group of UI Task, so no extra element is required in the task model.

(14) Alternatively, the terminate end event could also be substituted by a Stop unary operator [107], which means the unsuccessful termination of a UI Task that inevitably leaves the super-ordinate tasks incomplete, thus leading to the premature termination of a scenario. That could be done once this extension is accepted by a task modeling tool.

(15) Multiple end event means that there are multiple consequences of ending the Process. The attributes of the end event will define which of the other types of results apply. Since cancel and terminate do not have attributes and thus cannot be differentiated, they are not considered as a possible trigger when multiple is applied, so the possible options are: message, error, compensation and signal. Therefore, the task model elements will be selected according to the attributes that define which event detail has occurred.

Intermediate Events

There are 10 types of Intermediate Events in BPMN: None, Message, Timer, Error, Cancel, Compensation, Conditional, Link, Signal, and Multiple.

An Intermediate Event can be placed in one of two manners: placed within the normal flow of a Process and attached to the boundary of an

Activity. In both cases, two activities with the event between them is represented by three UI Tasks in which the first UI Task is an activity, the middle UI Task is the event and the last UI Task in the second activity.

An Intermediate Event that is placed within the normal flow of a Process can be used for one of two purposes. The Event can respond to ("catch") the Event Trigger or the Event can be used to set off ("throw") the Event Trigger. An Intermediate Event that is attached to the boundary of an Activity can only be used to "catch" the Event Trigger.

If the Event is used to "throw" the Event Trigger, the Trigger of the Event will immediately occur and the Token will move down the outgoing Sequence Flow.

If the Event is used to "catch" the Event Trigger, then the Token will remain at the Event until the Trigger occurs.

(16) A None intermediate event is determined by an event that does not specify an event detail. A None intermediate event can be used to indicate some change of state. It can be represented by a UI Task to change the state in the process.

(17) Message catch intermediate event can be represented by the task relationship "enabling" followed by a UI Task to receive a message, which causes the process to continue.

(18) Message throw intermediate event can be represented by a UI Task to send a message followed by the task relationship "enabling with information passing" to pass the message.

(19) Timer intermediate event can be represented by the task relationship "enabling" followed by a UI Task to verify that a specific time-date or specific cycle has been reached to trigger the next UI Task.

(20) Error intermediate event between two other tasks can be represented by the task relationship "enabling with information passing" followed by a UI Task with two sub-tasks linked by the task relationship "suspend/resume" since after the error is treated the other sub-task can be done.

(21) Cancel intermediate event indicates that the activity attached to this event should be canceled when the event is triggered, but it does not terminate the process. When this event is between two other tasks, it can be represented by the task relationship "enabling with information passing" followed by a UI Task with two sub-tasks linked by the task relationship "disabling" since when the cancel is selected, the other UI Task cannot be done.

(22) Compensation catch intermediate event between two other tasks can be represented by the task relationship "enabling with information passing" followed by a UI Task with two sub-tasks linked by the task relationship "disabling" since when the compensation is triggered, the other UI Task cannot be done.

(23) Compensation throw intermediate event can be represented by the task relationship "enabling with information passing" passing the outcome of the previous UI Task that is determined to be undesirable by some criteria followed by a UI Task to undo the effects, known as compensation.

(24) Conditional intermediate event is triggered when a condition becomes true. When it is between two other tasks, it can be represented by the task relationship "enabling with information passing" followed by a UI Task with two sub-tasks linked by the task relationship "deterministic choice" to do one of the sub-tasks, depending on the condition.

(25) Conditional intermediate event attached to a task and followed by another task can be represented by the task relationship "suspend/resume" followed by a UI Task to be done when the condition is true, which momentarily pauses the execution of the first UI Task.

(26) Link throw intermediate event connects one part of the process with the Link catch intermediate event, which is similar to a sequence flow and since these types of event are already connected to sequence flows, so no extra element is required in the task model.

(27) Signal catch intermediate event between two other tasks can be represented by the task relationship "enabling with information passing" followed by a UI Task with two sub-tasks linked by the task relationship "disabling" since when the signal is triggered, the other UI Task cannot be done.

(28) Signal catch intermediate event attached to a task and followed by another task can be represented by the task relationship "disabling" followed by a UI Task to be done when the signal is triggered, which interrupts the execution of the first UI Task.

(29) Signal throw intermediate event can be represented by a UI Task to send a signal followed by the task relationship "enabling with information passing" to pass the signal.

(30) Multiple intermediate event means that there are multiple triggers assigned to the event. The attributes of the event will define which of the types of event apply. Therefore, the task model elements will be selected according to the attributes that define which event detail has occurred.

Activities

Table 3.4 specifies the association of the mappings between Activities in business processes and elements in task models, with the rationale behind the decisions explained as follows.

Table 3.4: Association of BPMN activities and UI Task elements

No.	Business Process	Task Model
1	Sub-process	UI Task
2	Task	UI Task
3	Ad-Hoc marker in sub-process	Sub-tasks + Order Independence
4	Standard Loop + LoopCondition	Iterative
5	Multi-Instance Loop + MI Condition	Iterative
6	Standard Loop + Loop Maximum	Iterative
7	User	Interactive

(1) Sub-process is a compound activity, similar to high level UI Tasks.

(2) Tasks are atomic activities, similar to atomic UI Tasks.

Activity Attributes

(3) Ad-Hoc marker in a sub-process is composed of activities that can be performed in any order. Ad-Hoc marker in a sub-process can be represented by a UI Task to represent the sub-process composed of sub-tasks with the task relationship “order independence” between the sub-tasks.

(4) Standard loop type determines that an activity is performed repeated times. In addition it has the loop condition that is an expression that is evaluated as a condition for each cycle of the loop. This type of task or sub-process can be represented by the UI Task property “iterative”.

(5) Multi-instance loop has a numeric 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 type of task or sub-process can be represented by the UI Task property “iterative”, in which a task can be iterated n times.

(6) Standard Loop has the attribute Loop Maximum that informs the limit to the number of loops. This type of task or sub-process can be represented by the UI Task property "iterative", in which a task can be iterated n times.

Task Types

Task types provide information of the behavior that the tasks might represent. In business processes, tasks are atomic activities. Since we consider interactive UI tasks as the scope of the task model, tasks types that indicate a behavior different than interaction in the business process are out of the scope. For instance, service, receive, send and script task types are related to tasks performed by the application; and manual task types are related to tasks performed by users without an application.

(7) User tasks and Interactive UI Tasks represent tasks performed by a human with the assistance of a system.

Gateways

Table 3.5 specifies the association of the mappings between Gateways in business processes and elements in task models, with the rationale behind the decisions explained as follows.

Table 3.5: Association of BPMN gateways and UI Task elements

No.	Business Process	Task Model
1	Data-Based Exclusive gateway	Deterministic choice + condition
2	Event-Based Exclusive gateway	Deterministic choice + condition
3	Exclusive merge	Deterministic choice
4	Inclusive gateway	Order Independence + condition
5	Inclusive merge	Order Independence
6	Parallel gateway	Concurrency
7	Parallel merge	Concurrency
8	Complex gateway	Enabling + condition
9	Complex merge	Order Independence
10	Inclusive gateway + Exclusive merge	Order Independence + Enabling with info passing
11	Parallel gateway + Data Object	Concurrency with info passing

(1) 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. They restrict the flow such that only one of a set of alternatives may be chosen. Data-based Exclusive gateway is a branching point where the selection of an alternative is based on conditional expressions. It can be represented by a "deterministic choice" relationship with a condition for each target UI Task.

Suggestion: It is necessary to allocate a condition for each target UI Task in the task model for this representation to be complete, not a condition for the relationship as whole because a gateway can have several target activities each one with a different condition, it is not a relationship between just two UI tasks, as in most task models.

(2) Event-based exclusive gateway is a branching point where the selection of an alternative is based on an event that occurs at that point in the process. It can be represented by a "deterministic choice" with a condition for each target UI Task.

(3) Exclusive merge is used as a merge for alternative incoming sequence flows and only one of the alternatives is expected to pass through this point. It can be represented by a UI Task composed of sub-tasks with "deterministic choice" relationship linked to the following UI Task with "enabling with information passing" relationship so the following task knows the incoming task's result.

(4) Inclusive gateway represents a branching point where the selection of alternatives is based on conditional expressions where all combinations of paths may be taken, from zero to all possible alternatives. However, in this case, the true evaluation of one condition expression does not exclude the evaluation of other condition expressions. It can be represented by a UI Task linked with "enabling" relationship to a UI Task composed of sub-tasks with "order independence" relationship and condition for each sub-task because each conditional path is independent, which can be comprised of several target activities. Inclusive gateway does not represent a choice since each outgoing path represents a grouping of related independent decisions taken according to the defined condition.

(5) Inclusive merge is used to synchronize all tokens that have been produced upstream, but at most one for each incoming Sequence Flow. It can be represented by a UI Task composed of sub-tasks with "order independence" relationship linked to the following UI Task with "enabling with information passing" relationship so the following UI Task knows the incoming task's results.

(6) Parallel gateway is a mechanism to create parallel flow by dividing a path into two or more parallel paths. It can be represented by a UI Task linked with “enabling” relationship to a UI Task composed of sub-tasks with “concurrency” relationship.

(7) Parallel merge is a mechanism to synchronize parallel flow. It can be represented by a UI Task composed of sub-tasks with “concurrency” relationship linked to the following UI Task with “enabling with information passing” relationship so the following task knows the incoming task’s results.

(8) Complex gateway has an incoming expression that determines which of the outgoing sequence flows will be chosen for the process to continue. The expression may refer to process data and the status of the incoming sequence flow. It can be represented by a UI Task linked with “enabling” relationship to task composed of sub-tasks with “order independence” relationship and a condition attribute for each sub-task.

(9) Complex merge has an outgoing expression that determines which of the incoming sequence flows will be required for the process to continue. The expression may refer to process data and the status of the incoming sequence flow. It can be represented by a UI Task composed of sub-tasks with “order independence” relationship linked to the following UI Task with “enabling with information passing” relationship so the following UI Task knows the incoming task’s results.

(10) Many combinations of the alternatives can be selected with Inclusive gateway, but it is also necessary to use the Exclusive merge in order to express that only one of the activities is expected to be fully accomplished and its result to be passed through the flow. It can be represented by a UI Task linked with “enabling” relationship to a UI Task composed of sub-tasks with “order independence” relationship with a condition attribute for each sub-task and with “enabling with information passing” relationship so the following UI Task knows the incoming task’s result and accepts only one incoming result.

(11) Parallel gateway with a data object is a mechanism to create parallel flow that synchronizes data. It can be represented by a UI Task linked with “enabling” relationship to a UI Task composed of sub-tasks with “concurrency with info passing” relationship.

Connecting Objects:

Table 3.6 specifies the association of the mappings between Connecting Objects in business processes and elements in task models, with the rationale behind the decisions explained as follows.

Table 3.6: Association of BPMN Connecting Objects and UI Task elements

No.	Business Process	Task Model
1	Sequence Flow + ConditionType = None	Enabling
2	Sequence Flow + ConditionType = Default	Enabling
3	Sequence Flow + ConditionType = Expression	Enabling w/ info passing
4	Message Flow	Enabling w/ info passing
5	Association	Enabling

Sequence Flow

(1) Sequence Flow when the ConditionType is none and the "enabling" relationship represent the order in which activities are performed. By default, the ConditionType of a sequence flow is None.

(2) Sequence Flow when the ConditionType is default represents the order in which activities are performed, similar to the "enabling" relationship.

(3) When the Sequence Flow has ConditionType set to expression means that there is a condition that is evaluated to determine whether or not the flow will be used. This condition can be passed in the "enabling with information passing" relationship.

Message Flow

(4) Message Flow shows the flow of messages between two entities that can be presented by the "enabling with info passing" relationship.

Association

(5) Association is used to associate information (e.g. text and graphical non-flow objects) with Flow Objects. An arrowhead on the Association indicates a direction of flow that can be represented by the "enabling" relationship.

Swimlanes:

Table 3.7 specifies the association of the mappings between Swimlanes in business processes and elements in task models, with the rationale behind the decisions explained as follows.

Table 3.7: Association of BPMN Swimlanes and UI Task elements

No.	**Business Process**	**Task Model**
1	Pool	EndUser
2	Lane	EndUser

(1) Pool represents a participant in a process. A Participant can be a specific business entity (e.g. a company) or a general business role (e.g. buyer, seller, and manufacturer). For each activity within this pool, it can be represented by an EndUser associated to a UI Task.

(2) Lane is a sub-partition within a Pool, therefore, it represents sub-categories of a participant in a process. For each activity within this lane, it can be represented by an EndUser associated to a UI Task.

Artifacts:

Table 3.8 specifies the association of the mappings between Artifacts in business processes and elements in task models, with the rationale behind the decisions explained as follows.

Table 3.8: Association of BPMN Artifacts and UI Task elements

No.	Business Process	Task Model
1	Data Object + Task (in/out)	UI Task
2	Sequence Flow + Data Object	Enabling with info passing

Data Objects are considered Artifacts because they provide information about what activities require to be performed (input) and/or what they produce (output), but they do not have any direct effect on the flow of the process.

The fact of having a Data Object in the BP does not indicate what users will do with it while using the system. For instance, specifying a UI task in which users will upload a scanned document depicts that it is actually the UI task that will be linked to the UI components to fulfill that UI task. Because Data Objects do not have any direct effect on the flow of the process, they are generally associated to Flow Objects or in some cases with connecting objects, but in any case, the BP can be modeled without Data Objects to reduce clutter. Therefore, when Data Objects are present in the BP models, what is considered is the Flow Object or Connecting Object related to it.

(1) A Data Object as input or output of an activity, the activity is considered as the main element, which is represented by a UI Task.

(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. The Sequence Flow is considered as the main element, which is represented by an "enabling with information passing" relationship.

The following elements are not related to UI Tasks: A Group is used to group objects within the same category. A Text Annotation is a mechanism for a modeler to provide additional information for the reader of a BPMN Diagram. Both of these elements do not affect the sequence flow of the activities within the group and do not provide information that directly impacts the process. Therefore, they have not been associated to UI Task elements because they are useful for graphical organization and visualization, not to the performance of the process.

The transformation of business process models into task models is not sufficient to create a task model complete enough to represent all tasks expected from users. This is due to the fact that BPs may not be modeled in much richness of details. In this case, the task model acts as a complementary model, allowing further refinements. Such refinement can be done by human factors experts and UI designers, who update UI 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.

Mapping Task Model and UI Model

To instantiate the mapping of task models with UI components, we take a five-level task model as an example, as depicted in Figure 3.11. The first level of this task model is represented through a screen group and the second level is related to screens. For the second UI task related to the second screen, it is possible to depict the optional characteristic of screen fragments. Its first sub-task is represented through a screen fragment with two screen elements and its second sub-task is linked to a screen fragment that has another screen fragment within it. This example is useful to depict the representation of screen elements present in different levels of the task model, depicted in the third, fourth and fifth levels. It demonstrates the flexibility of task models to represent UI tasks in different levels of details, thus confirming that mapping UI tasks to UI components is specific to each case.

To map UI tasks with UI components there are some constraints that need to be respected to guarantee the coherence of the mapping, such as the ones listed as follows:

- A UI task with sub-tasks cannot be linked to a screen element;
- If the root UI task is a screen group, then the UI task directly below it is represented by a screen group or a screen;
- If the UI task is linked to a screen group, then the UI task directly below it is represented by a screen group or a screen;
- If the UI task is a screen, then the UI task directly below it is represented by screen fragment or a screen element;
- If the UI task is a screen and is the second to last level, then the UI task directly below it is represented by screen element;
- If the UI task is a screen fragment, then the UI task directly below it is represented by a screen fragment or a screen element;
- If there are three or more levels below a screen level, then the UI task directly below it is a screen fragment.
- All UI tasks should be linked with UI components because all UI tasks are interactive;
- If there are UI components not linked with UI tasks, it is probably due to the granularity of the task model;
- By default, the sequence in which UI tasks are executed coincides with the sequence of UI components navigation. However, this sequence could be changed in each model.

Since the UI-Business Alignment Framework is independent of the way UIs are designed and developed, it is valuable to use the task model hierarchical structure that does not limit in how many levels a screen can be decomposed, as previously specified in Table 3.2. Therefore, the more fine-grained is the specification of the UI tasks, the clearer it is to identify the UI components that will be used in the UI.

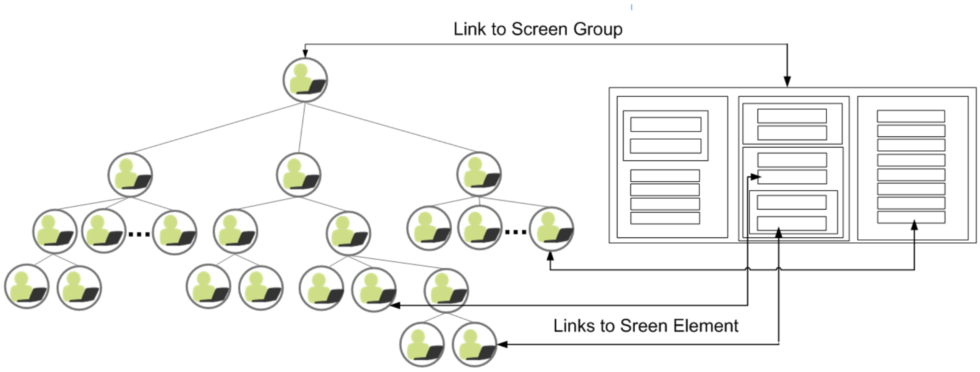

Figure 3.11: Association of business processes with user interfaces

The mapping between these different models represents a solution for the problem of how business processes influence user interfaces and vice-versa. This solution has been presented using specific notations for each model as an example. The business process model is represented using BPMN, the task model is represented using UsiXML task model and the user interface is represented using the UI model proposed in this thesis, which is also founded on UsiXML. Even though the task and UI models are based on UsiXML, these models are not tied to a particular UIDL, thus another UIDL could be used (e.g. XIML), if the language has the corresponding concepts. As an evolution of this work, the coverage of the meta-models can be tested by using other notations. In this case, the business process model can also be represented using the ArtiFact™Model, the task model can be represented using User Action Notation [38] and the user interface model can be represented using directly the UsiXML abstract UI model.

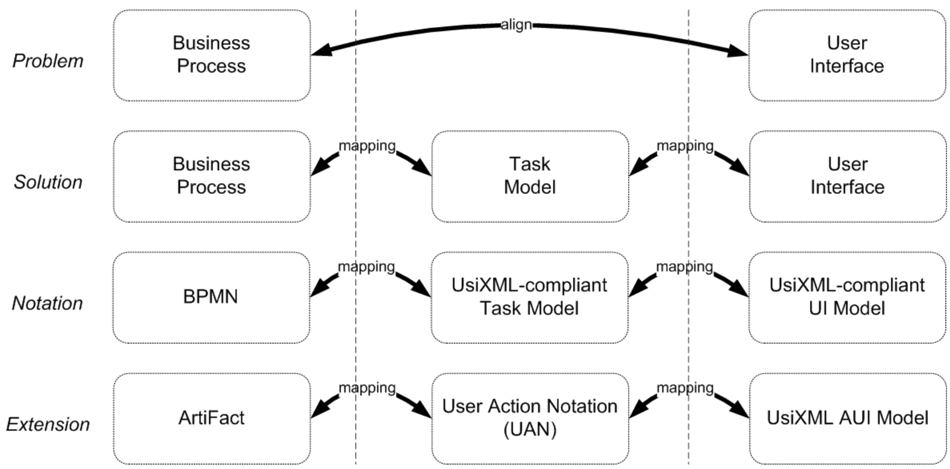

Figure 3.12: Framework in different levels

3.3 UI-Business Alignment Methodology

The methodology is explained with the specification of the organizational context, its core actions, how they can be applied through different approaches, depending on the corporate background, with the description of its characteristics that represent the main differentials of this methodology, presented in the following sub-sections.

3.3.1 Organizational context

The methodology is explained through its application in two large organizations, which illustrates the applicability of this framework in complex scenarios. Therefore, presenting a generic organizational context and specific scenarios of the case studies is primordial to understanding the methodology and the examples illustrated with cases experienced in these organizations.

The context in which this framework is appropriate is for companies that: 1) are large because they usually have plenty of systems in production, these systems are running in different platforms and there are hundreds of user interfaces being used every day; 2) adopt business processes, which are the foundation for their enterprise systems and are all in constant evolution and/or 3) have extensive enterprise systems that automate the corporate operations, but have not yet designed their business processes but are interested in adopting a business process oriented strategy. These cases represent situations in which the organization needs to maintain their systems in accordance with the business processes in order to have systems that evolve according to the business strategy. But such alignment is not easy to achieve throughout different departments within the organization. In addition, each department often works separately and has serious difficulties in understanding the work of other departments, such as business and IT departments.

There are two main points of views that are handled in the target organizational context: the business managers and the users of enterprise systems, who are business process performers. For the top-management, it is crucial for the organization to have enterprise systems supporting their business strategies that are translated into activities in the processes. Once systems are aligned with the defined processes, it is easier to achieve their strategic goals. On the other hand, there are the users of enterprise systems who are: 1) professionals within the organization who use them in their daily work, in some situations as a mediator for the organization's customer; and 2) the organization's customer who has access to the organization's products or services through an on-line version of the enterprise system.

This section provides information of two separate organizations in which the methodology has been applied, constituting the experience in two case studies. We use scenarios derived from these two real case studies to effectively cover the several topics and validate the framework. One organization is from the bank sector, a large bank and insurance company; the other organization is from the telecommunications industry, the largest telecommunications company in the country.

Bank Sector

This organization is decomposed in two main sub-divisions: the insurance and the bank. This sub-division resembles having two organizations in one, which is an even more complex situation to be addressed. 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: *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 professionals and departments are organized as follows: business analysts from the re-engineering department; system analysts, software architects and programmers from the IT department; and UI designers and human factors experts from the UI design department.

They adopt process practices in the whole organization. Their methodology is designed in the following manner: the Re-engineering team models business processes; the UI design team designs UIs based on templates and style guides; Human factors experts evaluate UI usability and coherence with organizational standards; the Re-engineering team reviews UIs and accordance to processes; and the IS team implements UIs. This methodology is executed in an iterative manner and when there are problems detected by human factors experts or when business analysts detect some needs for improvements or corrections, UIs need to be re-designed or processes need to be updated.

In their methodology, they specify business processes with flows of sub-processes and activities until reaching an atomic level in which tasks are associated to business rules and data. In general, their business process models are composed of 60 elements ranging from process, sub-processes, activities, and tasks, not even mentioning their business rules, which would increase this number by nearly 100 more elements. Because of this large number of processes; 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.

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. Business domain represents the core and supporting processes. Processes present the generic flows per business domain. Process in a logical channel shows the process flow taking into account a specific logical channel (e.g. home banking, branches, back office). Sub-process presents a process for a logical channel in a flow. Activities represent details of a sub-process in a flow. Tasks give further detail of an activity in a flow. They are documented in two different documents, a global and a detailed description. The Re-engineering team expects 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 an attempt to align and correlate business process layers with UIs elements, which can start with screens until basic objects (e.g. fields).

The IT and UI design 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 address various demands, especially for UI design and usability evaluations. As a consequence, they are required to spend a considerable amount of time in meetings to understand the business processes or doing quality assurance checks to make sure that the UIs are in accordance with the business processes; otherwise, 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 main challenge is to make these sub-divisions communicate clearly and efficiently, especially because they are physically separated. Their issues are related to scarceness of product-knowledge and human resources in the UI design department have led to two possible scenarios: (i) UI designers do their work in an ad-hoc manner, considering previous designs and insights from professionals who are more experienced with the products; or (ii) the IT department, with more staff and product knowledge, takes the lead and implements before UI design. In the first case, the business process is rarely followed. In the latter, it is difficult for human factors experts to make sure the outcome of usability evaluation is put into practice since development has started before UI design.

Telecommunications Industry

This telecommunications company is a reference in providing integrated communication services, including fixed line voice, mobile voice, broadband, and television services (for confidentiality reasons, the characteristics of this organization are presented in general terms). The link with the existing businesses lies in the technology; customers can access any of these four products with their existing infra-structure, which makes the diversification strategy conglomerate and offers simplicity to the customer. Enhancing its marketing strategy, it has promoted diversification and expanded the scope of its services to include Voice over IP (VoIP), which is a new trend, offering phone communications via the IP network.

The high-level management has stated that the organization will give more attention to its business processes. To address this goal, in 2007, they have started to use the Enhanced Telecoms Operations Map (eTOM) framework [26] as a background for modeling its BPs. This standard is used worldwide as a guidebook for the Telecommunication sector to provide such companies with a general description of processes. In addition to this framework, the consensus is to use Microsoft Visio for operational flows and Aris [6] for IT core system related flows. In practice, however, many documents exist to document the BPs, not centrally available and not in a common format. In other words, there is no centralized tool for BP modeling. It owns 40 licenses of Aris; some licenses are used in Human Resources, the others are spread in the IT unit. However, only 3 to 4 people are using the licenses regularly, and these people are not the Process Managers. In addition to the cost of such licenses, there is a resistance to use a modeling tool. Indeed, people are used to Microsoft tools (Visio, Word and PowerPoint) but such tools have not been created for such a purpose.

The organization has defined four levels of BPs design (independent of the tool used), each level adding more details to the previous one. Level 1 broadly describes who does what in the process; it contains the major entities and actors, but does not involve any decisions or systems. Level 2 complements by detailing the systems used by these professionals as well as the decisions which have to be made. Level 3 concretely represents the process flow and its description; this level is skipped for small processes. Finally, level 4 belongs to the Methods department; it explains how the user has to interact with the systems' User Interfaces. This documentation is often made of screen shots with some annotations for the user. In addition, the BP models are just used as documentation and are rarely updated.

Besides the greater importance given to modeling business processes, they are also concerned with managing them; hence one of its main goals for the next 5 years is to fuse its sales processes so that all different sales channels can provide services to their customers uniformly. However, their current situation at the time of the case study is the lack of consistency and traceability in the documentation. A business process engineer stated that depending on the needs, people produce documentation from scratch and once the project is finished, they "throw it away". In addition, there is no linkage between the existing documentation in different departments and everyone is using the modeling tool of their choice. From then on, it becomes increasingly difficult to predict the impact of changes (traceability) and to be aligned (consistency) across departments. This problem is frequent and relevant as knowledge represents the organization's ability to determine how it will deliver results, how it will differentiate from competitors (strategy) and how it will cope with changes. All of these characteristics make this organization an appropriate context to apply the UI-Business Alignment framework.

To address issues related to the alignment of enterprise systems faced in these organizational contexts, the methodology is proposed to align their business processes with UIs. The goal is to maintain both an organizational point of view, to be compliant with decisions from top-management; and an end-user perspective, to maintain systems usable and efficient. Situations from these two organizations are used throughout this work to explain the methodology and platform. In the following sections, the UI-Business Alignment Methodology is explained and exemplified using situations extracted from these two case studies.

3.3.2 Methodology Specification

The UI-Business Alignment Methodology proposes a strategy to map the core models (business process, task model, UI model) related to enterprise systems and build a network of links that supports traceability and impact analysis when changes are requested in any of these models.

The methodology has core actions to be performed by stakeholders involved in applying the methodology, which can include:

- *Business analysts*, who need to make sure that what is specified in BPs are done by system end-users;
- *UI designers*, who want to guarantee that what they design supports business processes;
- *Usability experts*, who want to adequate business operations into UI tasks that bring productivity to end-users;
- *Operation managers*, who want access to a high level impact analysis reports concerning business processes they manage;
- *Users*, who want their suggestions towards improving user interaction to be considered in the business context.

To present an overview of the methodology, we use a new formal graphical notation for modeling software development methodologies, the ISO/IEC 24744 [48] International Standard, a preliminary version of which is found in [33]. Although the underpinning meta model was standardized within the software engineering community, this notation is applicable to a wide variety of methodological domains, not only software but also hardware, systems and numerous engineering application domains. The use of this standard is motivated by the fact that a shared notation composed of graphical elements to represent methodologies aims at facilitating the communication between method engineers from different domains (e.g. SE, HCI, Business) or approaches (e.g. traditional, agile).

