# AN APPROACH TO INTEGRATE HCI AND SE IN REQUIREMENTS ENGINEERING

**Kenia Soares Sousa[1], Elizabeth Furtado[2]**
[1]Mentores Consultoria - Rua Joao Carvalho, 800 - S.511 –CE – Brasil
[2]Universidade de Fortaleza - Av. Washington Soares, 1321 – CE – Brasil

kenia@mentores.com.br, elizabet@unifor.br

**Abstract:** This research work intends to present three workflows from a new Software Development Process (SDP) that, besides focusing on cost and schedule, also includes some Human-Computer Interaction (HCI) aspects along the software life cycle. Such HCI aspects include usability, accessibility, acceptability requirements, guidelines application, model-based User Interface (UI) generation techniques, and evaluation techniques. This study is based on the world-wide renowned Rational Unified Process (RUP), which is a basis for the definition of this new SDP, called the UPi (Unified Process for Interactive systems). The purpose of this integration is to develop interactive systems that are easy to learn and use, therefore, helping users in performing their daily tasks in an efficient manner. Besides that, this integration intends to bring together researchers from the HCI and Software Engineering (SE) areas of study, which started to be a concern in the research community in the last years.

**Keywords:** Human-Computer Interaction, Software Engineering, Interactive Systems, Software Development Process, RUP.

## 1 Introduction

This research work intends to present a way to integrate experts from the HCI and SE fields while working in a Software Development Process (SDP). A software development organization can apply a SDP that only focuses on delivering software on the pre-defined cost and schedule, and that accommodates users' expectations related to functionalities. On the other hand, a software development organization can choose to apply a SDP that focuses on the users' characteristics, the tasks they perform, the work environment where they are located, and the devices they use to perform their tasks. On the first case, most activities performed during the software lifecycle are related to: scope and risk control; functionalities definition; components design, implementation, and integration; functionalities tests; change control; tools definition; and documentation development. On the second case, the focus is more closely related to human factors (Shneiderman, 1998), usability (Nielsen, 1993), acceptability, accessibility, guidelines, evaluation techniques (Preece, 1994) and UI modeling. These aspects, when applied in a SDP, may bring benefits such as decrease in time to learn how to use the software, increase in the speed in performing tasks using the software, and increase in the user satisfaction towards the software.

This work is a continuous research on the integration of HCI aspects in SE, more specifically in the SDP. Our intention is to associate both areas in order to increase user satisfaction from the use of software derived from a user-centered SDP. We exposed our first ideas related to the addition of HCI aspects in only four RUP workflows in (Sousa, 2003a), and detailed and exemplified this work in (Sousa, 2003b). We extended this work by applying HCI aspects in all of the RUP workflows in (Sousa, 2003c). Now, we focus on the UPi business modeling, requirements, and analysis and design workflows in order to achieve better results in terms of trying to help HCI and SE experts to work as a team. Thus, this research focus on the workflows performed early in the life cycle in order to provide developers with a solid basis for implementing usable systems.

We intend to demonstrate that HCI and SE experts can work together efficiently, that is, their work can complement each other's work. Since most software development companies in the market-place do not even know HCI concepts, we intend to insert HCI activities, workers, and artifacts in a SDP in an easy manner. Considering that the RUP is applied worldwide, we intend to provide those organizations a way to easily insert HCI in the SDP that they already apply in their software projects. Focusing on facilitating such application, we only recommend new activities, artifacts, and workers; maintaining the UPi workflows the same ones as in the RUP.

In this research work, we present how the RUP suggests the performance of three workflows (Kruchten, 2000), and we recommend the participation of HCI experts in those workflows. Each workflow is composed of workflow details, which have a set of activities to be performed by workers that produce valuable results, called artifacts.