Many efforts have defined meta-models with concepts for software development methods. Software & System Process Engineering Meta-Model (SPEM) [118], the Object-Oriented Process, Environment and Notation, known as the OPEN Process Framework (OPF) [27], and ISO/IEC 12207 [46] have concepts that are useful to describe a software development process, while ISO/IEC 15504 and Software Process Improvement and Capability dEtermination for Object-Orientation (OOSPICE) focus on concepts to perform process assessment and improvement. Even though each of these standards define a coherent set of concepts useful for their intended domains; they are neither as *complete*, *generic* and *extensible* as desired so as to cover most areas, enable application in different contexts, and enable adding new concepts as needed [32]. Besides the lack of these properties, neither of them accepts the idea of method engineering, as done by ISO/IEC 24744.

The following diagrams were chosen to graphically represent information about the methodology from different perspectives (definitions of the diagrams and entities used in these diagrams are from [40], [32]).

Lifecycle Diagram represents the overall structure of a methodology, depicting both the temporal and the content aspects: method-level specification with stage elements (the elements used here are: Time Cycle, Phase, Milestone) and work unit elements (the element used here is: Process). *Stages* represent managed time frames within a project. A *time cycle* is a managed interval of time within a project with the objective of delivering a final product or service. A *phase* is a managed interval of time within a project for which the objective is the transition between levels of abstraction. A *milestone* is a managed point in time that marks some significant event in the project. *Work Units* represent jobs performed, or intended to be performed, within a project. A *process* is a large-grained work unit that operates within a given area of expertise within the project.

The Lifecycle Diagram (Figure 3.13) depicts the time cycle of the UI-Business Alignment Methodology organized in two phases (Network Building and Impact Analysis) with a milestone between them to denote the need to fulfill certain pre-requirements to go from one phase to the next. The 'Network Building' phase is composed of two processes (Forward Building and Backward Building) and the 'Impact Analysis' phase is composed of three processes (Forward Analysis, Middle-Out Analysis and Backward Analysis).

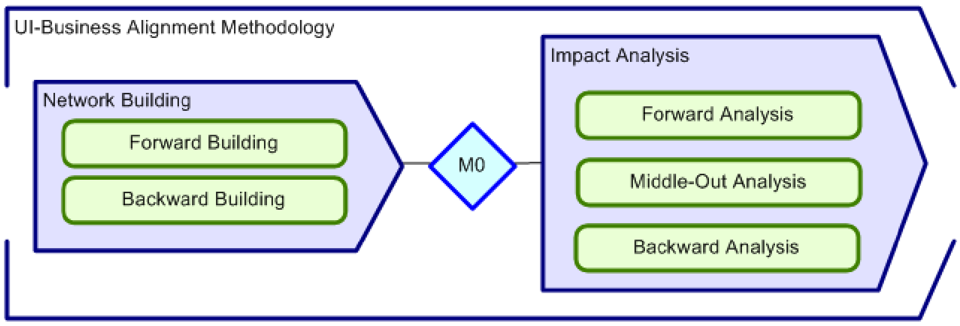

Figure 3.13: Lifecycle diagram of UI-Business Alignment Methodology

Process Diagram describes the details of a process or collection of processes used in a methodology. These details may include the relationships between Processes, the links between Processes and Tasks, and the Producers assigned to the processes and tasks. A *task* is a small-grained work unit that focuses on what must be done in order to achieve a given purpose. *Producers* represent agents that have the responsibility to execute work units. A *role* is a collection of responsibilities that a producer can take. A *Person* is the individual human being involved in a project.

A process diagram (Figure 3.14) for the set of processes for the 'Network Building' phase depicts the stakeholders (i.e. Business Analyst, Usability expert, UI designer) involved in these two processes. The 'Forward Building' process is composed of a set of tasks to be done when the organization has the BP modeled to start building the network of links from them. The 'Backward Building' process is composed of a set of tasks to be done when the organization has it enterprise systems, but not yet the BP modeled, so the network of links is built from systems, listing its UI components, creating task models and deriving the BP model.

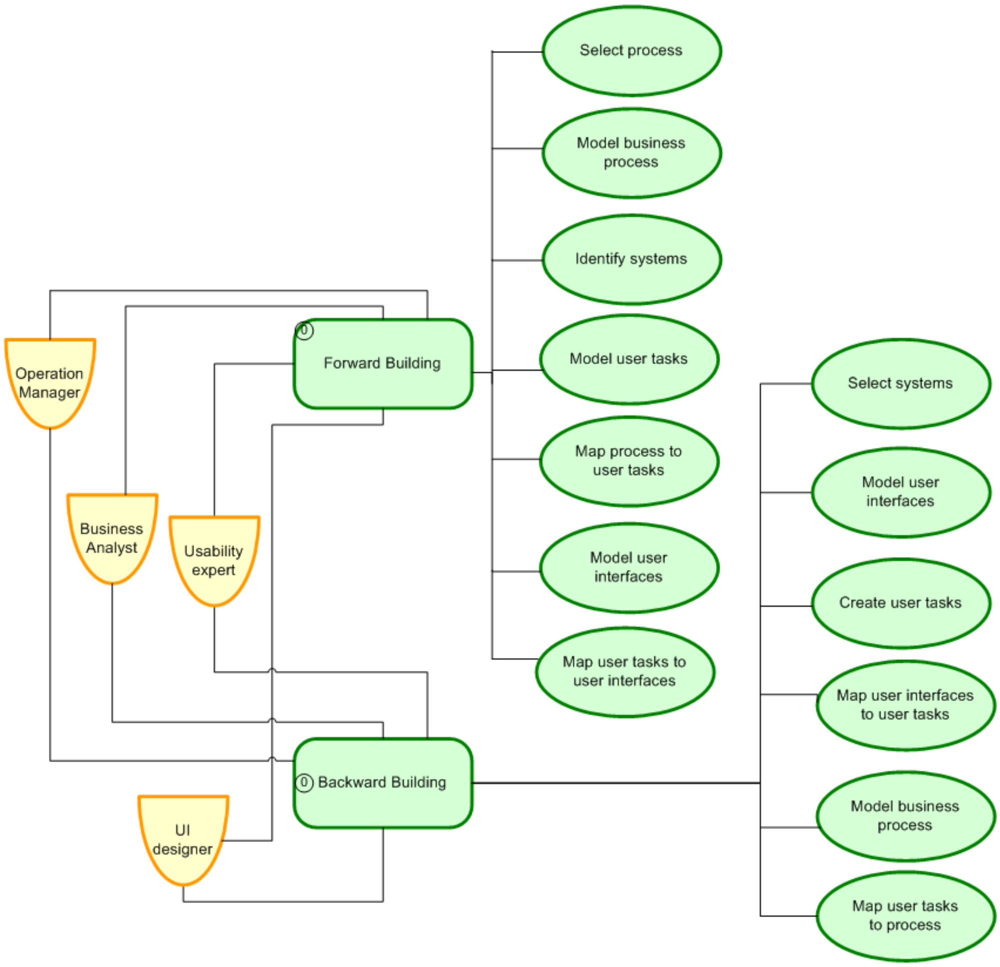

Figure 3.14: Process diagram for the creation of the network of links

A process diagram (Figure 3.15) for the set of processes for the 'Impact

Analysis' phase depicts the stakeholders (i.e. Business Analyst, Usability expert, UI designer, End-User) involved in these processes.

The 'Forward Analysis' process is composed of a set of tasks to check the impact of changes starting from the BP model. The 'Middle-Out Analysis' process is composed of a set of tasks to check the impact of changes starting from the task model. The 'Backward Analysis' process is composed of a set of tasks to check the impact of changes starting from the UI model.

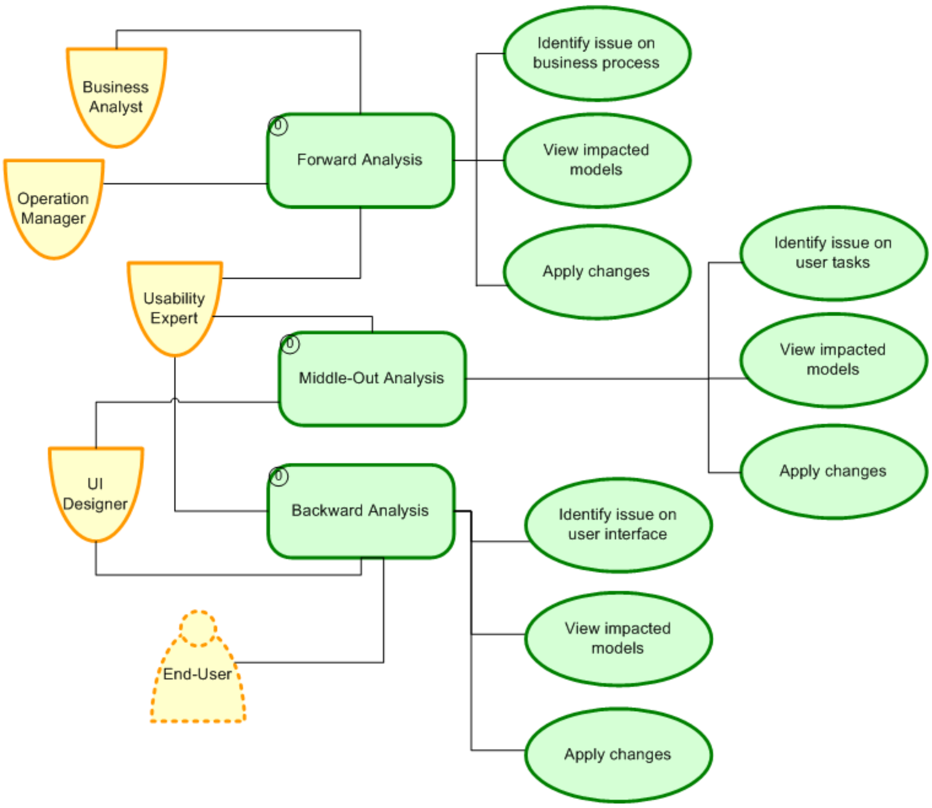

Figure 3.15: Process diagram for the impact analysis

Specification of Core Actions

The methodology is effectively adopted by starting with a critical process, which could be a great source of data to be analyzed in an initial project. After applying the methodology at the first time, a procedure is prepared to spread the methodology through other critical and non-critical processes considering the organizational context in other projects.

The core actions to create the network of links can be applied in two main approaches: forward and backward. Figure 3.16 depicts the alignment of business processes with UI components following these two approaches, in which the actions of stakeholders are organized in disciplines related to business process management methodologies and User-Centered Design methodologies.

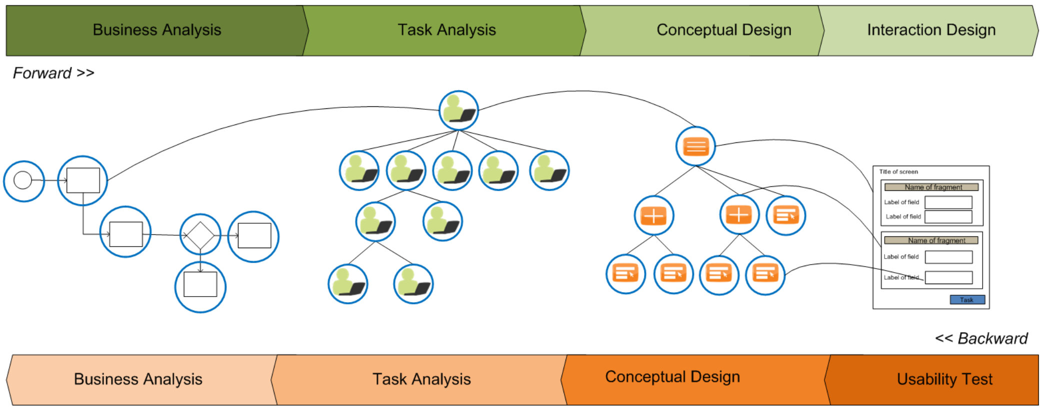

Figure 3.16: UI-Business Alignment in two approaches

The methodology considers the following knowledge areas (as named in bodies of knowledge [43]) or disciplines (as named in software development processes, as in the various instantiations of the Unified Process (UP) [49]):

Business Analysis is *"the set of tasks, knowledge, and techniques required to identify business needs and determine solutions to business problems. Solutions often include a systems development component, but may also consist of process improvement or organizational change."* [43].

Task Analysis *"is used to analyze the underlying rationale and purpose of what people are doing"* [105].

Conceptual Design means *high-level user interface design regarding the presentation structure that maps to interactive tasks* [70].

Interaction Design means *"designing interactive products to support the way people communicate and interact in their everyday and working lives"*, aimed for all kinds of technologies, systems and products. [105].

Usability Test means *"testing with real users to obtain direct information about how people use computers and what their exact problems are with the concrete interface being tested"* [82].

In the forward approach, business processes go through analysis and improvements can be identified that require changes on the business processes during "Business Analysis"; UI tasks are specified and updated during "Task Analysis"; UI components are specified and updated during "Conceptual Design"; and UIs are specified and updated accordingly during "Interaction Design".

In the backward approach, UIs are analyzed and improvements can be identified that require changes in UIs during "Usability Test"; UI components are specified and updated during "Conceptual Design"; UI tasks are specified and updated during "Task Analysis"; and business processes are specified and updated accordingly during "Business Analysis".

Forward Approach: In situations where there is a process and there is no system, the forward impact analysis can simulate what could be the future user interaction to execute the process; verify the alignment of the processes with existing interaction; and identify what impact the optimization of processes has on the user interaction.

The core activities of the methodology to guarantee the user interface-business alignment are listed as follows for a scenario application of the forward approach:

1. Business Analysis
   1.1. Select representative and critical processes of the organization
   1.2. Model business processes
   1.3. Identify systems related with those processes
2. Task Analysis
   2.1. Model UI tasks
   2.2. Map process activities with UI tasks
3. Conceptual Design
   3.1. Model user interfaces
   3.2. Map UI tasks with user interfaces
4. Interaction Design
   4.1. Design user interfaces
5. Impact Analysis
   5.1. Perform impact analysis and generate strategic information to analyze process performance and user experience.

Backward Approach: In situations where there is a system and there is no process, the backward analysis uses navigation patterns in the system as a source of data to exploit business processes. It helps identifying how the improved usability of UIs suggested by systems users (e.g. customers, employees) impacts BPs.

For a scenario application of the backward approach, the core actions are the following:

1. Usability Test
    1.1. Select representative and critical systems of the organization
    1.2. Observe users working with those systems to identify their tasks
2. Conceptual Design
    2.1. Model user interfaces
3. Task Analysis
    3.1. Model UI tasks;
    3.2. Map user interfaces with UI tasks
4. Business Analysis
    4.1. Model business processes
    4.2. Map UI tasks with process activities
5. Impact Analysis
    5.1. Perform impact analysis and generate strategic information to analyze the process performance and user experience.

After the mappings are created in any of these approaches, the impact analysis can be performed in forward, middle-out and backward approaches:

In the forward approach, when changes are made on BP models, they may impact UIs. Such changes can be done from a variety of reasons, including new or alternative ways of doing things, new business opportunities, organizational changes, new regulations; etc.

In the middle-out approach, changes are made on task models as suggestions from usability experts and UI designers. When changes are made first on task models, the impact is on BP models and UIs with possible updates.

Reasons for such changes include new ways to perform tasks that improve the user experience.

In the backward approach, changes are made on UIs as suggestions from systems users (e.g. customers, employees). When changes are made first on UIs, the impact is on BP models with possible updates. Reasons for such changes include defects to be fixed, better user understanding of the systems' features, new technology, etc.

The methodology can be applied in organizations with minor changes on the way they work, highlighting that the core actions are mandatory to guarantee UI-Business alignment, as depicted in Figure 3.17. It starts with BP models that can be done in any notation and specified in any level of granularity. The BP elements are transformed into BPMN in order to unify the business representation. The BP breakdown structure is associated with UI models through the hierarchical levels of task models. Each level of the BP is mapped with the different levels in the task models, then the task model is mapped with the UI model that has UI components ranging from composable to atomic components.

The methodology is flexible enough to enable stakeholders to specify at which level of granularity they will map the elements, depending on the complexity of the processes and systems, on the information available and other aspects that directly influence the impact analysis.

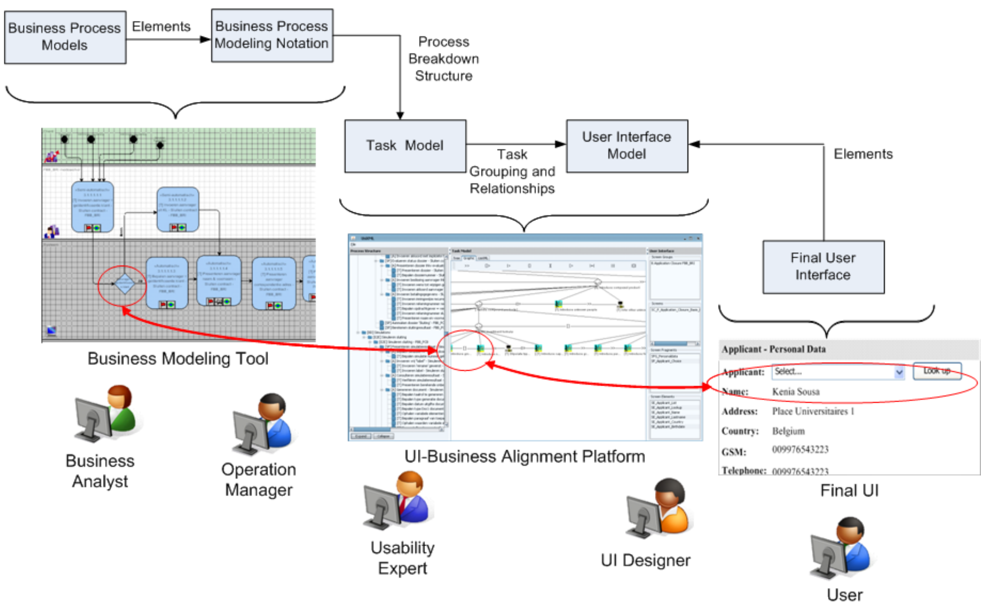

Figure 3.17: Outline of the method with roles, artifacts and tools

Business process models help to conceive task models since its most crucial parts of the processes can represent a high level specification of what users must do. Task models should be reviewed by UI designers or usability experts in order to make sure that the user perspective was taken into consideration. This review should be emphasized when a transformation is made. This review is done by creating relationships between UI tasks that are not considered during business analysis, but are pertaining to the user interaction, such as interruption of tasks that can be undertaken afterwards, disabling other alternative tasks when one task with higher priority is finished (e.g. form submission), optionality of tasks, etc.

Focus on business process performers

We rely on epistemological research to turn the recurring focus from business analysts and place it on business process performers. According to Giddens [31], agents are those who act in a world and are capable of making choices based on information and knowledge. Therefore, they reproduce the social structure and can bring about social change. Aligned with Giddens' duality of structure theory, structure contains agents' behavior, but agents' behavior also makes the structure itself possible. For this work, these agents are professionals responsible for performing activities of business processes and, consequently, they are also end-users of enterprise systems. Giddens' theory is well aligned with our framework, in which business process performers must adhere to the corporate structure following business processes, but, on the other hand, they also bring changes to such structure since they think and behave differently.

Concerning the first state of the dual structure, in order to be able to adhere to defined processes, process performers must understand well how to perform these processes. However, it is common to find business processes defined at a high level that does not represent the tasks they should perform in their daily work. That is an important consideration to have task models with a lower level of granularity to specify performers' tasks to have fewer deviations from the tasks assigned to business performers. That is a reason for the growing interest from organizations to know how specific agents, not generic roles, accomplish their work [137]. Therefore, we focus on actual users that we can identify within the organization since they are the ones who actually perform the process.

Looking at the second state of the dual structure, in order to enable process performers to bring changes to the corporate structure, they must play active roles by suggesting improvements on tasks defined at a low level of granularity, which is the manageable level of tasks corresponding to their work. Stoitsev et al. [119] has already highlighted that BPM has already perceived that the involvement of business users in process modeling can make processes better turned to users' needs and organizational changes, thus resulting in processes that are more consistent and still flexible.

Integration with Corporate Initiatives

This methodology can be combined with other corporate methods and processes, such as software development processes, HCI methods, process improvement methodologies, IT-Business alignment strategies, etc. The methodology is focused on disciplines, actions, roles and models directly related to UI-Business alignment. Other aspects are pertaining to the corporate software development process. For instance, the methodology considers one unit in the IT department: UI design, leaving aside other usual units, such as software architecture and development, which are directly related to code or architecture.

In a scenario in which the organization adopts the Rational Unified Process (RUP) [60], the core disciplines in the forward approach are performed according to the organizations' customization and the alignment methodology activity 1 is related to the Business Modeling discipline, and activities 2 to 5 are related to the Analysis & Design discipline.

In an agile context that adopts the Agile Model Driven Development [4], the core activities in the forward approach are performed according to the organizations' customization and the alignment methodology activities 1 and 2 generate outputs that serve as input to the Initial Requirements Envisioning, and activities 3 to 5 are related to the Iteration Modeling.

Following CMMi for Development [18], the core activities in the forward approach are performed according to the organizations' customization and the alignment methodology activity 1 is related to the Process Management process areas, such as Organizational Process Definition; and activities 2 to 5 are related to the Engineering process areas, such as Technical Solution. An association between our method and a standard software engineering approach is a good representation to highlight its importance to make professionals aware of the inclusion of a traceability method within their projects.

The Software Development Life Cycle as well as the UID Life Cycle are complex and unique in nature. The integration of the framework with SE and HCI techniques requires deep explanation of SE and HCI aspects, which would take the focus from the main contribution for UI-Business alignment. This general association with existing software development process techniques represents a proof of concept that has been further validated in two case studies.

In order to make our methodology feasible for application in the industry, we define it as an alignment methodology that can be used: 1) within a development method (e.g. UI development method, software development method) and 2) independently, such as the configuration management supporting process, which can be applied along with any primary life cycle that supports the entire software product life cycle, namely, development, operation or maintenance [46].

The view of the methodology depicted in Figure 3.16 shows the organization of the methodology in disciplines and actions, which is a pre-requirement to make this strategy applicable both in traditional and in agile approaches. In the first, the actions can be separated by roles, such as business analyst, system analyst, UI designer, interaction designer, developer, usability experts, ergonomist, etc. In the latter, there are generalizing specialists [4] who are multi-disciplinary professionals that can perform several activities in a same project. And it also enables people executing different roles to participate in the same activity (e.g. UI designers and usability experts participating in Task Analysis) as a team to obtain quick feedback about ideas.

Cross-Organizational Engagement

The forward approach stimulates the participation of managers and business analysts, who can identify prospective improvements on business processes to address new corporate strategies. The backward approach stimulates the participation of basically anyone in the organization, such as UI designers and end-users, who can identify improvements directly on the UIs of enterprise systems to address specific needs (e.g. facilitate interacting with the system).

In the backward approach, the reason for monitoring the process through user observation is that this solution is independent of how processes are designed and systems are developed. In most solutions in the market, business processes must be executed in specific engines and the system must be developed in a specific technology (e.g. web services) to allow collecting data as the system is used. Our solution is applicable in a wider range of corporate scenarios since we can monitor any kind of system that are representations of certain business processes, for instance, an order management system for provisioning products. In this approach, stakeholders do not need specific skills in order to be able to delineate which part of the system brings impediments for the successful completion of their goals.

The participation of professionals from different domains with a proactive approach towards process optimization makes this framework less obtrusive and enables *"social creativity"* as they create, design, develop or perform business activities. Social creativity is a solution for complex problems in which all stakeholders contribute with their different points of view and knowledge resulting in new ideas [28].

3.3.3 Building Network of Links

Building the network of links is the process of mapping the core models to create a network of links between them. It is explained and exemplified using scenarios from the two case studies at the bank sector and the telecommunications industry.

The first case study, applied in the financial sector, occurred in the business and IT departments with five professionals, who participated in several meetings and interviews during three months. The business process under analysis was related to insurance contracts, a major process for the organization. The second case study, applied in the telecommunications industry, occurred in the finance department and enterprise business department with twenty-three professionals, among managers, business analysts and IT professionals, who participated in meetings and interviews during four months. The business processes under analysis were related to accounting and provisioning products, where the first is a support process and the latter is a major process for the organization. More detailed information on the contexts of the case studies is presented in Chapter 5 Validation.

Even though both cases experienced the same issues concerning UI - Business alignment, they have different characteristics that are useful for different approaches as the network is being built. The bank case study is more appropriate to illustrate the forward approach because it already had its business processes well defined and modeled and their systems designed and developed. The telecommunications case study is more suitable to illustrate the backward approach since it did not have the business process modeled in the finance department.

Forward Approach

Using the bank sector from one of our case studies and considering the financial crisis in the global market, there have been several merger situations. Thus, major banks that have faced the need of process integration with other banks; are now, more than ever, struggling to identify inefficiencies in business process that may be significant cost drivers. They are also looking for ways to optimize the time between the process definition and maintenance and its effective implementation. In this bank case study, the bank is also in the insurance sector, as is the case of several major financial institutions.

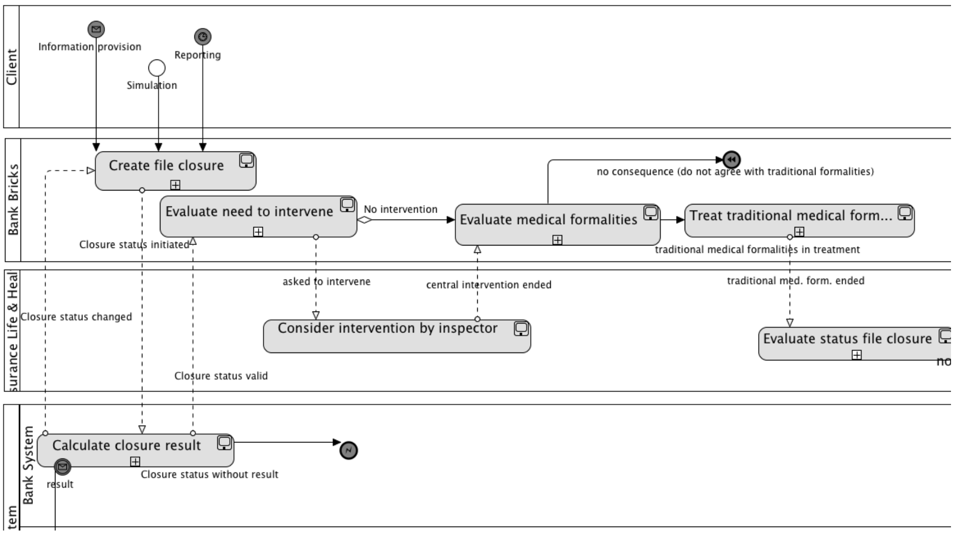

Figure 3.18: Business process for insurance contracts

The insurance sector has several business processes that define the services to be provided to customers aiming to attain organizational goals, especially customer satisfaction. The process for file closure, which creates insurance contracts, is illustrated using BPMN with a set of sub-processes in Figure 3.18.

In this process, first the contract file is created and then all the data checked and registered in the appropriate tariff calculations. When an intervention is sought because the calculations have not met pre-defined requirements, the contract as a whole is transferred to an inspector in the back office. The inspector is informed of the need to intervene via a telephone call. Then, the inspector will take the necessary steps to bring about a final decision from which the file is handed back to the bank agency for completion, after that, the intervention has officially ended and the evaluation of the medical formalities can start.

The bank already had their enterprise systems deployed in several branches in use by bank agents and back office inspectors. With access to the user manuals of the system with several screen shots and explanations, detailed in the interviews with the UI designers, it was possible to have an understanding of the UIs.

Once the possible list of interactive tasks was identified by looking at the UIs of the system, we specified the UI tasks in task models. The high level UI tasks were created based on the business activities, then these UI tasks were detailed based on the UIs. An extract of the task model is depicted in Figure 3.19, which is a visual representation of interactive tasks to create a contract file, without the task relationships. These UI tasks are related to the UI presented in Figure 3.20.