## 2 Related Work

(Kruchten, 2001) presents the concern on UI design in the RUP by including the worker UI Designer, responsible for performing the activities UI Modeling and UI Prototyping in the Requirements Workflow. In the activity UI Modeling, the UI Designer analyzes the use case model, and creates a use case storyboard. This artifact is composed of a textual description of the interaction between the user and the system, interaction diagrams, class diagrams, usability requirements, references to the UI prototype, and a trace dependency to the use case. In the activity UI Prototyping, the UI Designer designs and implements the UI prototype, then obtains feedback from other project members, external usability experts, and users. Even though there is a concern on the UI design, there is no concern on the specification of users' tasks. Use cases differ from task descriptions in the sense that they do not provide details related to purpose, preconditions, frequency, and post conditions. Such specification is not a use case goal, but it is responsibility of task descriptions. We suggest use case and task modeling to be used in a complementary manner.

(Phillips, 2002) suggests the use of tabular representation of use cases in order to describe the flow of events, and the use of UI element clusters, which can be used as references to the UI prototype. Tabular use cases separate user and system actions. (Lauesen, 2001) argues that separating users' and systems' actions as early as in requirements may be a barrier for future decisions in the project. Therefore, he suggests the use of task descriptions, which specify what the users and the system shall do, not dividing the work between them. On the other hand, (Constantine, 1999) focuses on preparing task cases that address users' intentions rather than their actions and on system responsibilities rather than on responses. This approach offers a requirement modeling more technology-independent, providing designers the flexibility related to the actual interface components when designing the UI, but maintaining conformity to users' needs.

There are other research works related to extending the RUP, such as (Piveta, 2002) and (Massoni, 2002), but they concern changes in the process regarding aspect-oriented principles. Piveta suggests changes in the Analysis and Design Workflow in order to suit the modeling techniques necessary for the Aspect-Oriented Software Development. Massoni suggests changes in the Implementation Workflow by creating the Progressive Implementation Method, which provides the progressive implementation of three different aspects: persistence, distribution, and concurrency. Therefore, none of their extensions affect the generation of usable UIs.

## 3 The Business Modeling Workflow

The RUP business modeling workflow focuses on establishing a common understanding of the target organization in order to derive the software requirements. The understanding of the organizational structure (e.g. processes, roles, etc.) has an impact on the delivered product, which should help workers in accomplishing their daily tasks. This workflow is depicted in figure 1.

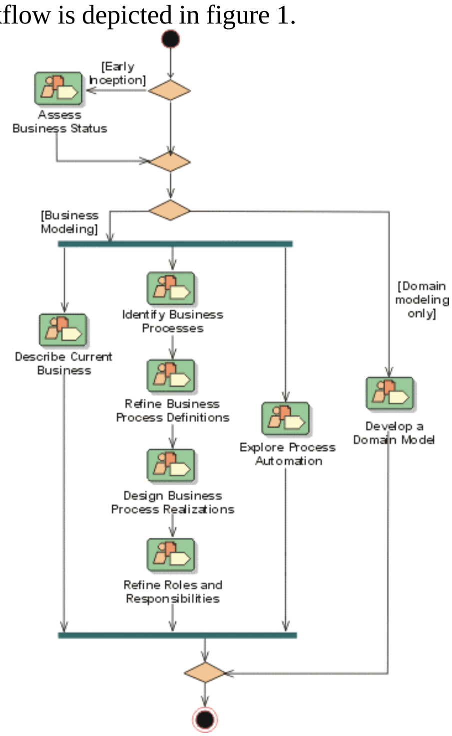


**Figure 1:** The Business Modeling Workflow from (RUP, 2003)

Most software development organizations perform the previously mentioned activities as part of the requirements workflow. Nevertheless, the RUP makes a clear distinction between these two workflows. The business modeling workflow results in an understanding of the organization, and the requirements workflow uses this outcome as an input in order to better define the system scope. Eventhough, they have distinct purposes, they are complementary, and their activities can be performed in parallel in order to provide a faster development process.

In the UPi, we suggest the focus on users' characteristics and environment, since they influence designers in better understanding the organizational structure. In order to fulfill this goal, two workflow details are changed:

- In the "Assess Business Status" workflow detail, the business-process analyst is responsible for understanding the target organization, that is, its vocabulary, goals and business rules, and documenting them on specific documents.

In the UPi, we recommend the business-process analyst and the UI designer to prepare the context of use model (figure 2). The first one collaborates with the knowledge on the business, and the latter collaborates with the knowledge on UI modeling. This model represents the environment where users are located in order to perform their tasks. The context of use model can be generated by the analysis of the environment where the interactive system will operate (e.g. analysis of users' surroundings, situation, and location). This model will be important to adapt the system's physical UI based on environment's constraints, especially for mobile devices that are used in a wide variety of locations. For instance, a palm will be necessary for a system while users work walking. The context of use model representation is based on the business use case model, and on scenarios' descriptions, which represent real situations faced by users while performing their daily tasks.

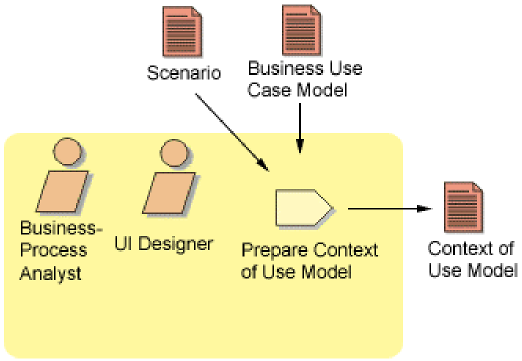


**Figure 2:** Extra Activities in the Assess Business Status Workflow Detail

- In the "Refine Roles and Responsibilities" workflow detail, the business designer is responsible for defining details of business workers and entities.

In the UPi, we recommend the business designer to use the user model while detailing business workers (figure 3). The user model, obtained from the requirements workflow, represents users' characteristics, focusing more on human factors than on work responsibilities. This addition brings a perspective related to users' reality, different from a business point of view. Considering that this workflow is performed in parallel with the requirements workflow, it is important to mention that this workflow detail needs to be performed after defining the system in the requirements workflow.

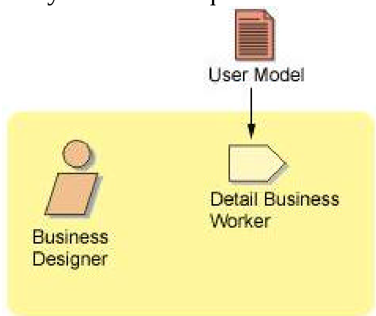


**Figure 3:** Extra Activities in the Refine Roles and Responsibilities Workflow Detail

# 4 The Requirements Workflow

The RUP requirements workflow has the purpose of defining the system scope. The system scope needs to be based on the detection of problems in the target organization and on understanding users' needs. Then, this scope needs to be managed, and refined in order to reflect users' changing requests. This workflow is depicted in figure 4.

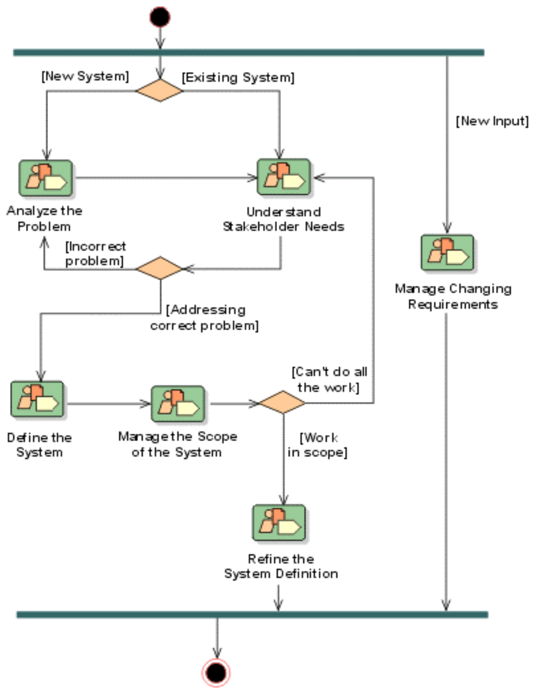


**Figure 4:** The RUP Requirements Workflow from (RUP, 2003)

The UPi requirements workflow focuses on users' needs, and requires the addition of activities that are related to the production of artifacts that are valuable to the generation of usable systems. For this reason, two workflow details in the RUP are changed, that is, HCI activities are added into them.