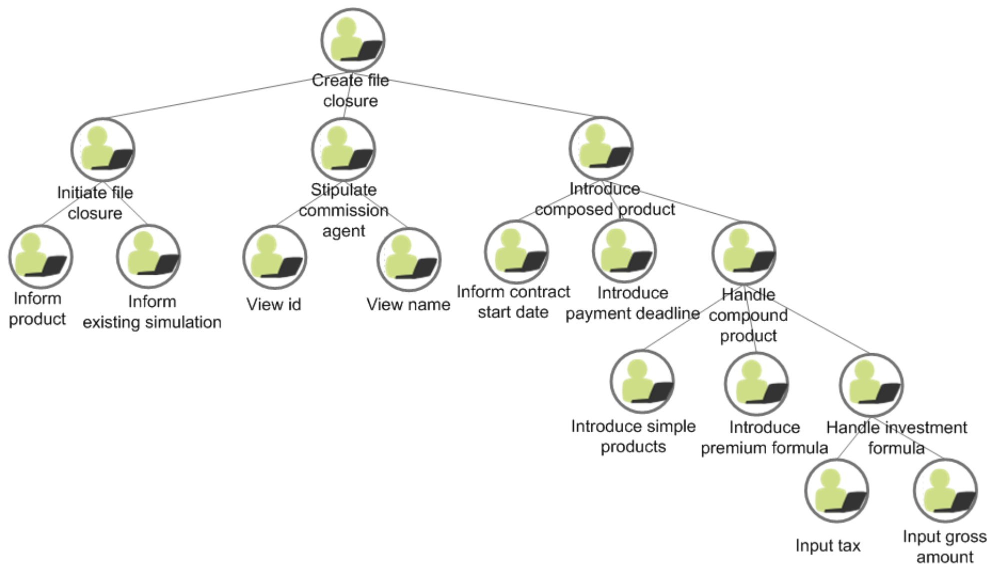

Figure 3.19: Task Model for File Closure

From the existing UIs, we extracted the UI model by identifying the UI components, as depicted in Figure 3.20. The screen for file closure is composed of six screen fragments containing: header with navigation and indication of the status of closing the contract, selection of the product, information of the commission agent responsible for this contract, general data of the contract, data on the type of investment and data on the type of savings. Each of these screen fragments is composed of screen elements and there are three screen elements at the bottom of the screen that enable saving, navigating or canceling the request.

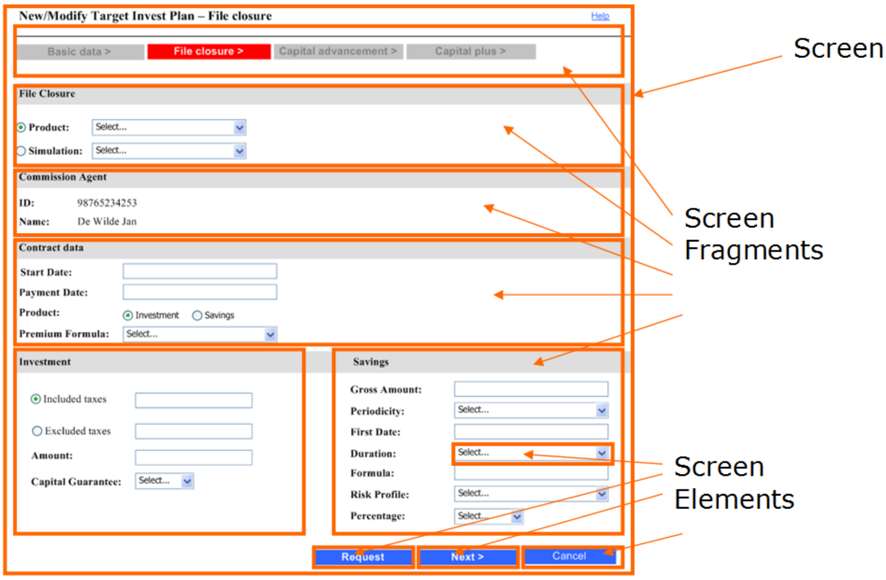

Figure 3.20: Extracting the UI model

With the models created, it was possible to map process activities with UI tasks and then map UI tasks with user interfaces. For instance, the sub-process 'Create file closure' was linked to the UI task 'Create file closure' and this task was linked to the screen 'File Closure', as depicted in Figure 3.21.

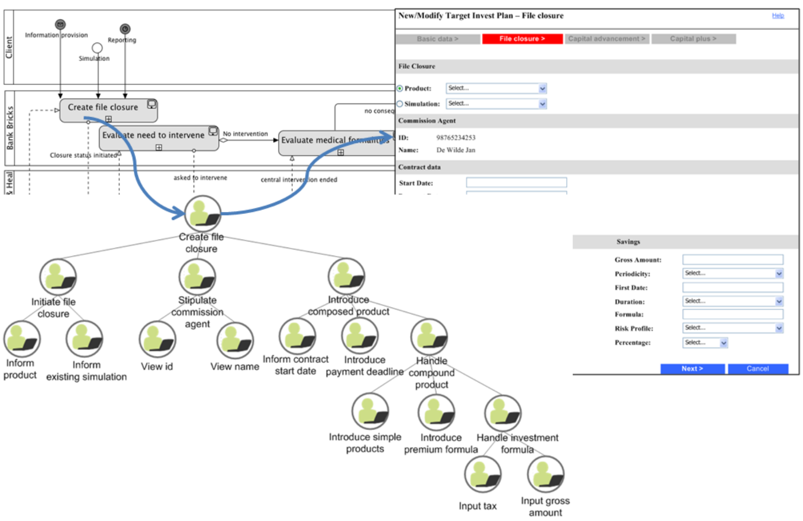

Figure 3.21: Model Mappings for File Closure

Backward Approach

Similar to many organizations that have enterprise systems, but have not yet defined their business processes, the telecommunication company has many departments in the same situation, thus the backward approach is useful to build the network starting with the system UIs.

Using the department in charge of the IT support for the Finance department as a first project in the telecommunications case study, the selected process is the accounts payable that controls invoices. An analyst in charge of the Finance IT support explained the process by using the enterprise system.

The part of the system presented here to exemplify building the network of links starts when a credit note is requested to correct the data in the invoice sent by the supplier that is different from the purchase order, either with price or quantity differences. If the approver requests a credit note, the credit note is created and sent to the requester who analyzes it for approval.

Figure 3.22 depicts the UI where approvers see the list of documents open for approval. When they select a credit note, the approver is able to approve, reject or cancel.

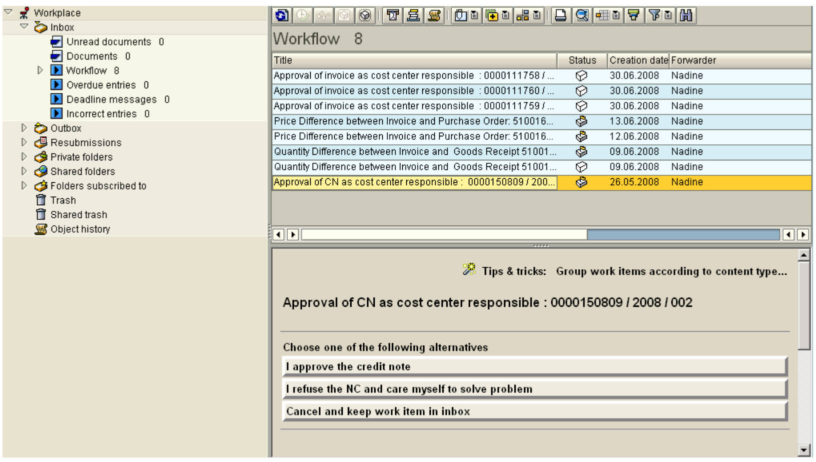

Figure 3.22: Screen for approval of credit notes

Based on the existing UI of the accounts payable enterprise system, we have specified the UI model for the UIs. Figure 3.23 depicts the UI model for the current screen group for approval of credit notes, composed of one screen for credit notes without purchase order and another screen for credit notes with purchase order.

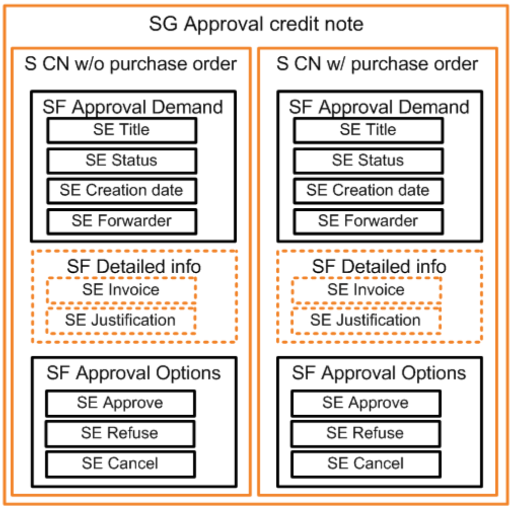

Figure 3.23: UI Model for approval of credit notes

Once the UIs were understood and the UI models created, we specified the UI tasks in task models. An extract of the task model for the UI of approval of credit notes is depicted in Figure 3.24, which details interactive tasks.

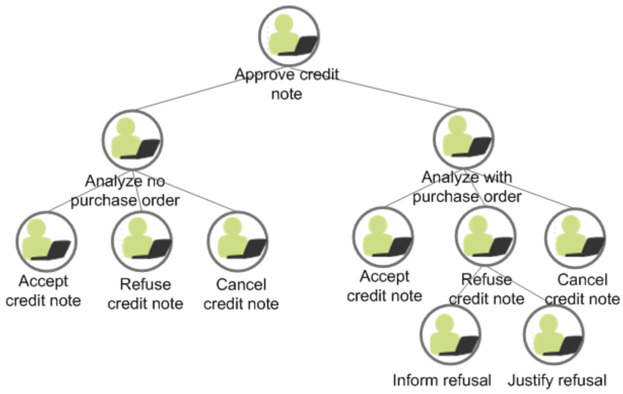

Figure 3.24: Task Model for approval of credit notes

With the details of UI tasks, it was possible to specify the high level activities of the business process for the approval of credit notes, depicted in Figure 3.25. In this process, when a credit note is open for approval, it is redirected to an approver, who can accept or refuse it. Once an action is chosen, the status of the credit note changes and an e-mail is sent to the finance department that processes the invoice and dispatches another e-mail with the latest status of the invoice and related credit note.

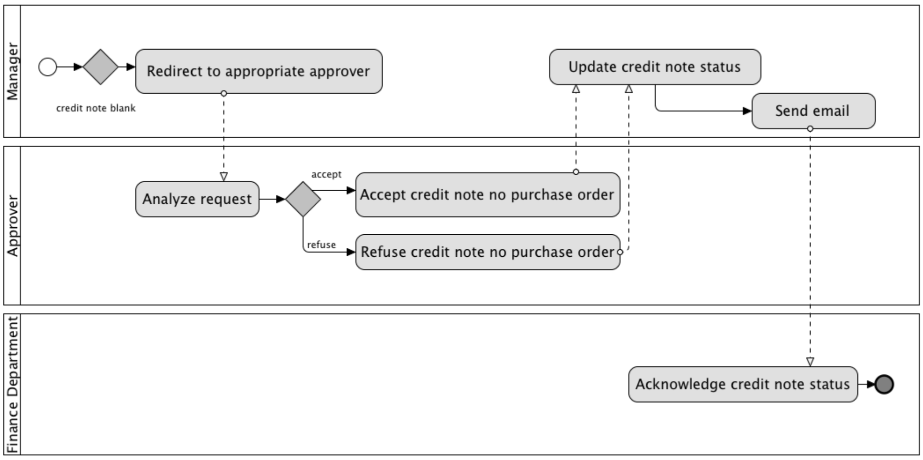

Figure 3.25: Business Process Model for approval of credit notes

With the models created, it was possible to map user interfaces with UI tasks and then map UI tasks with process activities. For instance, the screen 'Approval of credit note with purchase order' inside the screen group 'Approval of credit note' was linked to the UI task 'Analyze credit note with purchase order' and it was linked to the sub-process 'Analyze credit note with purchase order', as depicted in Figure 3.26.

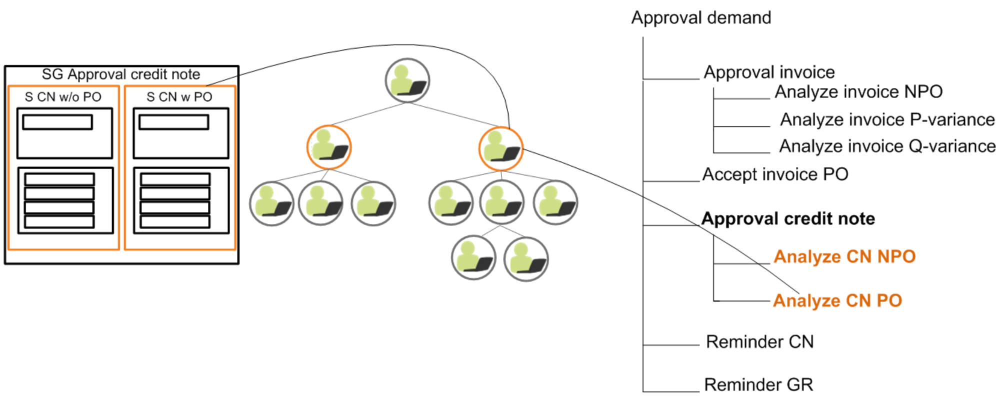

Figure 3.26: Scenario Support for Approvers

This example has illustrated how the models are created and how the associations between business process models, task models and UIs is done, which builds a network of links. With this network built, it is possible to navigate in the network in any direction, which aids in identifying the impact of changes, supported by the execution of rules, which are explained in the next chapter.

3.4 Applicability of the Framework

The framework is appropriate to be applied in the context of large organizations that have extensive enterprise systems and have their business processes or are at least interested in designing their business processes. To be able to apply the framework it is not required to have both. Having just the system is already a starting point since the framework enables to derive a business process model that can be evolved by business analysts.

In general, the stakeholders will be involved in designing business process models, evolving task models and creating UI models. We assume that the models created are correct, consistent and complete. They are complete considering that these models are designed in an iterative manner. Each iteration produces a version that is complete for the current context, but they are still open for changes and improvements. The platform is aimed to support stakeholders with information to help them maintain consistency between the models. However, the platform does not aim to indicate that a model is right or wrong because this is pertaining to the human abstraction skill. On the other hand, the platform indicates if a model is consistent or inconsistent, according to the conventions described in the rules. Being wrong may mean that it is different from reality or that it has a deadlock in the process flow, as an example of an error that stakeholders may make when modeling. As classified in [75], stakeholders normally fall in one of the seven sins of specification, such as over specification, contradiction and ambiguity. Therefore, it is an assumption that stakeholders are confident in creating the models and that they at least try to maximize the correctness, consistency and completeness of the models.

More specifically, for the framework to be applied in companies, the stakeholders involved shall be trained to update UI tasks derived from business process, delineate UI models and create the mappings between the models. Normally, we would expect usability experts to update UI tasks since they have an understanding of the UIs of systems and are used to work with textual representations of UIs, such as user manuals. Delineating UI models can be done by UI designers since it requires understanding the structure and navigation of the system UIs. Mapping the models can be done by stakeholders that understand both the business processes and the systems, such as business analysts. The business process is not considered as an added model to be learned by the stakeholders because it is considered as a given by the corporate setting. We assume the organization has decided to apply business process management techniques in their projects as part of their corporate strategy.

As in most methodologies, there is the need to consider the learning curve that represents the changing rate of learning how to update UI tasks, delineate UI models and create the mappings. Despite that, the task and UI models are simple, there is just a small set of elements to be learned. There are also the functionalities of the platform that make transformations from the business process into the task model, avoiding creating it from scratch and the plug-ins that enable evolving the platform with more support, such as creating a prototype of the UI. To quantitatively demonstrate that the application of the framework is worthwhile, a return on investment analysis can be applied for each project to check that the benefits of applying it outgrow its costs, thus proving its feasibility in a corporate setting and motivating stakeholders to continue applying it.

Besides the profit that the framework might bring from resulting improvements on business processes, the human perspective in this framework makes end-users to be heard in the company. The framework enables a free channel where they can give their opinion about user interaction. This opinion is translated into business processes so operation managers can ponder on their impact. When end-users' suggestions are put in practice and these changes improve their work, they are motivated to collaborate. The more an end-user's suggestions helps other end-users, the stronger is their reputation within the group (e.g. customer representative department) and higher is the trust between people.

3.5 Conclusion

This chapter presented a model-driven approach to map UIs and business processes. With this approach, models are mapped to each other and the user experience is considered in alignment with business needs.

If applied appropriately, this methodology can bring more benefits proportionally to the size of the system. The application on small systems could be problematic because it will add more complexity instead of keeping the work simpler, as preached by agile practices. On the other hand, large systems can benefit from it because of the activity volume and its respective screens, abstracted by models and their interrelationships.

This approach started to become consolidated by recent researches, which inspired us to specify a platform, explained in Chapter 4.

Chapter 4

Platform Specification

“”The user’s time is more valuable than ours. Respect it. Good UI design is humble.”, Jono DiCarlo, Developer, Humanized.

This chapter presents the definition of a software platform for the UI-Business Alignment Framework, which represents a support for the solution presented in Chapter 3. This platform is explained through the analysis of its structure, specification of its functionalities, design of prototypes to represent its behavior and definition of how models are managed using rules.

4.1 Introduction

To support the alignment between models, we envision the need for a software platform to map business models, task models and UI components in order to identify the impact of model changes. Therefore, we have developed the *UI-Business Alignment Platform*, called hereafter the *platform*.

This platform is a support for decision-making for the main stakeholders involved in applying the UI-Business Alignment methodology who want to track changes on BPs and UIs of enterprise systems. Even though there are *technical transformations* (e.g. BP to task model transformation), they are aimed to create mappings, they are not aimed towards generation of UIs, as explained in Section 3.2.3 of Chapter 3. There is also some degree of *interpretation* from the stakeholders, who consider their own values to make decisions when improving the transformed version of a model into a usable work product.

The UI-Business Alignment Platform is being developed in Java^{TM}, using:

- the Swing [128] library to implement the graphical UI thanks to its standard look and feel presentation in different platforms;
- StAX (Streaming API for XML) [121] 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;
- the Drools [50] rule engine to give the necessary flexibility to manage models;
- the Apache Derby [22] embedded database to store local and temporary data, simplifying the installation and distribution process;
- the Netbeans Visual Library [80] to render visual diagrams, such as task models and navigation models; and
- Jersey [51], the REST web services library to synchronize local data with a centralized repository, accessible by other people in the organization.

4.2 Functionalities of the Platform

The list of functionalities for the UI-Business Alignment Platform has been created based on the following sources: (1) the issues detected in the first case study demonstrated the need to correlate business process and UIs in one platform that integrates the models; the difficulties in doing impact analysis after changes showed us the need to supply a way to trace the elements and view impact of changes on elements; (2) the absence of concerns on user interaction in enterprise systems illustrates the need to generate task models based on business processes, associate task models with UI components, and view navigation between UIs related to certain activities of business processes; (3) the experience in the second case study illustrated new scenarios in which the support of a software platform brings more efficiency to show the results from applying the methodology; (4) the core actions defined in the methodology drive the specification of functionalities for the platform.

There are three core elements in our strategy that are necessary to provide UI-Business alignment: business process, task model and UI components. Adding to that, organizations, systems and end-users are also central elements that help identify the location of UIs and who are impacted by changes on these UIs, respectively. These six elements are related to the first six functionalities. The other functionalities are related to mapping models and supporting decision making when changes are necessary or have already been done in such models; and related to regulating the rules that manage the models. Rules are used in this platform to enable updating the meta-models. Whenever the meta models are updated, rules that manage their underlying models must be updated accordingly.

The use of this platform is done according to the methodology presented in chapter 3 UI-Business Alignment Framework. While using the platform, it is first needed to import the XML of a business process into the platform. The business process is a model external to UsiXML and any other UIDL and it can be created using any available process modeling tool. Most business process modeling tools are able to export their models into an XML format, which is appropriate to interchange information and communicate with other tools or systems.

When the models are imported into the platform (or even created within it), it is time for associating the models. One way to associate the models is done by business analysts requesting the generation of a task model. The transformation of business process models into task models is done through the mappings between the models, as presented in section 3.2.3, which are the foundation for a set of rules, to be presented in following sections. Their association is automatically created as the transformation is executed. However, there is also the scenario in which transformations between models are not performed; rather models are manually created, thus, making it necessary to make manual mappings between them. After an initial transformation, the generated task model can go under refinements to improve the specification of the user interaction. UI components are listed based on the screens of the enterprise systems for the selected process and manually associated with UI tasks using a drag-and-drop technique. Based on this overall specification of the platform, the list of functionalities is presented as follows:

Import Business Process - Import the business process diagram from business process modeling tools that provides the possibility to exchange models through the use of standards. A tool that allows exporting business process diagrams into the XML format enables interoperability with our platform. The traceability life cycle starts when the business process diagram is imported into our tool and can be viewed in the tab in the platform allocated to present the business process, which shows the processes, sub-processes and tasks organized in a hierarchical structure. This functionality represents the foundation for the alignment because the business process model is one of the starting points of the alignment.

Generate Task Model - A business process model is transformed into a task model by applying rules using as foundation the association of elements from business process and task models. Mapping these two models is possible through semi-automatic transformation of business process into task model. The generated task model is presented in a task model tab with different views: a hierarchical view similar to the business process model; and a graphical view that allows updating the task model with a user point of view.

List organizations - Identifying the organization and its sub-divisions where the framework is being applied, including units, departments, etc.

List systems - When specifying UI components, the system(s) in which the UI components are allocated must be specified. This functionality is important when the organization has several systems (e.g. billing, loans, payments, investments, and insurance) that should be specified to facilitate identifying where the UI components are located.

List UI components - UI components are listed according to how the system is structured or is to be structured, keeping consistency with the task models. The different UI components can be listed manually as in Figure 4.1.

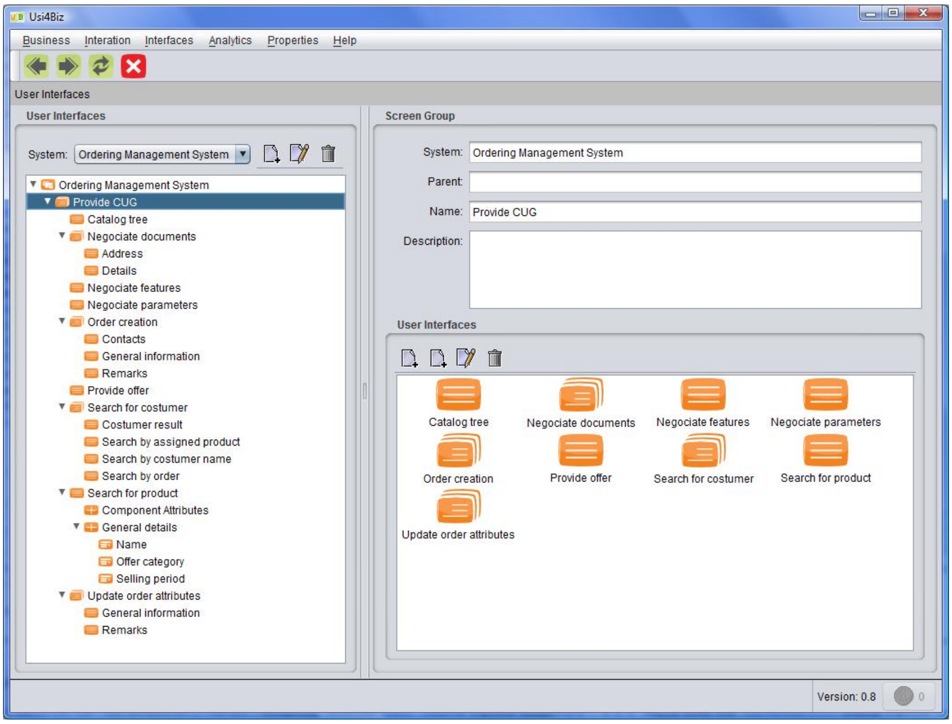

Figure 4.1: Prototype of the List of UI Components

List users - List the users of the systems. Any change on the business process models impact the way these users do their work, and consequently how the users interact with the updated systems' UIs.

Associate UI tasks with UI components - Creation of manual mappings between task models and UI components to create the network of links. Each UI component that has been listed is associated with a UI task in the task model by dragging the UI component and dropping it in the desired UI task.

Impact analysis - When any element is selected from either a business process model, task model, UI components, it is possible to see, for the selected element, which are the related elements on the other models that would be impacted if a change is to be made on the selected element. The traceability between these models is demonstrated by navigating in the network of links, so when a node in a model is selected, the traceability lists what is related to it in the other models.

Figure 4.2 depicts, on the left of the platform, the business process shown in a tree structure, where one activity is selected. On the middle, there is the task model with impacted UI tasks marked with a different color. On the right, there are the impacted UI components marked with a different color. This trace is created through the mappings between the models. This functionality provides clear visualization of information needed to make an impact analysis and decide whether or not a specific change will be made.

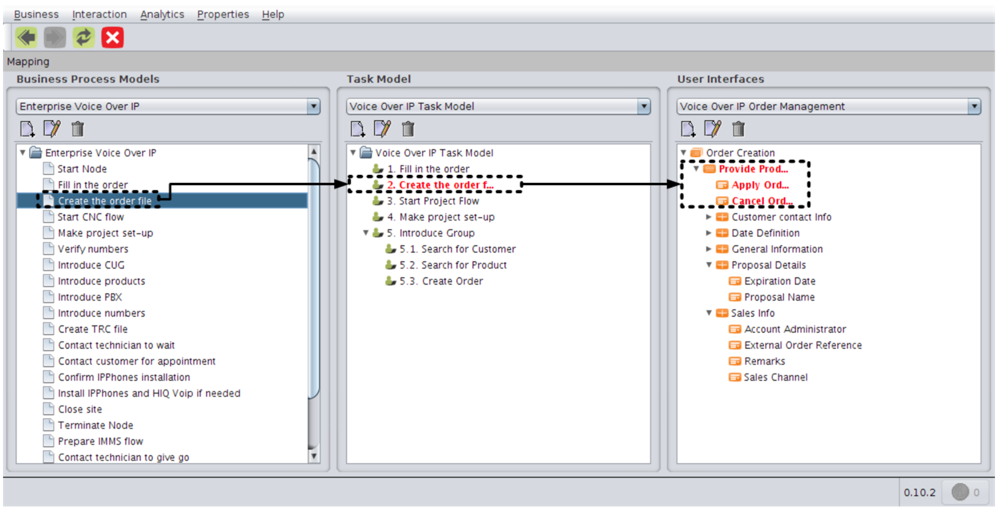

Figure 4.2: Prototype of the Impact Analysis

Generate impact report - A change can be performed on: a) the business process model, which is detected as a new version of this model is imported; b) the task model, which is done directly on the task model; c) the UI components, importing a new list of components or manually changing the components. Each detected change in any of the models is subject to analysis and approval. When a change is done in any of these models, a set of rules are executed in order to generate an impact report that serves as a support for decision-making. After the impact analysis is seen in Figure 4.2, the impact report lists all the elements on these models that have been impacted by the changes and the proposed changes, which can be applied directly or with a confirmation of the decision-makers. This functionality is primordial as a direction towards actions that need to be taken to maintain UI-Business alignment after changes have been done on any of the models.

Manage rules - The rules are defined according to the impact of each model element on the network of links. Given that, there is a set of rules for each model element. New rules can be created or existing rules can be updated according to the structure of the models (meta model) adopted by the organization.

4.3 Rule-Based Model Management

An expert system approach based on production rules provides flexibility to the platform since it enables updating the rules whenever there is a change on how the models are structured and managed, without the need for direct maintenance of the platform. This approach makes this framework adaptable to different organizational contexts by accepting that organizations may adapt their strategies to model processes and design user interfaces.

Rules support a rich representation of first-order logic to describe general knowledge about a certain domain [103]. The domain has a knowledge representation, whose elements are linked through formal symbols to compose rules as a collection of propositions [14]. To define the knowledge of an expert system, it is necessary to specify the objects that are important to the domain, the properties of these objects and the relationships among them, and assertions about these objects, composing a *working memory* that is constantly changing during the operation of the system [14]. The format of our working memory is a graph with links among the model elements, forming the *network of links*. Working memory elements are *objects*, that is, instances of these model elements. These objects are of different types, represented by the *entities* of the models, such as BP activities, UI tasks, UI components, among others depicted in the class diagram in Figure 3.3.

The rules are created using these entities and they adapt to different organizational context that adopt different ways to represent the structure of their business, task and UI models. Once rules are adapted to the organizational context, they can be applied in any application domain, as it has been demonstrated already in the bank and telecommunications sectors as two examples of domains. Because the way models are structured and managed might change in different business domains, rules bring more flexibility to the platform. However, it is important to point out that these rules differ from business rules that are representations of the logic of business operations. The rules hereby defined regulate changes on the models.

These rules have been defined based on the links between the models, according to the model mappings presented in Section 3.2.3 Model Mapping. For each object updated in a BP model, there is a rule that verifies if it has a related object in the task model and creates it, when necessary; the same happens from task models to UI components and backwards. 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 [76]. To support this, Model-driven development provides the traceability opportunity facilitating the creation and maintenance of relationships between models [2].

The basic properties of transformation rules are [55]: *tunability*, the possibility to adapt the transformation by parameters; *traceability*, the possibility to trace one element in the target model back to its causing element in the source model; *incremental consistency*, information added manually to the target model is not overwritten if the transformation is executed again (e.g. after the model is updated); *bidirectionality*, transformation applied from source to target and vice-versa.

As a contribution for this research work, we have defined a set of rules applied throughout the UI-Business alignment framework that adheres to these properties [117]. Figure 4.3 depicts the execution of the rules in the three main approaches: forward, backward and middle-out.

In the forward approach, when changes are made in the forward approach, rules are executed from the business process to the task model, persisting the changes on the task model, even if the task model does not yet exist, thus deriving a task model from the business process. Then, the rules are executed from the task model to the UI model, persisting the changes on the latter, assuming the UI model has been created beforehand.

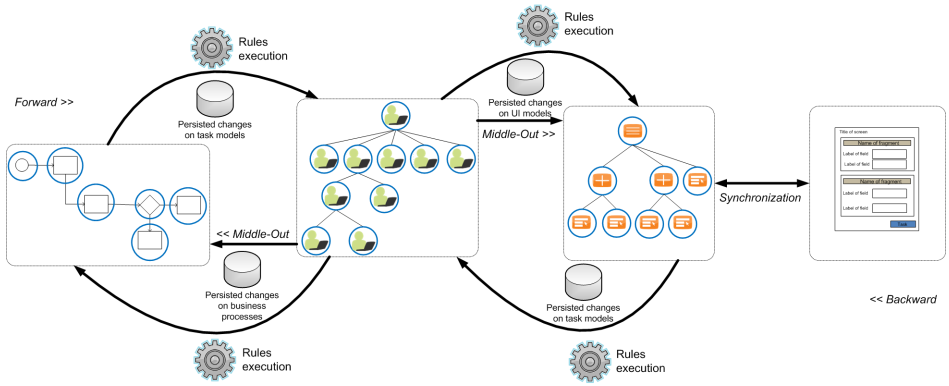

Figure 4.3: Automation of the Framework

In the backward approach, rules are executed from the UI model to the task model, persisting the changes on the task model, assuming the task model has been created beforehand. Then, the rules are executed from the task model to the business process, persisting the changes on the latter, even if the business process model does not yet exist, thus deriving a business process from the task model.

In the middle-out approach, the rules are executed from the task model to the business process, persisting the changes on the latter, even if the business process model does not yet exist, thus deriving a business process from the task model. Then, the rules are executed from the task model to the UI model, persisting the changes on the latter, assuming the UI model has been created beforehand. There are no rules specified from the UI model to the actual UIs of systems. There is a synchronization to indicate what has to be updated in the UIs directly by UI designers. The set of rules defined here are seen as sufficient for the goal of maintaining consistency between the models because they consider the entities existing in the meta models. They are no longer consider as sufficient when the meta-models are updated. Thus, new rules can be created and existing ones can be updated when any meta model is changed.

The rules specified here follow the general format of production rules with a two-part structure with an antecedent set of conditions and a consequent set of actions, according to [14]:

```
IF  conditions THEN actions
```

The body of the condition is a tuple of this form:

(type attribute$_1$*: specification*$_1$ *... attribute*$_k$*: specification*$_k$*)*

where a specification is one of the following: an atom, a variable, an evaluable expression, a test, the conjunction, disjunction, or negation of a specification.

The possible actions in the consequence part of the rule are: ADD (i.e. a new object is added in the working memory); REMOVE i (i.e. remove from the working memory the object that matched the i-th condition in the antecedent of the rule); MODIFY i (i.e. modify the object that matched the i-th condition in the antecedent of the rule by replacing its current value of *attribute* by the *specification*).

To perform these actions, it is necessary to verify if the object matches specific conditions. For that, there are three main conditions that are checked, which are if the object has been recently created, updated or deleted. The object is understood as recently created when the id is empty, which means that an object exists but still does not have its id persisted. The object is understood as updated when the id is different from empty, which means that the object already exists with a valid id. The object is understood as deleted when the id is null, which means that the object does not exist anymore.

An example of a rule is presented as follows, which verifies if there is an object of type `activity` that has been recently created (i.e. with an empty id), then add to the working memory an object of type `UITask`.

`IF (activity id:` $\{=\emptyset\}$`) THEN ADD (uiTask)`

The execution of these rules follows a cycle of three steps [14] that uses the rules from a rule catalog, as follows:

- *recognize*: find the rules whose conditions are satisfied in the current working memory (e.g. rules for a BP activity);
- *resolve conflict*: among these rules recognized in the first step, choose the one that should be executed (e.g. rules that process the addition of a new BP activity that is not yet related to any UI task);
- *act*: perform the consequent actions of the rules selected in the second step to change the working memory (e.g. add the new UI task for this added BP activity).

For each entity of the models, there is a set of rules that verifies the integrity of the models when an event happens on any object. An *event* is any action that changes the state of the object, such as create, update, delete, and other types of operations performed in the models, as listed in

Table 4.1. Performing an event on any given object triggers the execution of rules for that object. Performing an operation on multiple objects means that there is one event for each operation, thus, the set of rules are triggered for each updated object. For instance, adding ten BP activities in the BP model means that there are ten different events that trigger rules related to BP activities ten times.

For the rules to be executed, a working memory is created and the object being handled is passed to the working memory, along with other objects related to the handled object. The rules are executed when changes are made on the objects of a specific entity, and as each object is tested, if the condition is true, the rule is processed. Processing these rules means changing another type of object that triggers the execution of rules for that entity. This cycle continues until the conditions tested in the rules are false, which means that the models are supposed to be consistent at this point. These rules are organized by entity because any change in an object requires verifying the integrity of the entire network of links.

To exemplify the execution of rules for an event that happened in a BP activity, consider an event that changes the name of a BP activity; this event triggers the rule set for the entity BP activity that checks the conditions of all rules. The working memory receives instances of the updated BP activity and all instances of UI tasks related to this BP activity. For those rules concerning the change of the object's name, these rules are executed, thus changing the name of UI tasks associated to the updated BP activity. The change on a UI task is an event that triggers the rule set for UI tasks; the rules addressing the conditions are tested and executed, thus changing the name of UI components associated to the updated UI task. The change on a screen element is another event that triggers the rule set for Screen Elements and the rules addressing the conditions are tested. At this point, this updated screen element might have an impact on another UI task; which changes this UI task that triggers the rule set for UI tasks, continuing the cycle. However, the test results may be false, which means the network of links is consistent and the rule execution stops.

For each event triggered, the condition in the rules should be complete enough to help in resolving conflicts and also to avoid repeating the execution of the same rule more than once, which brings more agility to the process of propagating changes and maintaining consistency among the models. Another aspect that makes the conditions stronger is verifying if the object is not yet related to objects in the other models since, for instance, when a new BP activity is added, the assertion is that a UI task still needs to be created and associated to this BP activity. For that reason, the condition also verifies if this newly-added BP activity does not yet have a UI task related to it, before adding a new UI task.

These events are classified in operations that represent actions performed on any model element. Actions on BP elements result on actions that must be executed on task models and then on UI components, as listed in Table 4.1. The intent is to consider different types of operations that can change these elements and execute the existing rules at design time to maintain the alignment, exemplified with one rule for each operation.

The rules have been assessed through the association of: BPMN core elements (e.g. flow objects, connecting objects, artifacts, swimlanes) with UI task relationships (e.g. associate sequence flow with enabling); BP activity attributes with UI task properties (e.g. associate conditional flow with optional task); and process task type with UI task type (e.g. associate user task with interactive task), as presented in section 3.2.3. Such an assessment enabled the identification of similarities in the semantics of these two notations and the different aspects that may indicate a prospective need of adaptation in one of them. For example, a Data-Based Exclusive Gateway represents that only one of the various paths can be taken and it can be represented as a deterministic choice between UI tasks, but for this representation to be complete, it is necessary to allocate a condition for each target UI task in the task model, not only a condition in the relationship between two UI tasks because each target activity in a gateway can have a different condition. In addition, such associations depict a many-to-many mapping between BPMN and task model elements as depicted in the association of both BPMN "subprocesses" and "tasks" to "UI tasks" in the task model; and the association of BPMN "sequence flow" to both "enabling" and "enabling with information passing" in the task model, depending on the condition type of the sequence flow; etc.

A fine-grained granularity level adopted when mapping the models expresses details in the results of the impact analysis. These rules in the forward approach result in changes on: task models, which represent an initial version that is yet to be evolved by UI designers/ usability experts to consider grouping UI tasks, detailing tasks relationships and defining specific UI tasks that are intrinsic to user interaction; and UI components, which represent an abstract structure of what needs to be created by UI designers. These rules are not applied to automatically generate UIs since this methodology considers that UI designers create UIs for enterprise systems, as argued in Section 3.2.3. The result of these rules in the backward approach are changes

Table 4.1: Classification of operations

Operation	**Definition**	**Action on BP**	**Change on task model**	**Impact on UI components**
Add	Include element	Create element	Create TM element	Create UI component
Delete	Remove element	Remove element	Remove TM element	Remove UI component
Modify	Change attributes of element	Change type of element	Update TM element	Update UI component
		Change name of element	Change name of TM element	Change name of UI component
Order	Arrange elements	Change sequence of activities	Change sequence of tasks	Change sequence of UI component
Merge	Unite elements	Remove element	Remove TM element	Remove UI component
		Change type of element	Update TM element	Update UI component
		Change name of element	Change name of TM element	Change name of UI component
Split	Divide or separate elements	Change type of element	Update TM element	Update UI component
		Change name of element	Change name of TM element	Change name of UI component
		Create element	Create TM element	Create UI component
Decompose	Separate into constituent elements	Create element	Create TM element	Create UI component

on: task models, as specified in the previous approach; and business process models that represent a version to be reviewed by business analysts considering business strategies and corporate goals. The result of these rules in the middle-out approach are changes on: business process models that represent a version to be reviewed by business analysts; and UI components, considered by UI designers as a list of proposed changes on the impacted UIs. With these rules, we can analyze when a node is changed, what should be done in the other models to maintain consistency (e.g. when a screen is deleted, identify the related UI tasks and BP activity that are no longer supported).

There are many rule engines available on the market and open source as well. Most of them can be embedded in applications, allowing them to execute their own rules and get benefit from the decision-making features. These rules have been written using the Drools rule language because of various advantages of its a declarative style solution [8]:

Easier to understand: It is easier even for non-developers, such as a technically skilled business analyst to understand, verify and change rules than to use a programming language code because of the rules consistent representation.

Flexibility: Thanks to this formalism, it is much easier to add new rules, modify or remove existing rules than to change or rewrite an application.

Reasonable performance: The Rete algorithm that is behind Drools, in theory, makes the performance of the system independent of the number of rules. Because a rule engine is essentially a generic if-then statement executor, there are numerous performance optimizations that can be applied independently of the rules.

Reusability: Rules are kept in one place, which means easier reusability.

Independent Lifecycle: With Drools, the rules can be authored, deployed, versioned, managed, and so on, independently from the rest of the application.

Embedability: Drools can be easily embedded into existing applications in order to implement just a section of a system.

The Drools rule definition starts with the rule name; the condition and consequence sections follow. Drools keywords are *rule*, *when*, *then*, and *end*.

```
rule ‘‘name of rule’’
when
    conditions
then
    consequence;
end
```

The condition defines the patterns that the rule matches with. The consequence is a block of code that is executed when all of the patterns within the condition are matched.

The rule condition can declare variables of an Object type. The variable name starts with a $ symbol, it can be declared up front to be used later in the rule. The condition verifies if the Object field matches a certain value, which can be of different types, such as predefined variable, string, regular expression, date, boolean and more. A condition can have multiple field constraints joined by additional keywords, such as an implicit *and* (i.e. a comma), *or*, *not*, *exists*, etc.

$\$variable : Object\ (field_1 == value_1,\ field_k == value_k)$

In the rule consequence, Drools has convenience methods to be executed as simple actions: modify: to update existing facts in the working memory; insert to insert new facts in the working memory; retract to remove existing facts in the working memory.

Considering that each entity of the models has a set of rules that verifies the integrity of the models when an event happens on any object, we add the rule set name in the beginning of each rule definition. The rule set has the name of the entity to which it is related. The name of the rule follows this format: impact on <<element>> when <<operation>> <<element>>.

An example of a rule in the Drools rule language is presented as follows, which verifies when there is an element of type activity that has just been created (i.e. with a field id that is empty), then add to the working memory an element of type UITask related to this activity.

```
ruleset ‘‘BP Activity’’
rule ‘‘impact on UITask when adding BP Activity’’
when
    $activity: Activity(id == "")
then
    insert(new UITask($activity));
end
```

To make the rules understandable by a larger range of stakeholders involved in managing the rules, it is productive to use human-readable rules using *Domain Specific Language* (DSL) [56]. A DSL is a computer language that is focused on a clearly definable problem domain, rather than a general purpose language that is aimed at any kind of software problem. A DSL file specifies the translation of sentences from the problem-specific terminology into rules. In Drools, a DSL file maps DSLR sentence (human-readable rule) to a DRL rule (rule in the Drools rule language). A simple DSLR file can be written as follows to express when there is an element of type `activity` that has just been created (i.e. with a field id that is empty), then add to the working memory an element of type `UITask` related to this activity:

```
ruleset ''BP Activity''
rule ''impact on UITask when adding BP Activity''
when
    The Activity does not have id
then
    create UITask for activity;
end
```

The DSL file can be written as follows to map the sentences to the rules, written in the following format:

```
[<scope>] [<Type>] <language expression> = <rule mapping>}:

   [condition][] The Activity does not have id =
   $activity:  Activity(id == "")
   [consequence][] Create UITask for activity =
   insert(new UITask($activity));
```

To demonstrate how the alignment is achieved, the first rule is presented both in natural language and in the Drools rule language for demonstration; the other examples are presented using Drools human-readable language, using DSLR for clarity. DSLR is useful to demonstrate an important reason for using rules; which is that rules can be customized by the organization. Therefore, the organization can update and create new rules according to their approaches adopted for BP modeling, user experience, UI design, among other aspects.

These rules are explained and exemplified using scenarios from the two case studies. The scenarios from the bank sector are used for the forward and middle-out approaches and the scenarios from the telecommunications industry are used for the middle-out and backward approaches, illustrated in the following sub-sections.

4.3.1 Forward Approach

In the forward approach, we demonstrate how changes on BPs have impacts on task models, and changes on task models have impacts on UI components. For that, we use the context of customers requesting insurance contracts, from the case study in the bank sector [112]. In this context, when aiming to increase customer satisfaction and optimize the performance of enterprise operations (with a more efficient service through faster responses), business analysts decided to allow customers to follow the status of their requests through an on-line system.

From Business Process to Task Model

Scenario - Follow request status: *Business analysts suggest that customers could follow their request status.* In the BPs for insurance contracts, in order to enable customers to follow the status of their requests for insurance, business analysts have created a new sub-process called "Follow request status". This sub-process is composed of tasks informing that customers can login into the insurance contract on-line system and then view the request status.

For each of the newly created business tasks and their associations in the sub-process, we look at a couple of elements that have been created. With the creation of this new sub-process, mainly flow objects (e.g. tasks) and connecting objects (e.g. sequence flows) have been created. For these elements, rules for FlowObject and Connecting Object, respectively, are executed to create the equivalent elements in the task model.

For a BP sequence flow that is added, the working memory receives an instance of the updated BP sequence flow and of task model elements related to it. A set of rules related to BP sequence flow are triggered and its conditions are tested. The rules that positively address the condition that verifies if a sequence flow was created are executed, thus creating a new enabling relationship in the task model.

To exemplify, the following rule is executed, which expressed in natural language means that: "When a sequence flow is created in a BP, then it results in creating an enabling relationship in the task model with the same information as set for the sequence flow (including which are the source and target tasks linked to it)":

```
ruleset ''Sequence Flow''
rule ''impact on Enabling when creating BP Sequence Flow''
when
    $seqFlow: SequenceFlow(id == "")
then
    insert(new Enabling($seqFlow);
    modify($enabling){
           $enabling.setSource($seqFlow.getSource())
           $enabling.setTarget($seqFlow.getTarget())
     };
end
```

Scenario - Request intervention: *Business analysts suggest that bank agents could enter intervention within the contract request.* Business analysts have also decided to improve the internal communication so the customer can acknowledge the status updates more efficiently. They have decided that the bank agents do not need to request interventions through a different form; it is enough to enter the reasoning of the decisions within the insurance contract request. The sub-processes "Evaluate need to intervene" is directly impacted by this strategic decision.

The sub-process "Evaluate need to intervene" needs to have the BP task "Enter applicant intervention" removed since there is no longer a specific form for intervention requests. Figure 4.4 shows the deleted task from the sub-process and its equivalent UI task in the task model that needs to be deleted. This figure shows the task model evolved with an interactive task, not relevant for the business process, but important for user interaction. When the BP is changed and the task model is regenerated by executing the rules, this extra UI task must remain to guarantee incremental consistency; reason for adopting an incremental transformation approach [55], which updates the changed elements with the newly generated ones, keeping the extra elements and the unchanged elements untouched.

To delete this UI task and its relationships in the task model, rules for the deleted elements (i.e. BP Task and BP Connecting Object) are triggered. For a BP Task that is deleted, the working memory receives instance of the deleted BP task and of the UI tasks related to it. A set of rules related to BP task are triggered and its conditions are tested. The rules that positively address the condition that verifies if a BP task was deleted are executed, thus deleting the related UI tasks in the task model. The following rule is executed for a deleted BP Task:

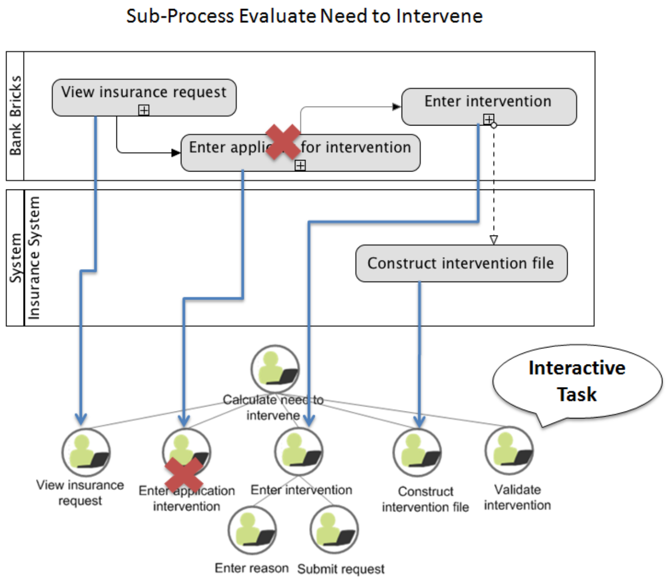

Figure 4.4: Impact of deleted UI task on the task model

```
ruleset ''Task''
rule ''impact on UITask when deleting BP Task''
when
    The Task has a null id
    There is a related UITask
then
    remove UITask;
end
```

Scenario - Enter intervention: *Business analysts suggest that inspectors could change a larger set of applicants' data.* The sub-process "Consider intervention by inspector" has its tasks updated to allow inspectors to change a larger set of applicants' data, not just the value for the insurance premium. This way, they can also change the type of insurance and the charged taxes depending on the applicant's income (Figure 4.5).

To add more detail in a BP task, it is necessary to modify it into a sub-process. The updated information is sent to the working memory with the instance of the BP sub-process and of the UI tasks related to it. A set of rules related to BP sub-process are triggered and its conditions are tested. The rules that positively address the condition that verifies if a BP sub-process has been modified are executed, thus updating the related UI tasks in the task model from a simple task to a parent task composed of sub-tasks with the same ones contained in the sub-process. The following rule is executed for an updated BP Task:

```
ruleset ''SubProcess''
rule ''impact on UITask when updating BP SubProcess''
when
     The SubProcess has an existing id and
     The UITask does not have subTasks
then
     modify UITask to create subTasks
            with subActivities from SubProcess;
end
```

Scenario - View intervention requests: *Business analysts suggest that inspectors could see all incoming intervention requests with access to detailed applicants' data.* Since inspectors have access to change a larger set of applicants' data, they also need an easier access to detailed applicants' data. To enable this, the sub-process "Consider intervention by inspector" is updated to allow inspectors to view more detailed information of the applicant and insurance (e.g. applicant name, data file, insurance file and date of request) for every new intervention that arrives without the need to previously search for it. Figure 4.5 depicts the tasks updated in a new sequence and its equivalent updated UI tasks in the task model.

To change the sequence of BP activities, the main object that is impacted is the BP Connecting Object because it makes a reference to the source and target activities related to it. In this case, the connecting object that is being used is the sequence flow. Thus, rules for BP Sequence Flow are triggered.

For a BP Sequence flow that is updated, the working memory receives an instance of the updated BP sequence flow and of the UI task relationship related to it. A set of rules related to BP sequence flow are triggered and its conditions are tested. The rules that positively address the condition that verifies if a BP sequence flow is modified are executed, thus updating the related UI task relationships in the task model with a new source and target tasks. The following rule is executed for an updated Sequence Flow:

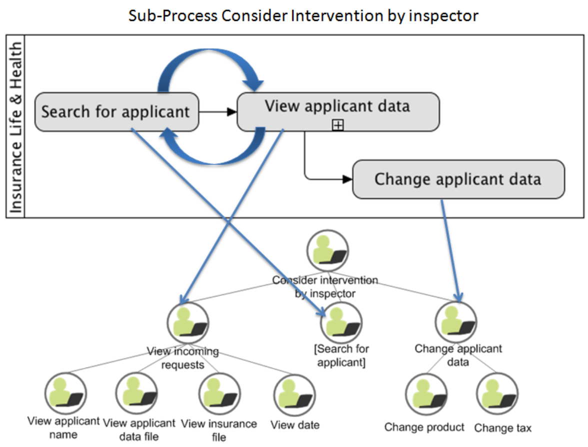

Figure 4.5: Impact of updated UI tasks on the task model

```
ruleset ''Sequence Flow''
rule ''impact on Enabling when updating BP Sequence Flow''
when
    The SequenceFlow has an existing id and
    The Enabling relationship has source and
        target UITasks different from the SequenceFlow
then
    modify Enabling to update source and target UITasks
           with source and target tasks from SequenceFlow;
end
```

From Task Model to User Interface Model

For elements changed on task models, there are possible impacts on UI components, which should be updated accordingly. When rules are executed to maintain consistency in the models, updated UI components represent advices of possible changes to be done on UIs by UI designers/usability experts.

Scenario - Request intervention: *Business analysts suggest that bank agents could enter intervention within the contract request.* Continuing this scenario, for the UI task "Evaluate need to intervene", when the UI task 'Enter applicant for intervention' and its relationships in the task model were deleted, the screen 'Enter Intervention' and screen element 'Request intervention' also had to be deleted.

When the UI task and its relationships are deleted, rules for UI Task and UI Task relationships are triggered. For a UI Task that is deleted, the working memory receives the instance of the deleted UI task and of the UI components related to it. A set of rules related to UI tasks are triggered and its conditions are tested. The rules that positively address the condition that verifies if a UI task was deleted are executed, thus deleting the related screen and screen element. The following rule is executed for a deleted UI task:

```
ruleset ''UITask''
rule ''impact on Screen when deleting UI Task''
when
    The UI Task has a null id and
    There is a Screen related to it
then
    remove Screen;
end
```

An important aspect is to consider the element that calls this removed screen. The screen element "Request intervention" is a button on the previous screen "File Closure" that calls the screen "Enter intervention". This screen element is associated to the enabling relationship of the deleted UI task "Enter applicant for intervention". Therefore, since the relationship is deleted, this screen element is also removed through the following rule. This rule guarantees that if there are any elements out of the deleted screen, but that are also associated to the deleted UI tasks and its relationships, that they will also be removed so there is no broken links on the UIs.

```
ruleset ''Enabling''
rule ''impact on Screen Element when deleting Enabling''
when
    The Enabling has a null id and
    There is a Screen Element related to it
then
    remove Screen Element;
end
```

Scenario - Enter intervention: *Business analysts suggest that inspectors could change a larger set of applicants' data.* Continuing this scenario, when a single UI task was changed into a parent task, it still had just a screen element related to it and it needed a screen composed of screen elements in order to support the new sub-tasks.

The updated information is sent to the working memory with the instance of the UI task and of its related sub-tasks and screen element. A set of rules related to UI tasks are triggered and its conditions are tested. In this case, we search for a rule that satisfies the conditions of an updated UI task with sub-tasks that only has a screen element related to it. The rule that positively address this condition is executed, thus creating a new screen (i.e. screen Change Financial data) with screen elements using the names of the new sub-tasks (e.g. Change product, Change tax). The following rule is executed for an updated UI task:

```
ruleset ''UITask''
rule ''impact on Screen when updating UI Task''
when
     The UITask has an existing id and has subTasks and
     There is a ScreenElement related to it
then
     create Screen for the UITask;
     modify Screen to create ScreenElements
            with subTasks from UITask;
end
```

Scenario - View intervention requests: *Business analysts suggest that inspectors could see all incoming intervention requests with access to detailed applicants' data.* Continuing this scenario, for the UI task "Consider intervention by inspector", inspectors view all incoming requests then decide to either search for a specific applicant or directly change the data of an applicant among the incoming requests. In this scenario, the order of UI tasks has been changed. Figure 4.6 depicts the updated UI tasks and its equivalent updated UI.

Changing the sequence of UI tasks, the main object that is impacted is the UI Task relationship because it makes a reference to its related source and target UI tasks. For an Enabling relationship that is updated, the working memory receives an instance of the updated relationship and of its related UI components. A set of rules for the Enabling relationship is triggered and the rules that positively address the condition that verifies if an Enabling relationship is modified are executed since the order of two UI tasks (i.e. 'Viewing incoming requests' and 'Search for applicant') has been changed, thus updating the UI components related to these UI tasks (i.e. two screen fragments 'View incoming requests' and 'Search for applicant'). The following rule is executed for an updated enabling relationship:

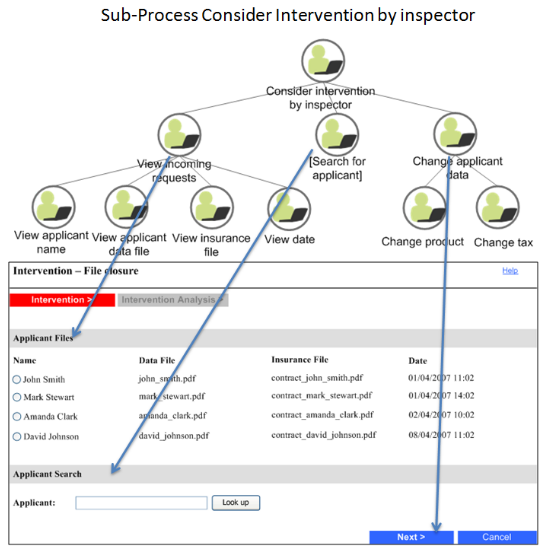

Figure 4.6: Impact of updated UI tasks on UI

```
ruleset ''Enabling''
rule ''impact on Screen Fragments when updating Enabling''
when
    The Enabling relationship has an existing id and
    The ScreenFragment has source and target
          UITasks different from the Enabling relationship
then
    modify ScreenFragment to update source and target relations
           with source and target UITasks from Enabling;
end
```

To summarize with a general overview of changes, the process 'Closure' was updated to provide transparent information for customers and more productivity with internal communication. This change has resulted in updating four sub-processes for a core product in the bank. Changes on these sub-processes were processed in the task models that were adequately evolved to consider the user interaction. The updated task models were then processed in order to indicate exactly which UIs components (linked to UI tasks) were impacted from such changes.

The impact analysis resulted in suggesting the following changes:

From the Scenario Request intervention: 1) The screen 'File Closure' has now a new screen fragment 'intervention' composed of two screen elements to allow bank agents to inform the reason and to do the request in the same screen, therefore; 2) There is no need for the screen ‘Enter Intervention’ with a specific form that added bureaucracy;

From the Scenario View intervention requests: 3) In the screen ‘Consider Intervention’, its screen fragments were updated by changing their order.

From the Scenario Enter intervention: 4) The screen ‘Change data’ is created because it is needed to support the new screen elements that gave more flexibility to inspectors; and

From the Scenario Follow request status: 5) The screen ‘Follow Request’ was created in the insurance contract on-line system to allow customers to follow the status of their requests;

4.3.2 Middle-Out Approach

The middle-out approach adds another level of flexibility when using this framework because it enables stakeholders to suggest improvements in terms of user experience. In this approach, the impact analysis starts with the analysis of UI tasks and shows the impacts on the business processes and UIs. This approach analyzes if the envisioned improvements in terms of user experience can be applied according to the business expectations and UI design decisions.

There are two scenarios presented in the middle-out approach. The first scenario is from the bank sector and the second one is from the telecommunications industry.

Scenario - Request second review: *Usability experts suggest that customers could request a second review of their insurance request.* In the bank context, after some weeks testing the updated systems, usability experts suggested changes on UI tasks. One suggestion was to enable customers to request a second review and send extra documentation (e.g. work contract, bank statement, salary statement) to prove specific financial data in order to improve the type of insurance they may receive.

To address this request, they suggested adding a UI task related to managing insurance requests. Once this UI task is added, the working memory receives an instance of the new UI task and of UI components related to it. A set of rules related to UI tasks are triggered and the rules that positively address the condition that verifies if a UI task was recently created (e.g. ‘Request Second Opinion’) are executed, thus creating a screen for this new UI task (i.e. Screen ‘Request Second Opinion’) with the execution of the following rule:

```
ruleset ‘‘UITask’’
rule ‘‘impact on Screen when creating UI Task’’
when
    The UITask has an empty id and has sub-tasks
then
    create Screen;
end
```

Once the UI components have been updated, the BPs are also updated accordingly. For each added UI task in the task model, rules were also executed to add the equivalent BP activities.

```
ruleset ‘‘UITask’’
rule ‘‘impact on Sub Process when creating UI Task’’
when
    The UITask has an empty id and has subTasks
then
    create SubProcess;
end
```

There are though some situations in which one task model element is mapped to several BP elements because of the many-to-many nature of the mappings. For instance, if two sub-tasks are first linked with ‘Concurrency’ relationship, then this operator is changed to an ‘Enabling’ relationship in the task model; the consequent change in the UI is that the updated UI should reflect the sequence by organizing the related UI components in two screens or in two screen fragments that are navigable as a wizard, depending on the style guide adopted by the organization. For this same example, the consequent change in the BP can be one of the following options: sequence flow, default sequence flow, message flow, association and directional association. As a solution, one of them is selected as default and used in the rules to execute the transformation, and then business analysts may review the resulting BP to make any necessary updates. This highlights the importance of using rules that can be customized by the organization according to their modeling approach, style guide, and other corporate choices.

Scenario - Products with multiple sites: *Usability experts suggest that users could group licenses of customers with multiple sites to better see where the product is installed.* In the telecommunications context, the selected business process to represent the middle-out approach is the provisioning of the Integrated Telephony Solution (ITS) products of the enterprise

Voice over IP (eVoIP) solution for large enterprises that want to integrate their data and voice traffics on the same data network. This scenario treats the category of products that are determined by the number of sites (location of customer's installation addresses) that can be provisioned.

Products installed in enterprise customers with multiple sites were previously associated with a one-way connection. However, it was not possible from a specific customer site to determine neither the main customer site nor the other sites that belong to the same customer. Therefore, the usability experts proposed users first to create a group of licenses before ordering the requested products. This group of licenses creates a two-way connection between the group and the customer sites that facilitates identifying in which sites the customer has the product installed.

This scenario requires splitting the UI task 'Provision eVoIP' in two UI tasks: 'Introduce client group' and 'Introduce products' (Figure 4.7). This change leads to splitting the screens in two different screen groups and the sub-process in two sub-processes.

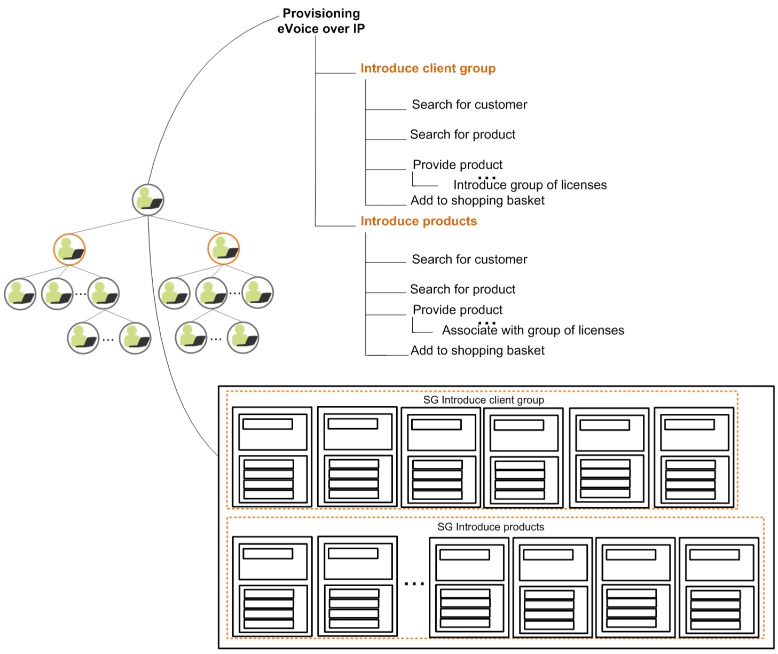

Figure 4.7: Scenario Products with multiple sites

Splitting UI tasks, new UI tasks are created to reorganize the split subtasks. For the new UI tasks that are created, the working memory receives instances of the new UI tasks and of its related UI components and related

BP Tasks. The conditions that are verified in this case are: if there is a new UI task with sub-tasks that are related to screens, but this new UI task is not related to a screen group. A set of rules related to UI tasks are triggered and the rule that positively address this condition is executed, thus creating a new screen group for this new UI task and associating it with the screens related to the existing UI sub-tasks.

```
ruleset ‘‘UITask’’
rule ‘‘impact on Screen Group when splitting UI Task’’
when
     The UITask has an empty id and has subTasks
     The ScreenGroup has a null id
then
    create ScreenGroup for the UITask;
    modify this ScreenGroup with the association of
           the screens related to the subTasks;
end
```

Once the UI components have been updated, the BPs are also updated accordingly. For each added UI task in the task model, rules were also executed to add the equivalent BP activities.

```
ruleset ‘‘UITask’’
rule ‘‘impact on Sub Process when splitting UI Task’’
when
     The UITask has an empty id
     The SubProcess has a null id
then
    create SubProcess for the UITask;
    modify this SubProcess with the association of
           the UITasks related to the subTasks;
end
```

4.3.3 Backward Approach

The UI-Business alignment framework also adheres to the property of bidirectionality, which means that the transformation can be applied not only from source to target but also backwards from target to source [55]. The backward approach is applied when users of enterprise systems suggest improvements on UIs or point out issues that interfere with the progress of their work. This opens a new channel for business process improvement that can start with actual users (bank employees using the system at the bank agency or customers using electronic system on the web at home) who aid to increase productivity within the organization.

The selected business process to represent the backward approach is the provisioning of the Integrated Telephony Solution (ITS) products of the enterprise Voice over IP (eVoIP) solution for large enterprises that want to integrate their data and voice traffics on the same data network. This is the same process used for the middle-out approach, but in this case, we consider two categories of products that are determined by the quantity of dialing numbers (number of the physical phones) that the customer requires and by the number of sites (location of customer's installation addresses) that can be provisioned. Each number may be associated to a package that represents voice service packages, such as call center, auto-attendant, receptionist, etc.

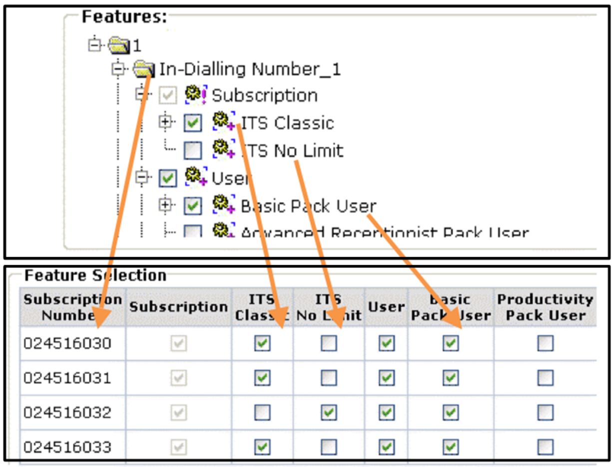

Figure 4.8: Repeated User Interfaces to Configure Numbers

Scenario - Configure numbers. *Users suggest to be able to select voice service packages in only one screen.* During tests sessions, users complained about doing twice the same thing on different UIs (Figure 4.8). On the first screen, they have to choose the voice service packages for each dialing number, and on the other screen, they have to check if the service packages previously chosen remain the same after the attribution of real numbers to the selected products. The change proposed is to merge these two UIs in one screen that is more productive to users.

The UIs that are similar are called 'Negotiate features' and the 'Update features'. Merging these UIs is done by selecting the most relevant UI components. This will decrease the number of UI tasks, which have to be deleted from the task model, and to be consistent, the related activities in the BP will be deleted as well (Figure 4.9).

When two screens are merged, what happens is that a new screen is created with existing content. The working memory receives an instance of the new screen, of its pre-existing screen fragments and the UI tasks related to these screen fragments. The conditions that are verified in this case are: if there is a new screen with screen fragments, and this screen is not yet related to any UI Task, but its screen fragments are already related to UITasks. A set of rules related to Screen are triggered and the rule that positively address this condition is executed. The execution of this rule creates a new UI task for this screen and creates sub-tasks for this UITask using the information from the screen fragments.

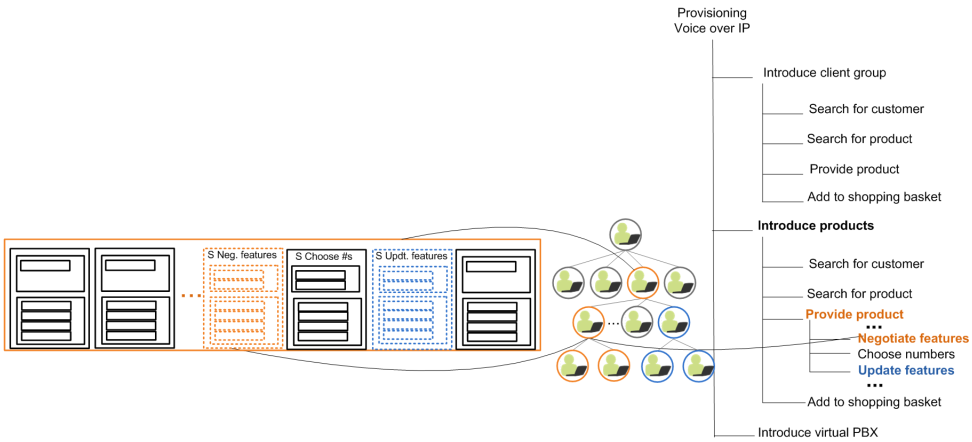

Figure 4.9: Scenario Configure Numbers

```
ruleset ''Screen''
rule ''impact on UI Task when merging Screens''
when
 The Screen has an empty id and does not have related UITasks
 The ScreenFragments have related UITasks
then
    create UITask for the Screen;
    modify this UITask with the association with sub-tasks
           related to the ScreenFragment;
end
```

To delete the screens that have been merged, since a new screen is already substituting them, the working memory receives the instance of the deleted screen and of the UI tasks related to it. A set of rules related to screens are triggered and the rules that positively address the condition that verifies if a screen with related UI tasks was deleted are executed, thus deleting the related UI tasks through the following rule :

```
ruleset ''Screen''
rule ''impact on UI Task when deleting Screen''
when
    The Screen has a null id and
    There is an existing UITask
then
    remove UITask;
end
```

Scenario - Licenses overview. *Users suggest to be able to see the overview of available licenses for the customer as they request the customer product.* When negotiating the features of product for customers with multiple sites, the information of licenses available for that customer does not appear on the screen and users have to remember the types of licenses and the quantities they have previously introduced in the group of licenses.

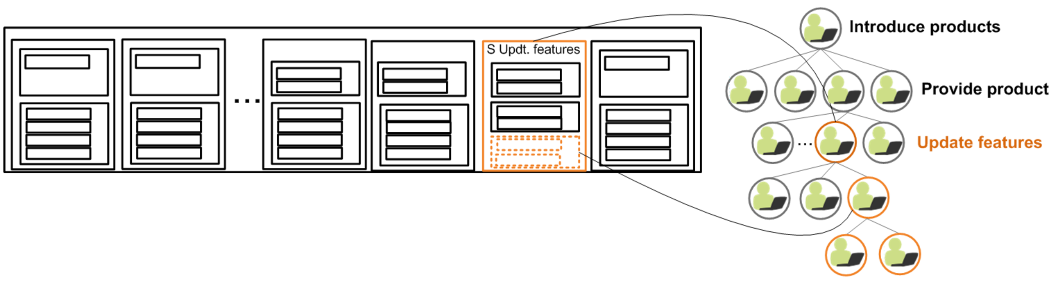

Figure 4.10: Scenario Licenses Overview

As a result, it is necessary to first check (and update) the group of licenses before provisioning the products. Most users forget and end up with an error message that they do not understand (for instance, the error message does not explain that more licenses are required for that specific group of licenses), which stops the process. Some users remember that they have to check the licenses; they navigate through five UIs, select each type of license to see the quantity ordered and write the information on a piece of paper; which is error-prone as well as a loss of time and efficiency. Introducing a product lasts, on average, twenty minutes; therefore taking into account such a scenario would decrease the time and increase the efficiency of users.

Adding a new screen fragment will create new UI tasks to view licenses information, but will not impact the BP (Figure 4.10). In this case, the change is done until the task model because it is only persistent to user interaction, which is not modeled in the BP.

For a new screen fragment that is added, the working memory receives instance of the new screen fragment and of its related UI components and UI tasks. A set of rules related to Screen Fragment are triggered and the rule that positively address the condition that a new screen fragment has been added in an existing screen is executed. This rule creates a new UI task for this screen fragment and creates its parent task related to the UI Task of its screen:

```
ruleset ``Screen Fragment''
rule ``impact on UI Task when creating Screen Fragment''
when
    The Screen has an existing id and has related UITask
    The ScreenFragment has an empty id
then
    create UITask for the ScreenFragment;
    modify this UITask with the association of
           the parent task related to the Screen;
end
```

The same happens with each screen element created within this new fragment with the execution of a rule that positively address the condition that a new screen element has been added in an existing screen fragment. This rule creates a new UI task for this screen element and creates its parent task related to the UI Task of its Screen Fragment:

```
ruleset ``Screen Element''
rule ``impact on UI Task when creating Screen Element''
when
    The ScreenFragment has an existing id and has related UITask
    The ScreenElement has an empty id
then
    create UITask for the ScreenElement;
    modify this UITask with the creation of
           the parent task related to the ScreenFragment;
end
```

The resulting UI to 'Negotiate features' including the proposed change supports the user with relevant information about the licenses. To achieve that, a new screen fragment is added on the bottom left of the screen and new UI Tasks are created to support it (Figure 4.11).

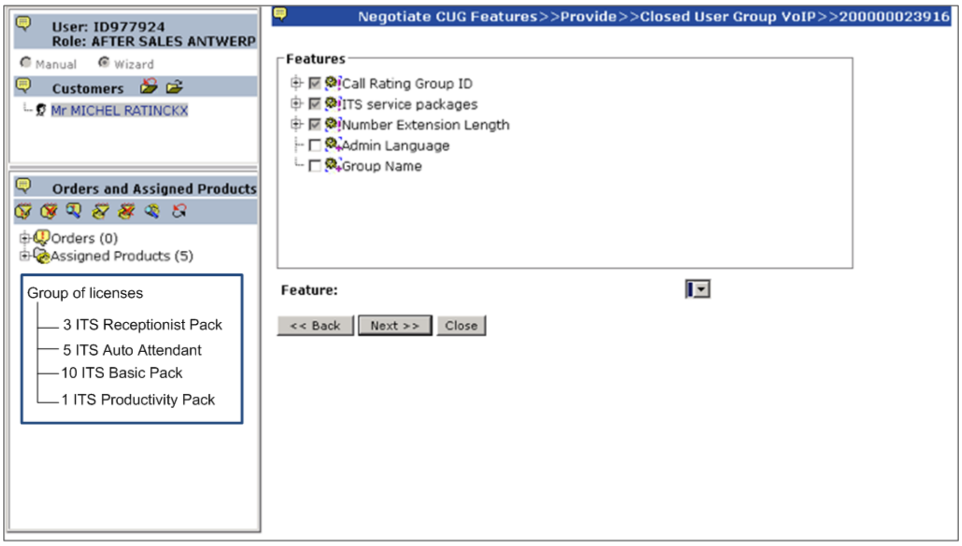

Figure 4.11: Suggested UI for License Overview

4.4 Software Analysis and Design

This section presents the result of the analysis of the software platform for the UI-Business Alignment Framework by depicting the structure of the software with a class diagram and the behavior of the software with an activity diagram using UML (Unified Modeling Language) models [30].

4.4.1 Software Structure

The structure of the software depicts what must be in the software being modeled and the UML class diagram is a static structure diagram that shows the classes, their attributes and the relationships among the classes [30].

The platform structure is presented in class diagrams, organized in four packages: Core, Process, Task and UI. This division is according to the main concepts of the framework, defined in Section 3.2.2 Meta Models.

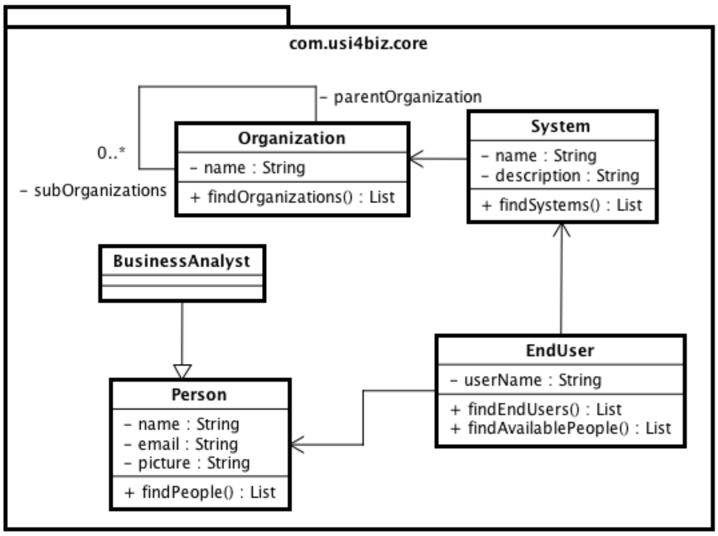

Figure 4.12: Core Package

The Core package contains two central elements:

- *Organization* - the organization entity that is composed of systems.
- *System* - the system entity that is composed of UI components.
- *EndUser* - the end users who use systems.
- *Person* - people who work with systems in the organization.
- *Business Analyst* - a person in the organization responsible for business processes.

The BP package is based on BPMN specification [87] and contains:

- *BusinessProcess* - the business processes of an organization.
- *FlowObject* - the elements that define the behavior of a business process.
- *Activity* - the work done in an organization as part of a Business Process.
- *SubProcess* - compound activities in a Business Process.
- *Task* - atomic activities in a Business Process.

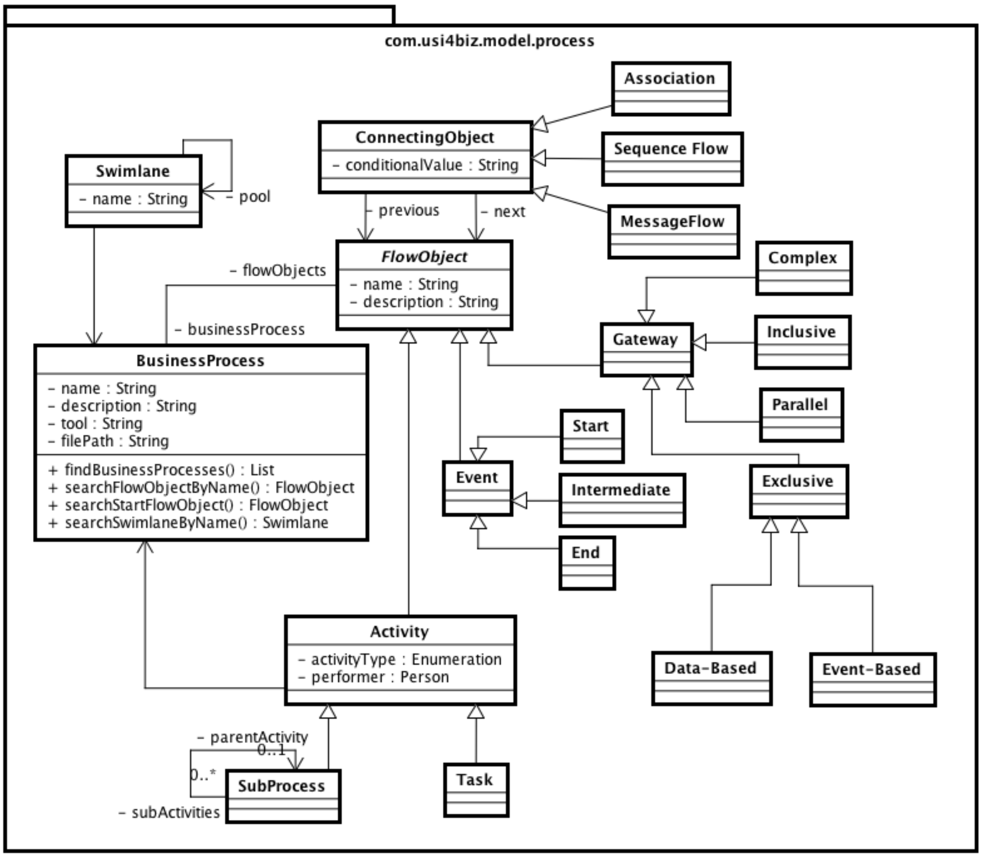

Figure 4.13: Business Process Package

- *ConnectingObject* - the ways to connect Flow Objects to each other.
- *Association* - the association of information with Flow Objects.
- *SequenceFlow* - the order in which activities will be performed in a Business Process.
- *MessageFlow* - the flow of messages between two swimlanes in a Business Process.
- *Event* - the events that happen during a Business Process.
- *Start* - the start event that indicates where a particular Business Process will start.
- *Intermediate* - the intermediate event that indicates where an Event happens somewhere between the Start and End of a Business Process.

- *End* - the end event that indicates where a particular Business Process will end.
- *Gateway* - the elements to control the flow of Connecting Objects.
- *Inclusive* - the inclusive gateway used as a branching point.
- *Parallel* - the parallel gateway used to create a parallel flow.
- *Exclusive* - the exclusive gateway used to create alternative paths.
- *Data-Based* - the data-based exclusive gateway that uses the values of process data to determine which path should be taken.
- *Event-Based* - the event-based exclusive gateway where the alternatives are based on events that occur at that point in the Business Process.
- *Complex* - complex gateways used in situations that cannot be handled by other gateways.
- *Swimlane* - the elements that partition and organize activities in process participants.

The Task package contains:

- *Task Model* - the task model composed of UI Tasks.
- *UITask* - UI Tasks that are organized hierarchically from a parent task and decomposed in sub tasks.
- *UITaskRelationship* - the relationships between UI Tasks.
- *Enabling* - the enabling relationship between UI Tasks, which represents a sequential flow.
- *EnablingInfo* - the enabling with information passing relationship between UI Tasks, which represents a sequential flow that passes information from one task to the next.
- *Disabling* - the disabling relationship between UI Tasks, which represents an interruption in the flow of tasks.
- *SuspendResume* - the suspend / resume relationship between UI Tasks, which represents a momentary interruption in the flow of tasks.

- *OrderIndependence* - the order independence relationship between UI Tasks, which represents the execution of independent tasks.
- *DeterministicChoice* - the deterministic choice relationship between UI Tasks, which represents a branching point in the flow of tasks.
- *Concurrency* - the concurrency relationship between UI Tasks, which represents a parallel flow of tasks.
- *ConcurrencyInfo* - the concurrency with information passing relationship between UI Tasks, which represents a parallel flow that passes information between tasks.
- *UITaskDecomposition* - the decomposition of UI Tasks in sub-tasks.
- *FlowObjectUITask* - the mapping between UI tasks and BP elements.

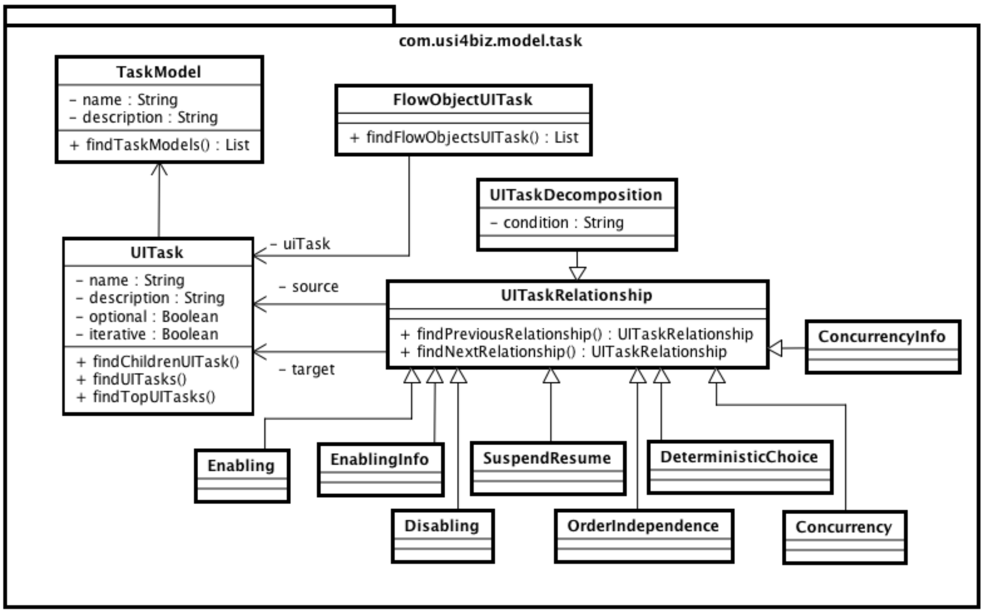

Figure 4.14: Task Model Package

The UI package contains:

- *UserInteraction* - the interaction between end users and UI components.
- *UIComponent* - the different types of components in UIs.

- *UIComponentRelationship* - the relationships between UI components.
- *ComposableUIComponent* - the different types of UI components that can be subdivided in smaller parts.
- *ScreenGroup* - a group of closely related screens.
- *Screen* - a class that represents screens.
- *ScreenFragment* - the fragments of screens that contains related elements.
- *ScreenElement* - atomic elements in screens and screen fragments.
- *UIComponentType* - the different types of UI components.
- *ScreenType* - the different types of screens.
- *ScreenFragmentType* - the different types of screen fragments.
- *ScreenElementType* - the different types of screen elements.
- *UIComponentUITask* - the mapping between UI components and UI tasks.

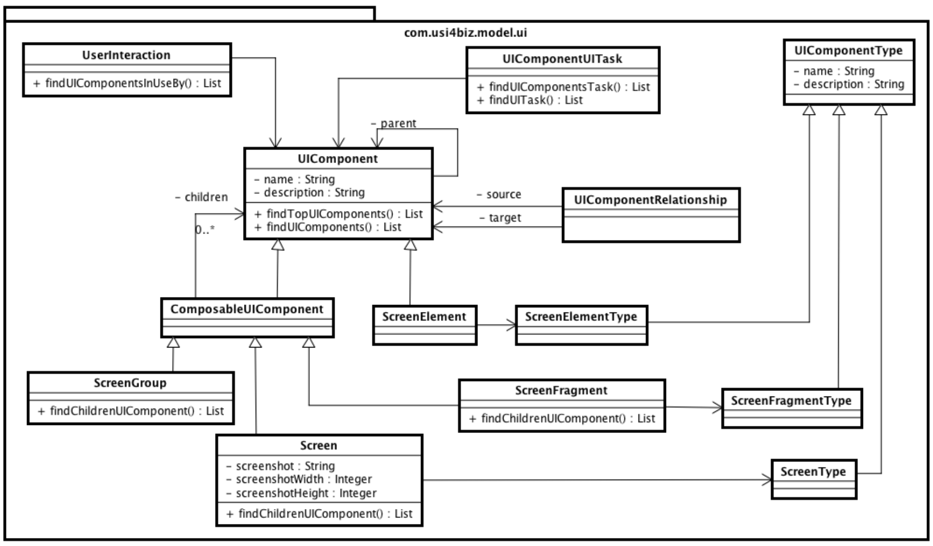

Figure 4.15: User Interface Package

4.4.2 Software Behavior

The behavior of the platform is based on the methodology; accordingly, this platform can be used by different stakeholders, depending on their goal within the organization. To support the creation of the network of links in the forward approach, each stakeholder has a specific role.

Business analyst - When business analysts either create a new business process or update an existing one, they import the business process model into the platform as a starting point to create the network in the forward approach. Once this is done, the task model is automatically created based on the imported business process.

Usability expert - Once a first version of the task model has been automatically created, most of the repetitious work is done and usability experts can focus on reviewing it and adding UI tasks that represent the user interaction.

UI designer - As the usability expert is working on the task model, the UI designer lists: the systems used to perform this business process, the users of these systems, and the UI components of each system. The level of detail given to the list of UI components depends on the level of detail expected for the result of the impact analysis. Once all these elements are listed and the usability expert finished updating the task model, the UI designer can associate the UI tasks with the listed UI components.

Operation Manager - When the network of links is created, the operation manager can request an impact analysis concerning a specific change that he/she prospects. When the operation manager, views the result of the impact analysis with the list of suggested changes, he/she can make a solid decision.

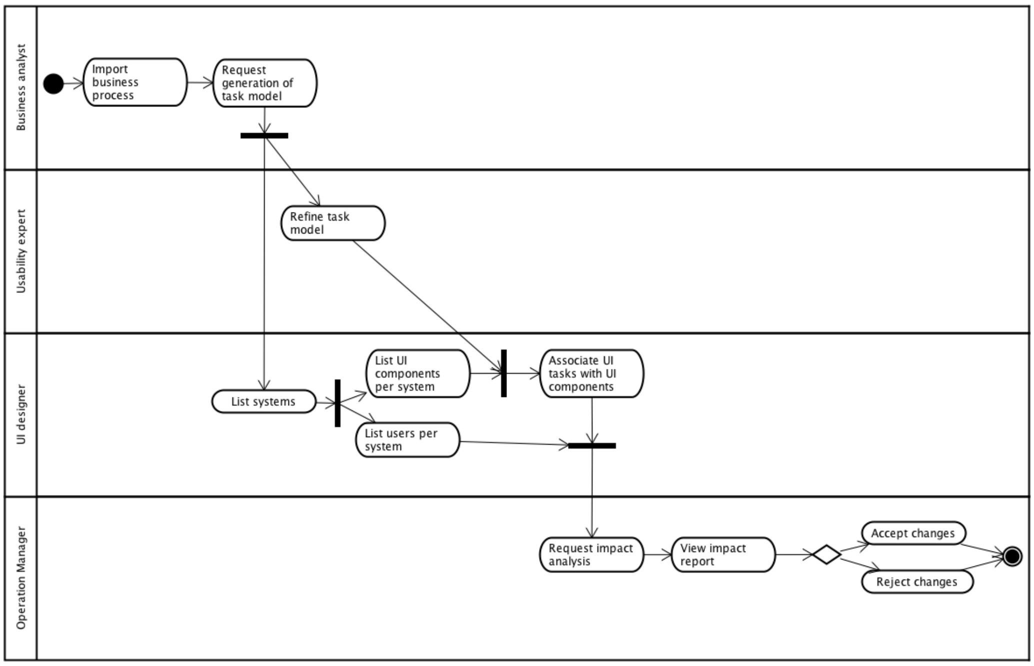

Figure 4.16: Software Behavior

The main functionalities presented in the process of the software behavior are listed as follows:

1. Export the business process XML from a BP tool and import it into the UI-Business Alignment Platform;
2. Request the generation of task models based on the BP;
3. Refine the generated task models by updating UI tasks and relationships;
4. List systems used to perform this business process;
5. List UI components based on task models;
6. List system users;
7. Associate UI tasks with UI components;
8. Request impact analysis;
9. View impact report.

This scenario is an illustration of the software behavior for the creation of the network of links and impact analysis in the forward approach. The other actions of the methodology (presented in Section 3.3.2) are also supported by the platform. And the use of the roles allocated to the functionalities depict suggestions, it does not represent any restriction in the use of the platform.

4.5 Conclusion

This chapter presented the UI-Business Alignment Platform as a support for the creation of a network of links between the models, which enables the performance of an impact analysis when changes are made in these models.

Completing the presentation of the methodology and the platform, forming the framework, it is possible to analyze the different aspects of this framework both in theoretical and in practical terms, subject to the next chapter.

Chapter 5

Validation

"People ignore design that ignores people.", Frank Chimero.

This chapter presents the validation of this research work in theoretical and empirical terms. The research results presented in this work are validated following the taxonomy used to classify empirical validation methods [141]. From that taxonomy, the selected validation methods are:

- *Literature Search* to evaluate published studies that analyze the behavior of similar researches.
- *Case Study* to collect detailed project data to determine the results of the application of the proposal.
- *Lessons Learned* to perform a qualitative analysis on completed projects to see if the proposal had an impact on the project.

5.1 Theoretical Validation

The first theoretical validation was made in sub-section 2.3.3 Comparison of the approaches in Chapter 2 State of the Art, where related works were compared using a set of relevant requirements. The second step is to demonstrate how the proposed framework addresses the requirements for UI-Business alignment.

The requirements that have been defined in section 2.3.2 Requirements of Chapter 2 State of the Art are analyzed in details as follows.

User-Centeredness: The UI-Business Alignment Framework contributes with its focus on user interaction putting task models as a cornerstone.

Task models: The use of task models as a bridge between business process and UIs values the user perspective, differently from existing solutions that have deliberately relied on IT artifacts for alignment, which is strongly focused on technical aspects.

The consequent benefit of addressing this requirement is to provide a framework that is concerned with the users who interact with enterprise systems, especially professionals in their daily work. The UI-Business alignment guarantees that these users are working towards achieving organizational goals, which have been broken down as activities and detailed in UI tasks to be performed using enterprise systems. And the user perspective put forward in this alignment provides a *richer user interaction*.

Adaptability: The UI-Business Alignment Framework has the ability to be adopted in different organizational contexts because of its generic structure and the use of rules.

Generic structure: The framework adopts a generic structure of its main models, which makes it appropriate for application in the context of large organizations that are constantly changing. This framework accepts different approaches to BP modeling because the main requirement to apply a business process notation is that it is able to decompose activities in smaller parts. The framework adopts a simple and hierarchical structure for task and UI models that is generic enough and it does not limit in how many levels a screen can be decomposed because it accepts that the organization adopts different strategies to design and develop UIs.

Rules: The framework uses specific models (BPMN, Task Model and UI Model) that address most business process modeling challenges and also most user interface interaction scenarios. In parallel, how people use these models varies a lot. The framework adopts rules exactly to give the flexibility to adapt our approach for many scenarios. Without adopting rules, we have to say to the user that we have to change the way they are doing things otherwise, they cannot use our framework and our intention is to enable easy customization and adoption.

The consequent benefit of adaptability is that the framework is adaptable to organizational structures and choices. Some companies ignore the idea of sub-processes, they just have a list of processes. For these cases, rules should be changed to adapt the transformation to the task model. The same applies to how the organization structures its UIs in different components, in such cases, rules should be changed to adapt the rules that check the association between task model and UIs.

Human-Centeredness: The framework considers the viewpoints of BP performers (users of enterprise systems) as active agents and presents the methodology in a standard and comprehensible manner.

Users: Users of enterprise systems are recognized in our approach as active agents who can point out issues that interfere with the progress of their work. The communication of users directly with business departments opens a new channel for business process improvement, it is a user-centered channel. Inputs for business improvement are no longer only from business and IT departments, they come from the people who actually make the organization work and grow.

Understandability: The methodology is also presented using the ISO/IEC 24744 graphical notation to depict the method in different diagrams: a) Lifecycle to present the overall structure of the method; and b) Process to describe the processes and its associated roles and tasks. The use of this standard notation is motivated by the fact that a clear understanding of the methodology facilitates spreading the culture of users as active agents for process improvement.

The consequent benefits of providing a human-centered perspective is that the framework is concerned with the people who perform the business processes and use enterprise systems by giving them the opportunity to be more active by giving suggestions that may improve their daily work. In addition, professionals involved in maintaining the traceability can profit from a framework that has a methodological support in how to apply it. This solution can be well adopted and maintained because it has a generic structure that enables integration with any software process and project management methodology.

Traceability: The framework provides traceability between the models by creating a network of links between them, opening a channel for people to suggest changes from any model, and executing rules that identify what is impacted from changes on any model.

Mappings: The definition of how specific models are mapped in both forward and backward strategies has been done to demonstrate the possibilities of links between elements of these models.

Multiple sources: The framework enables the impact analysis starting from any of the models to allow more people to suggest improvements on the processes, independent of their background knowledge, thus providing a forward, middle-out and backward traceability.

Rules: A set of rules has been defined in order to guarantee a consistent alignment between business process, task model and UI components. The rules can be executed to analyze prospect impacts before the changes are made to support deciding whether the change will be made; or to actually update the models to guarantee consistency after the changes have been made.

The consequent benefits of traceability is that all the models that support 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. And for those changes that have not been analyzed beforehand, the UI-Business Alignment Framework reviews the performed changes and suggests a list of actions that should be executed in order to maintain the alignment.

Software Support: The platform is implemented based on requirements elicited during the two case studies. The functionalities of the platform were detailed in Section 4.2 and the implementation takes some constraints into consideration: a) the organization has systems implemented in several technologies and the platform should not be intrusive by aiming to generate some artifact dependent on platform, such as final user interfaces. This generation would force the organization to adopt the technologies chosen by the platform or restrict its use to supported technologies. b) The platform should reuse what the organization already has as artifacts, such as business process models, importing them without any additional effort from the user and proceeding with the alignment according to the methodology. c) It should run in several technologies because there are users operating systems in those technologies as well, making Java a natural choice for the implementation.

Besides, the platform has been designed as a desktop tool due to the integral support for modeling, all data are synchronized with a centralized server, which is responsible for sharing all modeling and alignment work with several stakeholders. It characterizes the platform as a rich client platform (RCP), which can only demonstrate its full potential when connected to the network.

5.2 Empirical Validation

The framework has been empirically validated by applying a qualitative approach since it studies people, settings and situations as intimately interconnected and it enables researchers to emphasize validity in their research [129]. The selected research strategy is *case study*, which is distinctively advantageous for this research because we analyze a contemporary set of events in a real-life context, over which the researcher has little or no control [136]. For a research that is interactive and participative in nature, the level of deployment considers the following aspects: The degree of "participation" is high since the researcher performs interviews and participant observations. The degree of "revealedness" is the high since the participants are fully aware of the study that is going on. The "intensiveness" is established in one-hour interviewing sessions, one-hour observation sessions in the professional's setting and the "extensiveness" is established at first in a three or four-month period, in which sessions can happen once or twice a week in order not to disturb their ongoing works. The "focus" of the study concerns a specific research problem of maintaining alignment between business processes and UIs of its enterprise systems.

The sources of evidence most common in collecting data in case studies, as described by Yin [136] and Decrop [21], were applied in this research. Using different sources enables addressing a broader range of issues and provides triangulation by corroborating findings from different sources of information. The sources of information used in the case studies were: (i) *documentation*, analysis of business process models and UI design; (ii) *interviews*, semi-directive interview (with interview guide) and non-directive interview (informal conversational interview); (iii) *observations*, participant observation in which the researcher becomes immersed in the real settings; (iv) *discussions in group*, in which the researcher participates in meetings where stakeholders discuss topics relevant to the research.

The *unit of analysis* in this research is the alignment of business processes with UIs of its enterprise systems. The *site* selected to be studied focuses on a specific setting: large organizations that use business processes and/or have extensive enterprise systems, such as telecommunications, bank and insurance corporations, etc. It also focuses on particular *populations*: business analysts, UI designers, usability experts, operation managers and users of enterprise systems.

The framework has been empirically validated in two main case studies that took place in two large Belgian organizations. The first one in the bank sector and the second from the telecommunications industry, as presented in Section 3.3.1 Organizational Context in Chapter 3 UI-Business Alignment Framework. The analysis of the results of these case studies is presented in the following sections.

5.2.1 Bank Sector

This framework has been validated with its application in the context of a large bank and insurance organization. The characteristics of this organization have been presented in Section 3.3.1 in Chapter 3 - UI-Business Alignment Framework. The scenario related to creating the network of links in the forward approach has been presented in Section 3.3.3 in Chapter 3 - UI-Business Alignment Framework. The scenarios of changes in forward and middle-out approaches have been presented in Sections 4.3.1 and 4.3.2 in Chapter 4 - Platform Specification.

Conducting the Case Study

We have conducted several interviews with the business department (three business analysts) and IT department (two system analysts/developers and two UI designers/human factors experts) to better understand their context, their difficulties and needs. During the meetings, we worked with examples of business processes and systems for insurance contracts.

As professionals were interviewed and observed, we have gathered a deep understanding of their situation. In several meetings within this bank, we observed that 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.

Considering these issues, we proposed to them the first version of the UI-Business Alignment Methodology. The methodology did not have a great impact on their way of working because it was largely based upon their own methodology. The main difference was that instead of working with a lot of different artifacts, often not even aligned with each other and with information duplicities, their work is improved by using mapped conceptual models. New models with well delimited scope help them consider the user perspective and allow them to better communicate with other departments. Therefore, it is possible to have a smooth transition from a business focus into a hybrid perspective that also includes the user perspective.

Qualitative Analysis

This methodology has been analyzed by top managers from the business department, who have given positive feedback considering cost analysis, feasibility of tool support, maintainability and acceptance of change for the examples in which it was applied.

The organization aimed at a short-term solution that did not require much investment. Therefore, we acknowledged that there is an underlying cost in our proposal that characterizes it as a long-term proposal. First, there is the cost needed to develop a tool to manage the various models. Existing tools for business process modeling and task modeling can be used, but they lack major functionalities, such as mapping these models and tracking changes in one tool. Such a tool is crucial for allowing the correct alignment between the business process and the UI models, and also to ensure traceability and synchronization. In addition to that, there is also the cost of training the professionals on task modeling and the cost with time to actually create and refine the task models.

The effort of refining tasks models is justified in contrast to the time spent on updating long spreadsheets, highly error prone, and managing artifacts spread in different tools and servers. In addition, besides serving for traceability purposes, task models are also useful for user interaction analysis.

Even though, despite their time and budget restriction, we have called attention for crucial factors that make this proposal feasible. 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 large organization 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 is 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 makes the concern on the learning process not a main issue for acceptance in a prospect pilot project.

Lessons Learned

In order to apply this framework, 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.

Their concern with professionals having to learn the task model made us consider the need to simplify it with only interactive tasks to increase the task model simplicity and acceptance, as argued in Section 3.2.2.

This approach started to become concrete inspired by this case study and consolidated by recent researches, explored throughout this thesis. However, this application represented a first experience in a large organization that required this approach to be validated in other case studies considering different business profiles, such as the case presented as follows.

5.2.2 Telecommunications Industry

In the context of a large telecommunication company, we have gathered empirical data and performed qualitative analysis of the improvements gained following the adoption of the framework.

The characteristics of this organization have been presented in Section 3.3.1 in Chapter 3 - UI-Business Alignment Framework. The creation of the network of links in the backward approach has been presented in Section 3.3.3 in Chapter 3 - UI-Business Alignment Framework. The scenarios in the middle-out and backward approaches have been presented in Sections 4.3.2 and 4.3.3 in Chapter 4 - Platform Specification. Other scenarios are presented in the following sections.

Conducting the Case Study

The units involved in this case study are: an IT and networks support unit responsible for delivering IT solution and a customer-unit responsible for Business-to-Business transactions. The case study [66] consisted of two projects that took four months with visits twice a week, organized as follows:

The first month was devoted to understanding the organization and specific departments and professionals in order to select the business processes and involve committed professionals in the application of the methodology.

This initial analysis was supported by a survey prepared before the case study in order to guide the semi-structured interviews and direct the questions. In addition, the organizational documentation available in the Intranet was also considered in the organizational analysis.

The second and third months involved: (i) for the first project: meetings with business analysts to understand the business process and system and modeling the business process; (ii) for the second project: attending weekly meetings with the process core team (process manager, IT specialist, project managers and tester) who defined and modeled the business process, using the system under analysis in the test environment, and observing user acceptance tests of the system for this process; (iii) for both projects: weekly follow-up meetings to debrief the progress of the work. Twenty-three (23) professionals have been interviewed, working in twenty-one (21) different departments out of four (4) different business units.

The fourth month included the application of the methodology on the selected business processes and systems in both forward and backward approaches; the integration of this methodology with the organization's software development process and process improvement methodology; validation and dissemination of the results of the methodology, finalizing with the decision to include a suggestion that resulted from applying the methodology in their next release and the prospect continuation of this project to gather further data to support the cost-benefit analysis of the methodology.

Applying the Framework

The case study concerned a supporting business process and a core business process: 1) the Accounts Payable (AP) business process that controls invoices; and 2) the provisioning business process of Integrated Telephony Solution (ITS) products, which are specific products of the enterprise Voice over IP (eVoIP) solution of a telecommunications company. Each of the business processes is described and the alignment methodology is applied to perform the impact analysis in different change scenarios, presented as follows.

Accounts Payable

This project took place in the IT unit of the Finance department. The accounts payable business process starts with the creation of purchase orders to request products or services. Once the product is delivered, a goods receipt is created, when required, with a list of the goods ordered that have been effectively delivered. When the supplier sends the invoices for the ordered and delivered products, the accountants scan the invoices and introduce them in the AP system. The scanned invoice and its related data are sent to the approver to ensure that the price and quantity informed on the invoice are correct. Then, the approver analyzes the data and either accepts, refuses the invoice or requests a credit note to correct the data in the invoice that is different from the purchase order, which may be with price or quantity difference ("Approval of invoice"). If the approver requests a credit note, the credit note is sent to the requester who can accept or refuse the credit note and the related invoice ("Approval of credit note"). To guarantee that the process flows smoothly, there are sub-processes to remind the stakeholders to create the credit note ("Remind to do credit note") and to create the goods receipt ("Remind to do goods receipt").

We have identified scenarios of changes proposed by business analysts, operation managers or end-users in both forward and backward approaches. The scenario demonstrated in this section updates existing activities in a forward approach. A scenario related to creating new UI components in the backward approach has been presented in Section 3.3.3 in Chapter 3 - UI-Business Alignment Framework.

Scenario - Monitoring invoices: *Operation managers proposed to monitor the approval of invoices by asking approvers to justify their choices.*

This way, when approvers make a decision about the invoice (accept, refuse or request a credit note), they have to provide a justification.

This scenario requires the inclusion of a new activity in the "Approval of invoice", "Approval of credit note" and "Remind to do goods receipt" sub-processes (Figure 5.1). As a result, the activity to refuse invoice becomes a sub-process where the approver will also have to justify the refusal.

In the related task model, the refuse invoice task will now consist of informing and justifying the refusal. The justification can be predefined (select justification) or provided by the user (inform new justification). From the mapping, it is possible to detect new tasks without UI components (presented with dashed border), which shows the need to create a new screen and new screen elements to support the new UI tasks.

Results. The results of the impact analysis of the different scenarios have been validated by two SAP solution managers, one IT senior developer and a group of IT developers in the finance department in this project.

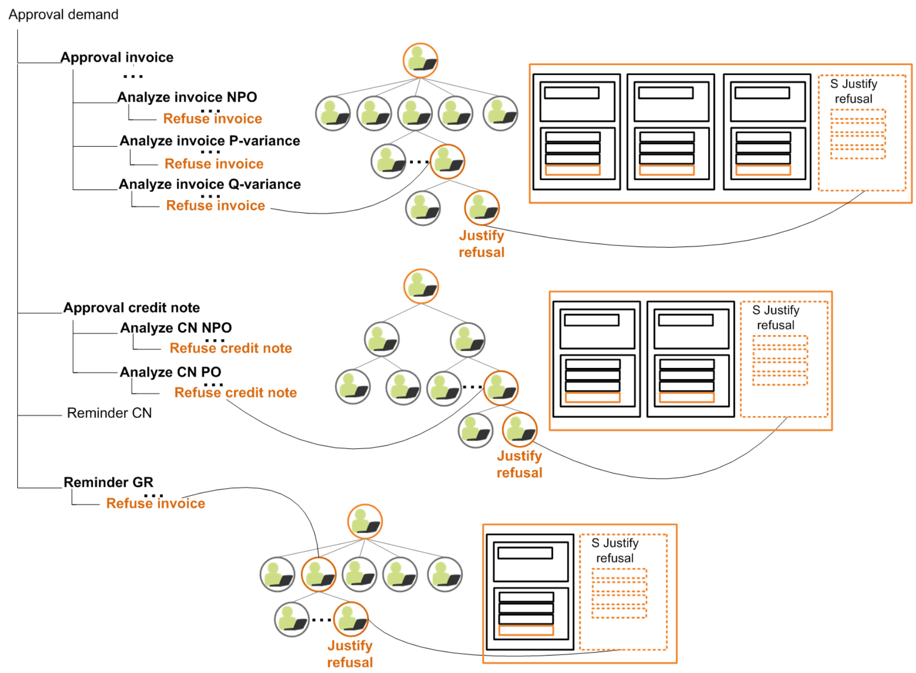

Figure 5.1: Scenario Monitoring Invoices

Provisioning

This project takes place in the team in charge of the enterprise business department. The selected business process is the provisioning of the ITS products of the eVoIP solution for professional customers who want to integrate their data and voice traffics on the same data network. There are two categories of products that are determined by the quantity of dialing numbers that the customer requires and by the number of sites (location of customer's installation addresses) that can be provisioned. The ITS products provide different ranges of dialing numbers, starting at one dialing number, going to ranges of 10 and/or ranges of 100 dialing numbers. Each number may be associated to a package that represents voice services, such as call center, auto-attendant, receptionist, etc. This provisioning business process starts with the accountant filling out the form with the information of the customer's request and creating the order file. Then, the Project Coordinator starts controlling the flow that drives the provisioning process. The team in charge of the project setup checks if the customer is new, if negative, this task can be directly closed; otherwise, the team has to make a series of tests with the order file. When the file is ready, the customer care department analyzes the order file to understand what the customer has requested.

If the customer has only one-site, the customer care department orders the requested products ("Introduce products"). The association between the product and the connectivity has to be made. If the customer has multi-sites, the customer care department creates a group with the needed licenses ("Introduce client group of licenses"). If the customer requests has multi-sites and has ranges less than 10, the customer care department orders the requested products and associates them with the previously created group ("Introduce products"). The association between the product and the connectivity has to be made.

The next step is the pre-activation of the product, where the lines and numbers are ready but not yet activated; only voice tests are possible. Afterwards, the technician plans an installation with the customer. Once installed, the technicians validate the installed products. The numbers are finally activated and the site request can be closed by the Project Coordinator.

We have identified scenarios of changes proposed by operation managers, business analysts or end-users in forward, middle-out and backward approaches in which the alignment methodology has been applied. The scenario demonstrated in this section creates an entirely new sub-process in a forward approach. The other scenarios that depict splitting, merging and creating new UI components in the middle-out and backward approaches are presented in Sections 4.3.2 and 4.3.3 in Chapter 4 - Platform Specification.

Scenario - Products with mixed ranges: *Operation managers proposed the creation of a virtual PBX to make the association among all the telephone installed in the multiple sites.* Products with mixed ranges; that is, products with ranges of 10 and also those greater than 10; were treated as distinct and not associated orders, which could even cause losing data.

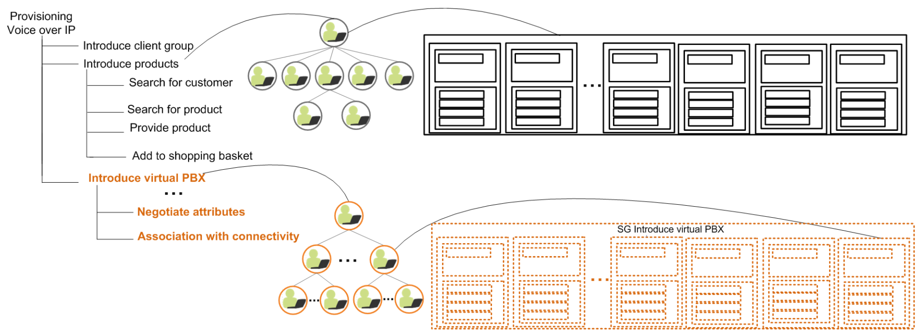

Figure 5.2: Scenario Products with mixed ranges

This scenario requires the creation of a new sub-process "Introduce virtual PBX" (Figure 5.2). When introducing the virtual PBX, the user has to select the attributes of the product and associate the product with the connectivity. This set of new UI tasks requires UI components to enable the user to perform these tasks.

Results. The results of the impact analysis have been validated by a Process Owner, a Project Architect, and by representatives of both the Business Transformation and Methodology departments in this project.

The impact analyses and changes performed through the methodology represent a step towards addressing their strategic goals. Such goals envision improved customer experience, hence, future improvements on their provisioning processes throughout the organization.

The application of the alignment methodology in a large telecommunications company enabled its qualitative and quantitative analyses. The qualitative analysis was made in terms of feasibility of integration with corporate initiatives, such as software development process and process improvement methodology. The quantitative analysis was done through the Return on Investment (ROI) forecast in terms of the gains achieved relative to the investment on the alignment methodology.

Qualitative Analysis

We have analyzed the organizational software development process and identified how it can be integrated with the alignment methodology.

Their iterative software development process is organized in nine phases, which are similar to the Rational Unified Process (RUP) disciplines that range from requirements, business modeling, analysis and design, implementation, deployment, test, project management, change management, environment [60]. Matching their process with RUP to facilitate explaining the integration with the methodology, we present it in phases.

In the *inception phase*, first, a project plan lists requests that are documented with a high level estimate of the domains impacted; then, a project description is prepared with the core project requirements, constraints and key features. To aid in adequately creating these deliverables, the alignment methodology provides for the project plan: a list of organizational units and systems impacted from the requests; and for the project description: a detailed list of organizational units, systems, processes and users impacted from the requests.

In the *elaboration phase*, a development plan for the overall project is described with solution architecture definition and domain analysis. For the development plan, the alignment methodology provides a list of user interfaces and users impacted from the requirements in order to better identify which kind of training users will need.

In the *construction phase*, the main goal is to build the software and the primary deliverables are resulting from coding, such as components, classes, services, etc. The methodology supports this phase when change requests are created and the impact analysis aids in creating a list of impacted organizational units, systems, processes, user interfaces and users.

In the *transition phase*, the goal is to launch the system in production where the system is validated with users. The methodology supports this phase when change requests are created from users' requests and the impact analysis aids in creating a list of impacted organizational units, systems, processes, user interfaces and users.

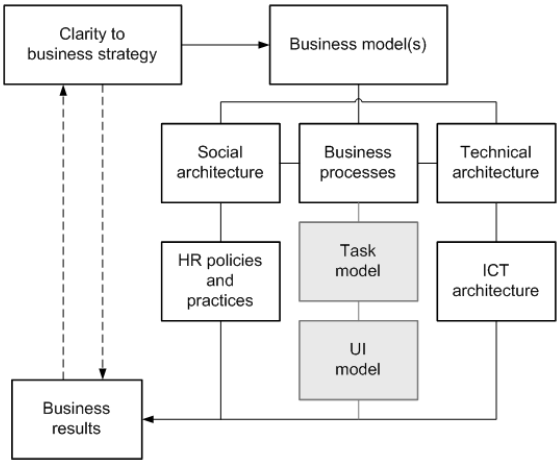

Figure 5.3: Capability Building Framework integrated with the UI-Business Alignment Framework, adapted from [96]

The alignment methodology is also easily integrated with enterprise architecture approaches, such as the Framework for Capability Building [96], which covers managing Business strategy, Business model, Business processes, Technical and Social structures, Human Resource policies and practices, ICT architecture and tools, and Business results. The alignment methodology complements this strategy framework by adding the Task Model and the UI Model (Figure 5.3), thus bridging the gap between enterprise architecture and HCI, which aids in acquiring a sustainable competitive advantage with end-users and customers.

The company's process improvement methodology; similarly to the typical process improvement methodology, e.g. PDCA [72]; is an iterative problem-solving process to optimize its business processes and achieve more efficiency. The result from applying a process improvement methodology is a list of processes that need to be changed in order to achieve the organization's objectives. To implement the proposed changes, a typical solution [135] consists of new/updated functions, services, applications that have to be managed in order to solve the identified problems. The alignment methodology extends the solution given by typical business process improvement methodologies by listing the impacted UIs and going further by listing impacted organizational units and users of impacted enterprise systems.

Quantitative Analysis

In our analysis, we aimed to acknowledge the longer term and temporal aspect of decisions, which enabled us to consider risk and uncertainty and how the organization regards them. This analysis is tightly related to Operations Strategy (OS), which is concerned with developing *resources* and configuring *processes* so that the resulting *competencies* best deliver the proposed corporate value [133].

The purpose of resources is to work and generate value, one type of resource is a tangible real asset, which is a human resource; in our case, they are business performers and simultaneously system end-users who apply their intangible real assets (their "know-how") to do their work. The purpose of processes is to specify how resources perform their activities and add value; in our case, the business processes of AP and eVoIP. The purpose of competencies is to specify what organizations can and cannot do and how well they provide their outputs and differentiate in the competitive market; in our case; the ability to do impact analysis from a business and user perspective.

The results of these two projects have been presented to the teams in charge of these projects. The evaluators have appreciated the innovative approach adopted for their projects. We went further the qualitative analysis and have worked on quantifying the results through a cost-benefit analysis of applying the alignment methodology.

We have analyzed the approach under three cases that consider optimistic, pessimistic and probable visions of the future. Sensitivity analysis is a technique for systematically changing parameters in a model to determine the effects of such change [57]. We selected the sensitivity analysis to obtain information from the system with a minimum of numerical experiments since we have observed eight users interacting with the ordering system to provide eVoIP products. The steps we took to evaluate the cost-benefit analysis were:

1. Calculate the time and cost to apply the methodology;
2. Observe users interacting with the system;
3. Identify issues as candidates for a change scenario;
4. Count interaction time for a specific scenario;
5. Calculate the time and cost to develop and deploy the change;
6. Calculate the cost-benefit analysis.

The calculations of the time and cost to apply the methodology were based on our experience in this case study considering the methodology actions and the complexity of each scenario. The calculations of the time and cost to develop and deploy the change were made based on information provided by the IT department. The calculations of the gained time were made based on users' observations by quantifying the time spent on each step in the eVoIP process before any change; then observing them and quantifying the time spent on UIs after one change.

Two scenarios have been selected for a cost-benefit analysis as a request from the team who was closely following the application of the framework, including a Process Owner, a Project Architect, and representatives of both the Business Transformation and Methodology departments. These scenarios are from the backward approach (presented in Section 4.3.3 in Chapter 4 - Platform Specification).

Scenario 1 - Configure numbers: In this scenario, users have to do twice the same task on different UIs when choosing the service packages for each dialing number of eVoIP products.

This issue decreases their efficiency when managing the orders. In general, it takes an average of two hours per site to introduce a product; thus, a gain in efficiency in such a complex scenario makes a big difference for the users managing customers with several sites. To solve it, users suggested to be able to select voice service packages in only one screen.

The time to apply the methodology for this scenario took a total time of nearly 11 hours. In the *probable case*, the gained time after the alignment is 2:30 minutes per user, nearly 4 hours considering 88 users of the ordering system for eVoIP products. After a year executing this process 2 times per month, the benefit amounts to nearly 63% and the ROI is achieved after the third month. In the *pessimistic case*, the gained time after the alignment is 1 minute per user, nearly 2 hours considering 88 users of the ordering system for eVoIP products. After a year executing this process 2 times per month, the benefit amounts to nearly 15%. In the *optimistic case*, the gained time after the alignment is 2:30 minutes per user, nearly 4 hours considering 88 users of the ordering system for eVoIP products. After a year executing this process 8 times per month, the benefit amounts to nearly 315%. Figure 5.4 depicts the ROI in details for the probable case and the comparison of the three cases for this scenario.

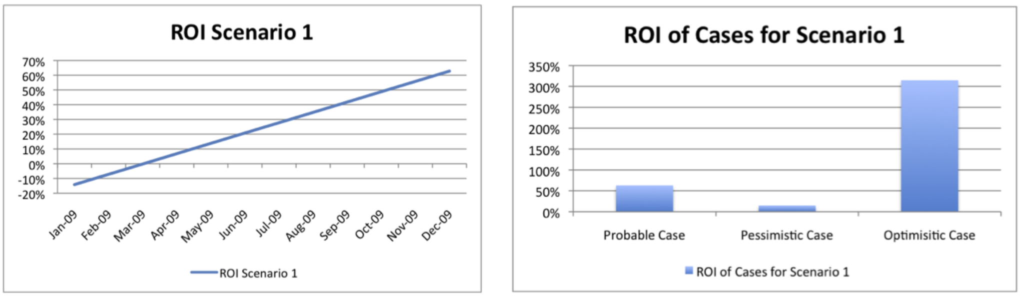

Figure 5.4: Return on investment applying the UI-Business Alignment Methodology for Scenario 1

The empirical data was collected from observation of 8 users interacting with the ordering system to provision enterprise Voice over IP (eVoIP) products. The few detected deviations on the timing between these users are due to loss of data: user lost contract number; user lost customer address; or due to user interaction: user could not find products in the list.

Scenario 2 - Licenses overview: In this scenario, users negotiate features of products, however they have to remember the types and quantities of licenses they introduced in previous screens of the system.

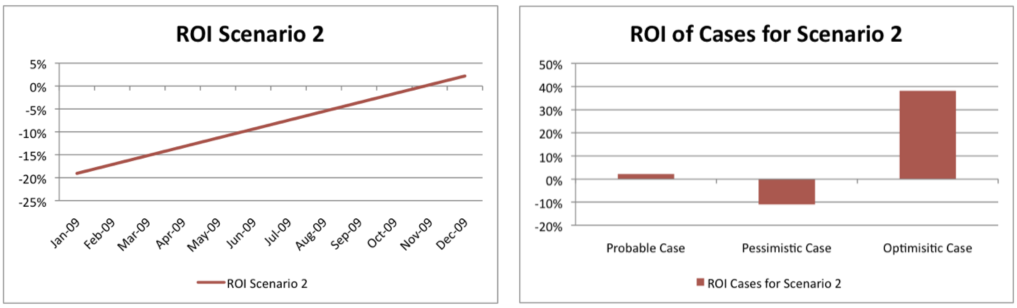

Figure 5.5: Return on investment applying the UI-Business Alignment Methodology for Scenario 2

As a result, users forget the information, write the information on a piece of paper or have to navigate through five UIs to find the licenses and select each type of license to see the previously informed quantities. To solve it, users suggested to be able to see the overview of available licenses for the customer as they request the customer product. The proposed change consists of adding information in the UI where users negotiate features in order to support the user with relevant information about the licenses.

The time to apply the methodology for this scenario took a total time of nearly 10 hours. In the *probable case*, the gained time after the alignment is 3:40 minutes per user, around 30 minutes considering 8 users of the ordering system for Integrated Telephony Solution (ITS) products. After a year executing this process 4 times per month, the benefit amounts to nearly 3% and the ROI is achieved after the tenth month. In the *pessimistic case*, the gained time after the alignment is 3:10 minutes per user, nearly 30 minutes considering 8 users of the ordering system for ITS products. After a year executing this process 2 times per month, the result is -11%. In the *optimistic case*, the gained time after the alignment is 4:42 minutes per user, nearly 40 minutes considering 8 users of the ordering system for ITS products. After a year executing this process 8 times per month, the benefit amounts to nearly 40%. Figure 5.5 depicts the ROI in details for the probable case and the comparison of the three cases for this scenario.

Comparison of Scenarios: The first scenario clearly brings faster return on investment than the second scenario, as depicted in Figure 5.6 through the comparison of these scenarios showing the ROI in details for the probable case and comparison of the three cases for both scenarios.

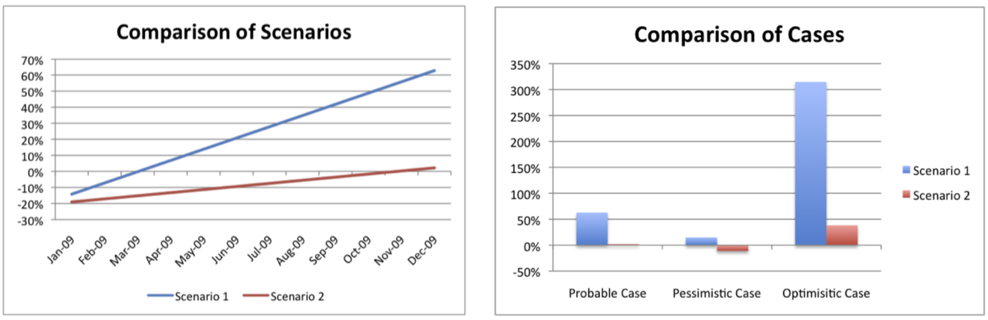

Figure 5.6: Comparison of the Return on investment of both scenarios

This quantitative representation of the return from the second scenario is only minimal because it does not consider the loss from input of wrong data for customers' requested products (as users see the data on the system, note it on paper, then pass it to the system again). Examples of direct costs are the re-work of technicians; negative impact on the brand, among others. Examples of indirect costs are the work on corrections instead of actual work. These types of costs influence the decision depending on the organization's aversion to risk to assume such costs. After this quantitative analysis, this scenario was selected as a change request to be included in their next release.

Lessons Learned

There were lessons learned in this case study: One lesson learned is related to the time consumed to apply the methodology. As a software platform has been envisioned [113], but since it is still being developed, the work is currently being done manually. The experience in this case study has been a major source of requirements to specify the functionalities of the platform.

Adding requirements as another artifact that can be linked to enterprise systems can increase the acceptance of this framework. This is especially true for organizations that originate their projects through software requirements, as observed in the analysis of their software development process.

Furthermore, people are resistant to change and we acknowledge that the alignment methodology is itself a change within the corporate culture and a support for decision making, which facilitates changes. Change management is a structured process that manages changes and it aims at communicating changes to the people involved in order to effectively implement these changes. The key to implement the new methodology is to start where the least resistance is perceived; usually through innovators who pull changes and opinion leaders who like to try new ideas, according to the Diffusion of Innovations Theory of Rogers [100].

5.3 Cross-Case Synthesis

We have compared the results from these different cases by applying cross-case synthesis, an analytic technique that is mostly applied in the analysis of at least two cases by aggregating findings across the cases [136].

The alignment methodology has already been previously applied in the bank sector and the two projects in the telecommunications sector contribute to the robustness of the methodology. Two main differences come out when comparing the case studies: the nature of processes and the engineering approach adopted.

5.3.1 Nature of Processes

A distinction is made between process types. Core business processes have a direct impact on the organization's customers; while support business processes enable the core processes to work effectively. The AP process is a support process while the eVoIP and the insurance processes are core processes.

It was easier to convince the teams in charge of core processes of the relevance of the methodology than the team in charge of support processes. This was noticed because the ones involved with eVoIP and insurance processes have an in-depth understanding of the business strategy, which facilitates understanding the value of process alignment and improvement.

5.3.2 Engineering Approach

For the AP and insurance cases, the methodology has been applied in a forward engineering way. The departments had their existing BP models that could be used as a starting point to create the task model and the mappings between the models.

In the eVoIP project, the backward engineering approach has been adopted to apply the methodology. The BP had not been previously modeled and consequently, the existing UIs of the supporting system constituted the starting point. From then on, the task model was designed and the BP modeled.

This reality allowed the methodology to be assessed in different contexts. These case studies made the methodology more robust because it has now been tested for different engineering approaches to build the network of links between the models.

5.4 Conclusion

Some advantages of the UI-Business Alignment framework highlighted from the experience in these case studies are its explicitness, simplicity, flexibility and innovation. It was easy for stakeholders to understand the methodology since it can be easily communicated to people. The output of the data analysis will be used as reflection on other aspects to continue evolving this work with some ideas of future work that we have already elicited, subject to the next chapter.

Chapter 6

Conclusion

"If you can dream, you can do it.", Paulo Coelho, international best-selling and award-winning author.

This chapter concludes this research work by presenting our main contribution and some ideas for future work.

6.1 Summary of Contributions

This work presented the *UI-Business Alignment Framework*, an innovation to solve issues of well-known IT-Business alignment strategies by integrating concepts and techniques from Model-Driven UI Development.

From IT-Business alignment, the framework inherits the essential links between the business and IT domains. From Model-Driven UI Development, the framework inherits task modeling as a major cornerstone to bring the user perspective into the spotlight in a scenario where it is commonly forgotten, since a much more prominent attention is given to business and IT. This integration leads to the identification of three main models used in the framework, which are: business process, task and UI models.

More globally, this research results in a formal integration of the Business domain with HCI, which leads to valuable collaborations. As we have experienced in collaborations with business-driven IT management, enterprise engineering and business process modeling.

The Framework is composed of a methodology and a platform that support the creation of a network of links between the models, which enables the performance of impact analyses when changes are to be made or made in any of these models. The network of links is created through semi-automatic transformation using predefined rules or manual mappings directly using the platform, thus, letting the stakeholders free to decide whether the context is more appropriate for transformations or not. According to the automation of the framework, as depicted in Figure 4.3, stakeholders may use rules to create mappings automatically from transformations of BP into TM and vice-versa or they may create the mappings themselves. When creating the mappings manually, stakeholders need to understand the business and the systems to be able to create these mappings.

This framework is aimed for enterprise systems in organizations with extensive business processes and hundreds of users using thousand of user interfaces. The main goal of this framework is to keep enterprise systems aligned with business processes from the point of view of process performers and process designers, offering a complementary approach to existing solutions of IT-Business alignment. These solutions are more concerned with requirements and implementations, while we focus on UIs, the part effectively in use by process performers (end-users). Consequently, the alignment of business with UI brings several benefits, as the ones listed as follows:

- Operation managers become aware of who is responsible for what, not only the user roles.
- Users can report bugs, suggest improvements and report occurrences during the process execution and this valuable information will be instantly linked with the corresponding process elements, helping business analysts to make decisions that may effectively improve process performance.
- System maintenance requests by business analysts are sent in a well contextualized manner to system analysts, with information about: parts of the processes that are impacted by those changes, users who will interact with the changes when the new version is released, and UI design decisions that were not well accepted by end-users.

The experience in a large bank-insurance company enabled us to propose a solution to align business processes and UIs of their enterprise systems, a major issue in this company as well as in many others in the competitive business world. This reality encouraged us to validate this approach in another organization with a different business profile aiming at analyzing their interest on the strategy and their openness to change.

The main contributions of this research work have been done considering in the following aspects:

Innovation: Definition of a new area that integrates IT-Business Alignment and Model-Driven Development

Conceptual: Definition of a framework for UI-Business Alignment

Methodological: Definition of a methodology for UI-Business Alignment

Platform: Specification of a platform to support the framework

Application: Application of the proposed strategy in case studies

Innovation: Concerning the innovation, this research has resulted in the definition of a framework for UI-Business Alignment (Chapter 3) that is model-driven, user-centered, adaptable to specific organizational structures, human-centered and provides model traceability with a software support [116].

To help support the theoretical definition of the framework, we have made an investigation of approaches on IT-Business Alignment and on Model-Driven UI Development that are concerned with business process modeling and UIs of enterprise systems in Chapter 2. From the detailed analysis of these works, we have identified a list of shortcomings that must be addressed in this domain, which were considered an inspiration to define the main requirements to be fulfilled in our research. These requirements were also used as evaluation criteria to analyze and compare these existing approaches.

Based on these requirements, we present the main characteristics fulfilled by the framework, which makes it:

- *model-driven* because it adopts a model-driven engineering paradigm with the use of models that are primordial for enterprise systems.
- *user-centered* because it follows a user-centered design approach that provides a vision for both user interfaces and user interaction using task models.
- *adaptable* because it adopts a generic structure of its main models, which makes it appropriate for application in the context of large organizations that are constantly changing.

- *human-centered* because it is concerned with the people who perform the business processes and use enterprise systems; more than merely on technological or methodological aspects, supported by the theory of structuration [31].
- *traceable* because meta-models are explicitly defined and rules for model transformation are structured to facilitate mapping the models, creating a network of links, analyzing the impact of changes, thus providing a stronger traceability.
- supported by a *software platform* that has been specified, analyzed and designed according to the needs of the framework with the flexibility of being extended by future researches.

We have innovated when using task models as a source for traceability, as presented in [111]. Generally, task models had been used to support requirement analysis, design of ISs and usability evaluation [77], and even to model user roles with different responsibilities and rights in large applications [93]. And the existing solutions for traceability, specify the links between business processes directly with systems or UIs. But we have stated that business processes are often an abstract representation of the business tasks, different from task models, which are concrete set of tasks of what could happen during user interaction. And we argue that it is important to recognize that we need to unite both worlds to map business processes with UIs of enterprise systems in order to add a user-centered perspective.

Conceptual: The conceptual contribution of this research is presented in meta models for business process, task and UI models in Chapter 3, Section 3.2.2.

The *business process model* (Section 3.2.2) is created according to BPMN because of its worldwide recognition, easily understandable diagrams, coverage in terms of business process modeling concepts, available tools supporting the notation, among other benefits as flexibility, and expandability. The meta model for business process defined for the framework is resumed to elements that are primordial for UI-Business Alignment with the possibilities of expansion.

The use of BPMN as the selected notation for the framework is founded on its benefits; however, it does not exclude other business process notations to be used with the framework, addressing the adaptability of the framework. The main requirement for a business process notation to be used is that its activities are decomposable in smaller parts, such as BPMN sub-processes that can be decomposed in as many levels as needed to specify a complex activity.

The definition of the task and UI meta-models are founded on UsiXML because of its recognition from the W3C standardization process, compliance to the model-driven engineering paradigm, adherence to UIDL requirements, among other benefits, such as flexibility and constant evolution.

The *task model* (Section 3.2.2) is simplified by using only interactive tasks because they are the ones that are useful to detect if the user has problems or not with the UI. In addition, we have contributed with a new graphical representation for the current task relationships that follow the LOTOS temporal operators [45].

The *UI model* (Section 3.2.2) uses abstract concepts (i.e. screen group, screen, screen fragment and screen element) that are simple and general enough to describe any kind of UI of enterprise systems. Our contribution with the definition of these UI concepts are solidified with their representation using informal predicate logic [37], also called first-order predicate calculus or simply first-order logic [103].

The *mapping* of these models, presented in Section 3.2.3, is done in a generic fashion, considering that organizations have divergent modeling techniques and notations and use different approaches and technologies to design and develop UIs, benefiting from the hierarchical structure of the models.

The mappings adhere to a classification of mappings [74] that support creating a traceability chain that is useful, among other things, for performing impact analysis and providing specifications for transformations between these models.

Methodological: The UI-Business Alignment Methodology (Section 3.3) is composed of core actions that guide stakeholders to map the models (business process, task model, UI model) and build a network of links that supports traceability and impact analysis when changes are requested in any of these models.

The core actions to create the network of links can be applied in two main approaches: forward and backward. The core actions to perform impact analysis can be applied in three main approaches: forward, middle-out and backward. These actions are organized in phases related to business process improvement methodologies and User-Centered Design methodologies.

The methodology is presented using a new formal graphical notation for modeling software development methodologies, the ISO/IEC 24744 [48] International Standard, a preliminary version of which is found in [33]. The use of this standard is of special interest because it is *complete*, *generic* and *extensible* as desired so as to cover most areas, enable application in different contexts, and enable adding new concepts as needed [32]. In addition, we have participated as evaluators of the notation along with the ISO authors. Using a cognitive dimensions framework, we have made a standard evaluation of the elements and diagrams of the notation proposed to support the ISO/IEC 24744 methodology meta model standard, considering the trade-offs between different dimensions. The use of this notation provides a clear understanding of the methodology, which can contribute to spread the culture of users as active agents for process improvement within the organization.

The methodology has relevant characteristics:

- *Flexibility:* The methodology is flexible enough to enable stakeholders to specify at which level of granularity they will map the elements, depending on the complexity of the processes and systems, on the information available and other aspects that directly influence the impact analysis.

- *Integrability:* This methodology can be combined with other corporate methods and processes, such as software development processes, HCI methods, process improvement methodologies, IT-Business alignment strategies, etc. It has been tested with the software development process adopted at the telecommunication organization and analyzed according to renowned processes, such as the RUP [60], the Agile Model Driven Development [4], CMMi for Development [18].

- *Cross-Organizational Engagement:* Each approach stimulates the participation of stakeholders from different departments. The forward approach stimulates the participation of managers and business analysts, who can identify prospective improvements on business processes. The backward approach stimulates the participation of basically anyone in the organization, such as UI designers and end-users, who can identify improvements directly on the UIs. The middle-out approach stimulates the participation of usability experts and UI designers, who can identify improvements in terms of user interaction. The participation of professionals from different domains with a proactive approach towards process optimization makes this framework less obtrusive and enables *"social creativity"* [28].

- *Applicability in Large Organizations:* The methodology enables large organizations to adapt efficiently to business challenges in the midst of its extensive business processes, thousands of UIs, hundreds of users and heterogeneous systems, forming an intricate structure. Large systems can benefit from it because of the activity volume and its respective screens, abstracted by models and their interrelationships.

Platform: The UI-Business Alignment Platform (Chapter 4) is a support for decision-making for the main stakeholders involved in applying the UI-Business Alignment methodology who want to track changes of BPs and UIs of enterprise systems.

The platform is used according to the methodology, which drives the specification of functionalities for the platform, presented in Section 4.2 and summarized as follows:

- *Import Business Process:* It is first needed to import the business process into the platform using a standard format, such as the XML format provided by business process modeling tools.
- *Generate Task Model:* The transformation of business process models into task models is done through the mappings between the models, which are the foundation for a set of rules. The association of business process models with task models is automatically created as the transformation is executed.
- *List organizations:* Specify the list of organizations that defines the structure in which enterprise systems are allocated.
- *List systems:* Specify the list of enterprise systems and system modules to facilitate identifying where the UI components are located.
- *List UI components:* Specify the list of UI components organized by systems, based on the screens of the enterprise systems.
- *List users:* Specify the list of users to facilitate identifying who is impacted with specific changes.
- *Associate tasks with UI components:* Creation of manual mappings between task models and UI components to create the network of links.
- *Impact analysis:* Selection of an element from a model and view of the list of related elements on the other models that would be impacted if a change is to be made on the selected element.

- *Generate impact report:* When a change is done in any of the models, a set of rules are executed in order to generate an impact report that lists all the elements on these models that have been impacted by the changes and the proposed changes.
- *Manage rules:* The rules can be used "as-is" or it can be updated according to the specific structure of the adopted models.

The rules (Section 4.3) have been defined to provide flexibility to the platform since they can be updated whenever there is a change on how the models are managed, without the need for direct maintenance of the platform.

As a contribution for this research, we have defined rules that adhere to the basic properties of transformation rules [55] and follow the general format of production rules [14]. The rules have been written using the Drools rule language because it is easy to understand, flexible, reusable, has a reasonable performance, independent lifecycle, can be easily embedded into existing applications, among other advantages of its declarative style solution [8].

Besides using the Drools rule language (DRL), these rules have also been written in a human-readable manner using Domain Specific Language (DSL)[56] in order to make them understandable by non-technical stakeholders involved in managing the rules.

The rules are exemplified using scenarios from the two case studies. The scenarios from the bank sector are used for the forward and middle-out approaches and the scenarios from the telecommunications industry are used for the middle-out and backward approaches.

Application: The application of the framework has been done in two large organizations, which illustrates the applicability of this framework in complex scenarios presented throughout Chapters 3, 4 and 5. One organization is from the bank sector, a large bank and insurance company; the other organization is from the telecommunications industry, the largest telecommunications company in the country.

The bank case study was the first application of the framework. The participants were from the business department (three business analysts) and IT department (two system analysts/developers and two UI designers/human factors experts), who participated in several meetings and interviews during three months. The business process under analysis was related to insurance contracts, a major process for the organization. The methodology was qualitatively analyzed according to a set of criteria specified by the top-management of the organization, citing them: cost, time, flexibility, traceability, and maintainability. Exemplification of how the methodology was applied in this case study is presented in Chapter 3, Section 3.3.3; examples of how the rules have been using in the insurance context is presented in Chapter 4, Sections 4.3.1 and 4.3.2; and it is summarized in Chapter 5, Section 5.2.1.

The telecommunications case study was the second application of the framework. The participants were from the finance department and enterprise business department with twenty-three professionals, among managers, business analysts and IT professionals, who participated in meetings and interviews during four months. The business processes under analysis were related to accounting and provisioning products, where the first is a support process and the latter a major process for the organization. The methodology was qualitatively analyzed by identifying how it could be integrated with their selected software development process. The methodology was quantitatively analyzed using sensitivity analysis [57] to evaluate the cost-benefit analysis of applying the methodology. Exemplification of how the methodology was applied in this case study is presented in Chapter 3, Section 3.3.3; examples of how the rules have been used in the telecommunication context is presented in Chapter 4, Sections 4.3.2 and 4.3.3; and it is summarized in Chapter 5, Section 5.2.2.

Even though both cases experienced the same issues concerning UI - Business alignment, they have different characteristics that are useful for different approaches as the network of links is being built. The bank case study was more appropriate to illustrate the forward approach because it already had its business processes well defined and their systems designed and developed. The telecommunications case study was more suitable to illustrate the backward approach since it did not have business processes modeled in the finance department, but all operations were supported by enterprise systems.

This thesis demonstrated that changes on business processes do influence user interfaces and vice-versa. To achieve this goal, a methodology has been applied in large corporations and validated using return on investment techniques. The scope of this thesis has been well-delimited to a set of requirements that had not yet been addressed in the literature neither in the industry. On the other hand, this framework does not consider how well business process models are understood by different stakeholders, how data is handled in the execution of business processes, etc. In the current state of the framework, it does not control the business process in execution time, especially since it is risky to execute the rules and change the models at runtime. But for guidance and visualization purposes, a solution for stakeholders to be aware of the state of the process in execution time would be to change corporate applications to store information when users interact with specific UIs to know where they are in the process by looking at the mappings. Consequently, there are other requirements that can be considered as an extension of this framework, as analyzed in the following section.

6.2 Future Work

This is an ongoing work that will be continued with future projects envisioned through collaborations with other universities, organizations and research institutions. The ideas of future works are organized as long and short term, according to some ongoing collaboration:

Short-Term:

Integration with different approaches for BP modeling. In a shorter term, integration of this framework with the ArtiFact^{TM}-based Business Operations Model [83]. One of the goals is to verify how a different BP modeling approach influences: the way UIs are designed to support these BP models, how these BP models are associated to UI models, and the work of business process performers as they interact with UIs of systems.

A new interest has arisen to include requirements in this alignment framework, with a model-driven approach to link business processes, UI models and requirements. A first proposal has been defined [115] and this reality encourages us to validate this new approach considering requirements in the industry, aiming at analyzing their interest on this strategy and their openness to change. The next step is to analyze requirements engineering modeling approaches in order to select a suitable strategy. 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.

In order to analyze the adoption of this methodology in large organizations, we have emphasized on bringing concrete results in terms of efficiency. Therefore, a new investigation is to demonstrate the effectiveness of the methodology by verifying its adherence to principles and values of the Agile Alliance, bringing fast results with its application, not adding bureaucracy, since it is a human-centered methodology that is concerned with what people say to improve their work. This future work is aimed to provide a comparative analysis of the methodology with agile principles. Since this is a methodology that embraces changes, is result-driven and focuses on end-users of enterprise systems, it is important to analyze the methodology under this perspective to verify how such agile principles and values are useful.

An emerging collaboration is the integration with an IT-Business alignment strategy. First, we consider a strategy that aligns business strategies with business processes and with IT architecture. Then, we consider our framework that aligns business processes with UI tasks and UI tasks with UIs of enterprise systems. Consequently, it is possible to bridge the gap between architecture and HCI from a new perspective. When any maintenance is made in the business process, the impact analysis can be broader by including architectural and implementation changes, but also including the UI, considering all subjective elements surrounding users and systems. It will complement this research by achieving alignment in terms of both the IT architecture and the user perspective. If services are considered in the IT architecture (e.g. using BPEL), when services are changed, we can indicate which UIs are impacted and vice-versa.

Long-Term:

Since we recognize that even though UI models are important for model-driven approaches, they are not widely known in the business environment, thus, this motivates us to study the cognition aspects of the models applied on the business context.

Once the platform is concluded, it needs to be largely applied in several situations and empirical data needs to be gathered in order to analyze its use. To do that, certain empirical metrics need to be established, such as time to learn, feasibility in the corporate environment and acceptance of the task model. The latter can also lead to a new research that evaluates the completeness of the task model in respect to user interaction possibilities (e.g. how to return to a previous task), which may lead to improvements in the representation of this model.

An empirical analysis can be made to evaluate how the association between task models and UIs and between UIs and end-users is made using the platform. This analysis shall consider how people perceive manually creating these links, how they collaborate with other to update the links, among other issues.

The rules can go through an improvement process when we will keep tracking which aspects in organizational scenarios motivate the adaptation of the rules to address specific contexts.

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.

There are several questions that should be answered to support decision-making as this framework is being applied. We have listed a set of prominent questions to be analyzed as the platform is developed: Is the user interaction a bottleneck for the performance of the process? Can the business process be improved based on efficient user interfaces that allow a faster execution of many activities? Are people navigating through the system in the same sequence of the business process or have they identified a better way to do the same thing? Which user interfaces are missing to be part of the scope of the next iteration? Using a catalog of user interfaces, which activities can be automated using existing user interfaces? Which user interfaces will be rarely used for the purpose of the organization and can be discarded or disabled from a product?

Overall, this research has achieved good results both in the academia and in the industry. Its characteristic of being multi-disciplinary and complementary to other strategies has made it an interesting theme for collaboration with researchers and practitioners from other communities.

Bibliography

[1] Aberdeen Group. Aligning IT to Business Process: How BPM is Complementing ERP and Custom Applications . Research Brief. Abardeen Group, May 2007.

[2] Aizenbud-Reshef, N., Nolan, B. T., Rubin, J., and Shaham-Gafni, Y. Model traceability. IBM Systems Journal, Vol. 45, No. 3, 2006, pp. 515-526.

[3] Ali, M.F., Perez-Quinones, 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.

[4] Ambler, Scott W. Agile Modeling. New York, John Wiley and Sons, 2002.

[5] Ambler, S. The Object Primer: Agile Model-Driven Development with UML 2.0: Agile Model-driven Development with UML 2.0. 3rd edition. Cambridge University Press, 2004. Available at: http://www.agilemodeling.com/artifacts/essentialUI.htm Accessed on: September 12, 2008.

[6] Aris Platform. Available at: http://www.ids- scheer.com/en/ARIS_ ARIS_ Platform/3730.html. Accessed on: January 27, 2010.

[7] Aversano, L. Bodhuin, T., and Tortorella, M. Assessment and impact analysis for aligning business processes and software systems. In Proceedings of Symposium of Applied Computing (SAC'05), 2005, pp. 1338-1343

[8] Bali, M. Drools JBoss Rules 5.0 - Developer's Guide. Packt Publishing, Birmingham, UK, 2009.

[9] Barbosa, S. D. J., Paula, M. G. de. Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering. In Proceedings of DSV-IS' 2003, Lecture Notes in Computer Science, Springer Verlag, 2003, pp. 16-33.

[10] Barclay, P. J., Griffiths, T., McKirdy, J., Kennedy, J. B., Cooper, R., Paton, N. W., Gray, P. Teallach - a flexible user-interface development environment for object database applications. Journal of Visual Language and Computing. Vol. 14, No. 1, 2003, pp. 47-77.

[11] Beaudouin-Lafon, M. Designing interaction, not interfaces. Proceedings of the working conference on Advanced visual interfaces, AVI'2004, Gallipoli, Italy, May 25-28, 2004. ACM Press, 2004. pp. 15-22.

[12] Belrose, G., Brand, K., Edwards, N., Graupner, S. Business-driven IT for SAP The Model Information Flow. Proceedings of the Second IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM'07), 2007.

[13] Bhattacharya, K. , Caswell, N., Kumaran, S., Nigam, A., Wu, F. Artifact-centric Operational Modeling: Lessons learned from customer engagements. IBM Systems Journal, Vol. 46, No. 4, 2007.

[14] Brachman, R. J., Levesque, H. J. Knowledge Representation and Reasoning. Elsevier, San Francisco, CA, 2004.

[15] 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, Vol. 15, No. 3, June 2003, pp. 289-308.

[16] Campos, P. and Nunes, N. Canonsketch: a User-Centered Tool for Canonical Abstract Prototyping. In Proceedings of DSV-IS'2004, 11th International Workshop on Design, Specification and Verification of Interactive Systems. Springer-Verlag, 2004.

[17] Cassell, P. The Giddens Reader. The MacMillan Press, London, 1993.

[18] CMMI for Development, Version 1.2. CMMI-DEV (Version 1.2, August 2006). Carnegie Mellon University Software Engineering Institute, 2006.

[19] Costa, D., Nobrega, L., Nunes, N. An MDA Approach for Generating Web Interfaces with UML ConcurTaskTrees and Canonical Abstract Prototypes. TAMODIA'06, 2006, pp. 137-152.

[20] Davenport, T. Process Innovation: Reengineering work through information technology, Harvard Business School Press, Boston, 1993.

[21] Decrop, A. Qualitative research methods for the study of tourist behavior, in Consumer Behavior in Travel and Tourism, 1999, pp. 335-365.

[22] Apache Derby. `http://db.apache.org/derby`, November 2008.

[23] Devaraj, S. and Kohli, R. Performance Impacts of Information Technology: Is Actual Usage the Missing Link?, Management Science Vol. 49, No. 3, 2003, pp. 273-289.

[24] Dix, A., Finlay, J. E., Abowd, G. D., Beale, R. Human-Computer Interaction. 3rd Ed. Pearson Education Limited, 2004.

[25] Eisenstein, J., Vanderdonckt, J., Puerta, A. R. Applying Model-Based Techniques to the Development of UIs for Mobile Computers. In: International Conference on Intelligent User Interfaces 2001 January 14-17, 2001, Santa Fe, New Mexico, USA, 2001, pp. 69-76.

[26] Enhanced Telecoms Operations Map (eTOM). Available at: http://www.tmforum.org/BestPracticesStandards/BusinessProcessFramework/6637/Home.html. Accessed on: January 27, 2010.

[27] Firesmith, D.G. and B. Henderson-Sellers. The OPEN Process Framework. An Introduction, Addison-Wesley, Harlow, Herts, 2002.

[28] Fischer, G. Giaccardi, E., Ye, Y., Sutcliffe, A. G., Mehandjiev, N. Meta-design: A Manifesto for End-User Development. Communications of ACM, Vol. 47, No. 9, 2004, pp. 33-37.

[29] Fleischmann, A., Lippe, S., Meyer, N., Stary, C. Coherent Task Modeling and Execution based on Subject-Oriented Representations. TAMODIA2009, Springer-Verlag, Berlin, 2009.

[30] Fowler, M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Addison-Wesley, 2003.

[31] Giddens, A. The Construction of Society. Political Press, London, 1984.

[32] Gonzalez-Perez, C., Henderson-Sellers, B. An Ontology for Software Development Methodologies and Endeavours. In Ontologies for Software Engineering and Software Technology, F. Ruiz, C. Calero and M. Piattini (eds.). Springer, 2006.

[33] C. Gonzalez-Perez, B. Henderson - Sellers, Metamodelling for Software Engineering, J. Wiley & Sons, Chichester, UK, 2008.

[34] Gotel, O.C.Z. and Finkelstein, A.C.W. An Analysis of the Requirements Traceability Problem. In Proc. of ICRE94, IEEE Computer Society Press, Los Alamitos, 1994, 94-101.

[35] Griffiths, T., Barclay, P.J., McKirdy, J., Paton, N.W., Gray, P.D., Kennedy, J.B., Cooper, R., Goble, C.A., West, A., Smyth, M. Teallach: A Model-Based User Interface Development Environment for Object Databases. In: Proc. of UIDIS99 (Edimburgh, 5-6 September 1999). IEEE Computer Society Press, Los Alamitos, 1999, pp. 86-96.

[36] Hailpern, B., Tarr, P. Model-driven development: The good, the bad, and the ugly. IBM Systems Journal, Vol. 45, No. 3, 2006, pp. 451-461.

[37] Hamilton, A. G. Logic for Mathematicians, Cambridge University Press, Cambridge, UK, 1978.

[38] Hartson, H. R., Gray, P. D. Temporal aspects of tasks in the user action notation. Human-Computer Interaction, Vol. 7, No. 1, 1992.

[39] Henderson, J. C., Venkatraman, N. Strategic Alignment: Leveraging information technology for transforming organizations. IBM Systems Journal, Vol. 38, No. 2&3, 1999.

[40] Henderson-Sellers, B., Gonzalez-Perez, C. Standardizing Methodology Metamodelling and Notation: An ISO Exemplar. UNISCON 2008, Information Systems and e-Business Technologies, Springer, Berlin, 2008, pp. 1-12.

[41] Henry, P. Process-User Interface Alignment: New Value From a New Level of Alignment. Align Journal. October 03, 2007. Available at: http://www.alignjournal.com/index.cfm?section=article&aid=446. Accessed on October 09, 2008.

[42] IBM, IBM Global Business Services, The New Voice of the CIO, 2009.

[43] IIBA, International Institute of Business Analysis, Business Analysis Body of Knowledge (BABOK), Version 1.6, 2006. Available at: http://www.theiiba.org/AM/Template.cfm?Section=Home. Accessed on October 09, 2008.

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

[45] ISO Information Processing Systems Open Systems Interconnection - LOTOS - A Formal Description Technique Based on temporal Ordering of Observational Behavior. ISO/IS 8807, ISO Central Secretariat, 1988.

[46] ISO/IEC 12207. Information technology software life cycle processes, International Standard Organization / International Electronic Commission, 1995.

[47] ISO13407, International Standard ISO 13407. Human-Centred Design Processes for Interactive Systems, ISO, Geneva, Switzerland, 1999.

[48] ISO/IEC: 24744: Software Engineering - Metamodel for Development Methodologies. International Organization for Standardization/International Electrotechnical Commission, Geneva, 2007.

[49] Jacobson, I., Booch, G., Rumbaugh, J. The Unified Software Development Process. Addison-Wesley, 1999.

[50] JBoss. Drools. `http://www.jboss.org/drools`, November 2008.

[51] Jersey. `https://jersey.dev.java.net`, November 2008.

[52] Jones, S. Enterprise SOA Adoption Strategies. C4Media Inc. Publisher, 2006.

[53] Johnson, P., Duberley, J. Understanding Management Research. Sage, London, 2000.

[54] Kaplan, R., Norton, D. The Balanced Scorecard: Translating strategy into action. Harvard Business School Press, Boston, 1996.

[55] Kleppe, A., Warmer, J., Bast, W. MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Professional, 2003.

[56] Kleppe, A. Software Language Engineering: Creating Domain-Specific Languages Using Metamodels. Addison-Wesley, 2008.

[57] Krajewski, L., Ritzman, L., Malhotra, M. Operations Management Processes and Value Chains. Pearson International Edition, 8th ed., NJ, 2007.

[58] Kristiansen, R., Traetteberg, H. Model-Based User Interface Design in the Context of Workflow Models. TAMODIA 2007, Springer-Verlag, Berlin, 2007, pp.227-239.

[59] Kumaran, S., Liu, R., Wu, F. Y. On the Duality of Information-Centric and Activity-Centric Models of Business Processes. In: 20th International Conference on Advanced Information Systems Engineering, CAiSE 2008, Springer-Verlag, Berlin, 2008, pp. 32-47.

[60] Kruchten, P. The Rational Unified Process. Addison-Wesley, NJ, 2000.

[61] Lakos, C. Object Oriented Modeling with Object Petri Nets. Concurrent Object-Oriented Programming and Petri Nets, 2001, pp. 1-37.

[62] Larsen, G. Model-driven development: Assets and reuse. IBM Systems Journal, 45(3), 2006, pp. 541-553.

[63] Lincoln, Y.S., Guba, E. Naturalistic inquiry. Beverly Hills, CA, Sage, 1985.

[64] 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, pp. 325-338.

[65] 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.

[66] Lievyns, A. User Interface and Business Process Alignment - A strategy-based approach in Telecom. Master in Business Administration. June 2009.

[67] Luftman, J. Assessing Business-IT Alignment Maturity. Communications of AIS, Volume 4, Article 14, 2000.

[68] Luftman, J., Kempaiah, R. Key Issues for IT Executives 2007. MIS Quarterly Executive. Vol. 7, No. 2, 2008.

[69] Magalhaes, R., Zacarias, M., Tribolet, J. Making Sense of Enterprise Architectures as Tools of Organizational Self-Awareness (OSA). Workshop on Trends in Enterprise Architecture Research (TEAR 2007), 2007.

[70] Mayhew, D. The Usability Engineering Lifecycle - A Practitioners Handbook for User Interface Design. Morgan Kaufmann Publishers, San Francisco, 1999.

[71] Mbaki, E., Vanderdonckt, J. Winckler, M. Multi-level Dialog Modelling in Highly Interactive Web Interfaces. In proceedings of the 7th International Workshop on Web-Oriented Software Technologies (IWWOST'2008). Gaedke, M., Bielikova, M. (eds.) Vydavatestvo STU Vazovova 5, Bratislava, Slovakia, 2008, pp. 44-49.

[72] Meisenheimer, C. Improving Quality: A Guide to Effective Programs. Jones and Barlett Publishers, 1997.

[73] Mendling, J., Neumann, G. Nttgens, M. A Comparison of XML Interchange Formats for Business Process Modelling. In Proceedings of EMISA 2004 - Information Systems in E-Business and E-Government. 2004, pp. 129-140.

[74] Mens, T., Gorp, P. V., Varró, D., Karsai, G. Applying a Model Transformation Taxonomy to Graph Transformation Technology. Electronic Notes in Theoretical Computer Science, Elsevier, 2006, pp. 143-159.

[75] Meyer, B. On Formalism in Specifications. IEEE Software, Vol. 3, No. 1. 1985, pp. 6-25.

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

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

[78] Mori, G., Paterno, F., Santoro, C. Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. IEEE Transactions on Software Engineering, August 2004, pp.507-520.

[79] Moura, A., Sauve, J., Bartolini, C. Research Challenges of Business-Driven IT Management. Proceedings of the Second IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM'07), 2007, pp. 19-28.

[80] NetBeans Visual Library. `http://graph.netbeans.org`, November 2008.

[81] Neubauer, M. Oppl, S., Stary, C. Towards Intuitive Modeling of Business Processes: Prospects for Flow- and Natural-Language Orientation. Proc. of TAMODIA2009, Springer-Verlag, Berlin, 2009.

[82] Nielsen, J. Usability Engineering. Academic Press, Cambridge, MA, 1993.

[83] Nigam, A. and Caswell, N. S. Business artifacts: An approach to operational specification. IBM Systems Journal, Vol. 42, No. 3 (Jul. 2003), pp. 428-445, 2003.

[84] OASIS, ebXML Technical Architecture Specification, 1.04, February, 2001.

[85] OASIS, Web Services Business Process Execution Language, Version 2.0, April, 2007.

[86] OMG, Object Management Group. http://www.omg.org.

[87] OMG, Business Process Modeling Notation Specification, 1.2, January, 2009.

[88] OMG, Model-Driven Architecture Guide, 1.0.1, June, 2003.

[89] Owen, M., Raj, J. BPMN and Business Process Management. Popkin Software, 2003.

[90] Paterno, F. Task Models in Interactive Software Systems. In "Handbook of Software Engineering & Knowledge Engineering, S. K. Chang (ed.), ISBN: 981-02-4973-X, World Scientific Publishing Co., 2001.

[91] Paterno, F., Santoro, C. One Model, Many Interfaces. Proceedings Fourth International Conference on Computer-Aided Design of User Interfaces, pp. 143-154, Kluwer Academics Publishers, Valenciennes, 2002.

[92] Paterno, 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.

[93] Penichet, V. M. R., Paterno, F., Gallud, J.A., Lozano, M.D. Collaborative Social Structures and Task Modeling Integration. In Proc. of DSVIS' 2006, Springer-Verlag, Berlin, 2007, pp. 67-80.

[94] Pontico, F., Farenc, C., Winckler, M. Model-Based support for specifying eService eGovernment Applications. TAMODIA 2006, EDM-Luc, Hasselt, 2006, pp. 43-50.

[95] Porter, M. Competitive Advantage. Free Press, New York, 1985.

[96] Prahalad, C.K., Krishnan, M.S. The New Age of Innovation: Driving Cocreated Value Through Global Networks, McGraw-Hill, USA, 2008.

[97] Puerta, A. R., Eisenstein, J. XIML: A Common Representation for Interaction Data. International Conference on Intelligent User Interfaces, 2002, pp. 216-217.

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

[99] W. Reisig and G. Rozenberg (Eds.). Lectures on Petri Nets I: Basic Models. Advances in Petri Nets, Springer, LNCS 1491, 1998.

[100] Rogers, E. M. Diffusion of Innovations. Free Press, 5 ed., New York, NY, 2003.

[101] 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, pp. 1228-1233.

[102] Rummler, A., Grammel, B., Pohl, C. Improving Traceability in Model-Driven Development of Business Applications. ECMDA Traceability Workshop, 2007.

[103] Russel, S., Norvig, P. Artificial Intelligence A Modern Approach. Prentice Hall, 2 ed., Upper Saddle River, NJ, 2003.

[104] Sauve, J., Moura, A., Sampaio, M., Jornada, J., Radziuk, E. An Introductory Overview and Survey of Business-Driven IT Management. Proceedings of the First IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM'06), 2006, pp. 1-10.

[105] Sharp, H., Rogers, Y., Preece, J. Interaction Design: beyond human-computer interaction. 2nd edition. John Wiley & Sons, 2007.

[106] Singh, I., Stearns, B., Johnson. M. Designing Enterprise Applications with the J2EE Platform, Second Edition. Addison-Wesley, 2002.

[107] Sinnig, D., Wurdel, M., Forbrig, P., Chalin, P., Khendek, F. Practical Extensions for Task Models. Proc. of TAMODIA2007, Springer-Verlag, Berlin, 2007, pp. 42-55.

[108] Smith, Ralph. Business Process Management and the Bal-anced Scorecard. John Wiley & Sons, New Jersey, 2007.

[109] Souchon, N., Vanderdonckt, J., A Review of XML-Compliant User Interface Description Languages, Proc. of 10th Int. Conf. on Design, Specification, and Verification of Interactive Systems DSV-IS'2003 (Madeira, 4-6 June 2003), Jorge, J., Nunes, N.J., Falcao e Cunha, J. (Eds.), Lecture Notes in Computer Science, Vol. 2844, Springer-Verlag, Berlin, 2003, pp. 377-391.

[110] Sousa, P., Caetano, A., Vasconcelos, A., Pereira, C., Tribolet, J. Enterprise Architecture Modelling with the Unified Modelling Language 2.0. In P. Ritten (Ed), Enterprise Modelling and Computing with UML. Hershey, PA: IRM Press, 2006.

[111] Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., Vandermeulen, J. User Interface Derivation from Business Processes: A Model-Driven Approach for Organizational Engineering. Proc. of 23rd Annual ACM Symposium on Applied Computing SAC2008, ACM Press, New York, 2008, pp. 553-560.

[112] Sousa, K., Mendonça, H., Vanderdonckt, J. Addressing the Impact of Business Process Changes on Software User Interfaces. Proc. of 3rd IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM'08), 2008, pp. 11-20.

[113] Sousa, K., Mendonça, H., Vanderdonckt, J. User Interface Development Lifecycle for Business-Driven Enterprise Applications. 7th International Conference on Computer-Aided Design of User Interfaces CADUI08, 2008, pp. 23-34.

[114] Sousa, K., Mendonça, H., Vanderdonckt, J. A Model-Driven Approach to Align Business Processes with User Interfaces, International Journal of Universal Computer Science, Special issue on Human-Computer Interaction. Vol. 14. No. 19, 2008, pp. 3236-3249.

[115] Sousa, K., Mendonça, H., Vanderdonckt, J., Pimenta, M. S. Supporting Requiremetns in a Traceability Approach between Business Processes and User Interfaces. Short Paper on the 8th Symposium on Human Factors in Computer Systems IHC'08, 2008, pp. 272-275.

[116] Sousa, K. Model-Driven Approach for User Interface - Business Alignment. Proc. of 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems EICS2009, ACM Press, New York, 2009, pp. 325-328.

[117] Sousa, K., Mendonça, H., Vanderdonckt, J. A Rule-Based Approach for Model Management in a User Interface Business Alignment Framework. In: TASK MODEL AND DIAGRAMS FOR USER INTERFACE DESIGN, TAMODIA2009, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2009, pp. 1-14.

[118] OMG, Software Process Engineering Metamodel Specification, v2.0, February 2007.

[119] Stoitsev, T., Scheidl, S., Flentge, F., Mühlhäuser, M. From Personal Task Management to End-User Driven Business Process Modeling. 6th International Conference on Business Process Management BPM 2008, Springer, 2008, pp. 84-99.

[120] Stolze, M., Riand, P., Wallace, M., Heath, T. Agile Development of Workflow Applications with Interpreted Task Models. TAMODIA'2007, 2007, pp.2-14.

[121] Streaming API for XML. `http://www.jcp.org/en/jsr/detail?id=173`, Nov 2008.

[122] Strong, D. M., Volkoff, O. A Roadmap for Enterprise System Implementation. Computer, vol. 37, no. 6, pp. 22-29, Jun., 2004.

[123] 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 MDDAUI05 (Montego Bay, Jamaica), CEUR Workshop Series, Vol. 159, 2005.

[124] Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S., and Stolze, M. User-centered Design & Business Process Modeling: Cross Road in Rapid Prototyping Tools. In Proc. of Interact07, Springer-Verlag, Berlin, 2007, pp. 165-178.

[125] Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S. Model-Driven Approach for Managing Human Interface Design Life Cycle. MoDELS 2007, 2007, pp. 226-240.

[126] Sukaviriya, N., Mani, S., Sinha, V. Reflection of a Year Long Model-Driven Business and UI Modeling Development Project. In Proc. of Interact'09, Springer-Verlag, Berlin, 2009, pp. 749-762.

[127] Sujitparapitaya, S., Janz, B. D., Wetherbe, J. C., Sammet, D. Ascension Health Systems: Enterprise user Interface Approach to Organizational Data Management. In 34th Hawaii International Conference on System Science, 2001.

[128] Sun Microsystems. Swing. `http://java.sun.com/javase/6/docs/technotes/guides/swing/index.html`, November 2008.

[129] Taylor, S. J., Bogdan, R. Introduction to Qualitative Research Methods: The Search for Meanings. John Wiley & Sons, 1984.

[130] Traetteberg, H. UI Design without a Task Modeling Language - Using BPMN and Diamodl for Task Modeling and Dialog Design. Proc. of HCSE/TAMODIA2008, LNCS, Springer-Verlag, Berlin, 2008, pp. 110-117.

[131] Vanderdonckt, J. A MDA-Compliant Environment for Developing User Interfaces of In-formation Systems. In: Proc. of CAiSE'05. Lecture Notes in Computer Science, Vol. 3520, Springer-Verlag, Berlin, 2005, pp. 16-31.

[132] Vanhooff, B., Van Baelen, S., Joosen, W., Berbers, Y. Traceability as Input for Model Transformations. ECMDA Traceability Workshop (Haifa, Israel), 2007.

[133] Van Mieghem, J. A. Operations Strategy Principles and Practice. Dynamic Ideas, Belmont, MA, 2008.

[134] 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 Fifth International Enterprise Distributed Object Computing Conference (EDOC 2001), IEEE Computer Society Press, Los Alamitos, 2001, pp. 69-81.

[135] Vasconcelos, A., da Silva, M. M., Fernandes, A. Tribolet, J. M. An Information System Architectural Framework for Enterprise Application Integration. In Proc. of Thirty-Seventh Hawaii International Conference on System Sciences (HICSS 2004). Hawaii, 2004.

[136] Yin, R. Case Study Research - Design and Methods. Third Edition. Sage Publications, California, 2003.

[137] Zacarias, M., Caetano,A., Magalhaes, R., Sofia Pinto, H., Tribolet, J. Adding a Human Perspective to Enterprise Architectures. DEXA -

WEISE 2007 - International Workshop on Enterprise Information Systems Engineering. Germany, 2007, pp. 840-844.

[138] Zacarias, M., Serendero, P.,Pinto, H. S. Tribolet, J. M. Capturing and Modelling Work Practice. A Context-based Methodological Approach. Revue d'Intelligence Artificielle, Vol. 22, No. 5, 2008, pp. 669-688.

[139] Zacarias, M., Pinto, H. S., Magalhaes, R., Tribolet, J. M. A 'context-aware' and agent-centric perspective for the alignment between individuals and organizations. Information Systems, Vol. 35, No. 4, 2010, pp. 441-466.

[140] Zachman, J. A. A Framework for Information System Architecture. IBM Systems Journal, Vol. 26, No. 3, 1987.

[141] Zelkowitz, M. V. An update to experimental models for validating computer technology. Journal of Systems and Software 82(3), 2009, pp. 373-376.

[142] Wagner, H-T., Beimborn, D., Franke, J., Weitzel, T. IT Business Alignment and IT Usage in Operational Processes: A Retail Banking Case. In Proc of HICSS2006, IEEE Computer Society Press, Los Alamitos, 2006, pp. 172-173.

[143] WfMC. Workflow Management Coalition, Terminology & Glossary. February, 1999.

[144] WfMC. Process Definition Interface - XML Process Definition Language, Version 2.1a. October, 2008.

[145] Wolff, A., Forbrig, P., Dittmar, A., Reichart, D. Linking GUI elements to tasks: supporting an evolutionary design process. In: Proc. of TAMODIA2005. ACM Press, 2005, pp. 27-34.