Before stating the changes suggested in the workflow details, it is important to explain the "Understand Stakeholder Needs" workflow detail. In this workflow detail, the system analyst is responsible for eliciting stakeholder requests in order to understand and document them, and prepare the use case model. Most of these requests are related to the system functionality; therefore, it is recommended that non-functional requirements are textually registered in a different document. Instead of that, we suggest the registration of non-functional requirements in the use-case, user, and task models, which are generated in the following workflow detail.

- In the "Define the System" workflow detail, the system analyst is responsible for analyzing the stakeholders' requests, and refining the use case model.

In the UPi, we, otherwise, recommend the participation of the UI designer, and the ergonomist in this workflow detail in order to produce the use case, the user, and the task model (figure 5). It becomes easier to achieve better results when users participate in these modeling activities.

This way, the system analyst provides the information that reflects stakeholder's needs as a basis for the UI designer and the ergonomist to prepare those models that are useful throughout the development process, especially in the UI generation.

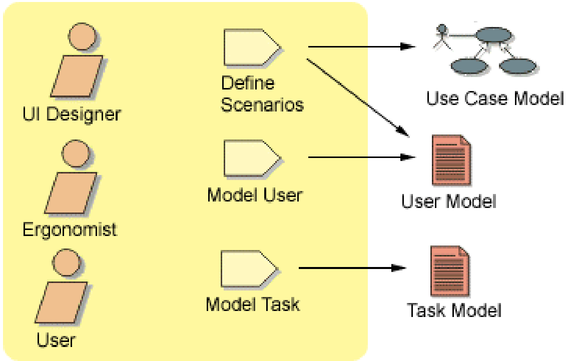


**Figure 5:** Extra Activities in the Define the System Workflow Detail

We recommend the preparation of scenarios, which are useful to define the use case model, the user model, and the context of use model, prepared in the business modeling workflow. Scenarios guarantee that the use case model represents users' reality, in terms of what are their real needs towards the system, not what systems analysts think the system should do.

The user model specifies characteristics of users who will interact with the system. The user model defines the scope of the user population, including characteristics, such as intellectual and physical abilities, previous experience with using interactive systems and performing the required task. Its greatest advantage is the impact of users' classification on the UI suitability to their individuality.

We also recommend the preparation of the task model, which defines the tasks performed by users. The task model is a hierarchical structure of tasks and sub-tasks graphically represented. In this work, we used the MAD formalism (Pierret-Golbreich, 1989), in which a task is divided in two or more subtasks that are performed in order to fulfill a certain objective. The task model is generated based on the use case model, defined during this workflow, which defines system's processes. The task model is useful to allow the system analyst to envision what tasks users perform, predict difficulties, evaluate systems, measure system complexity, and predict performance, among other aspects.

- In the "Refine the System Definition" workflow detail, the requirements specifier is responsible for detailing software requirements, and documenting the results in the software requirements specification and in the supplementary specifications documents. Since these requirements are focused on functionality, we enhance this workflow detail with activities that result in HCI artifacts. Therefore, the requirements specifier has its artifacts complemented with information generated by the UI designer and the ergonomist.

In the UPi, we recommend the participation of the UI designer in this workflow detail in order to analyze the UI models generated in the previuos workflow detail aiming at preparing a usability requirements list (figure 6). These requirements are related to users' satisfaction and the performance of the system. Usability requirements directly influence aspects of the system quality of use (e.g. easy access to shortcuts, number of data users need to remember, etc.), because the guidelines applied in the UI during the implementation workflow, and the system evaluation during the test workflow are based on these requirements. These requirements can be gathered by analyzing the user, the task, and the context of use models prepared during this workflow; and they are organized in use cases. One usability requirement can be associated to many use cases. So, the matrix that represents the association of requirements with use cases is updated after each analysis.

Besides that, we also recommend the participation of the UI designer, and the ergonomist in order to define usability recommendations based on the users, their tasks, their working environment, and the system domain (figure 6), thus, generating guidelines. It becomes easier to achieve better results when users participate in these definitions.

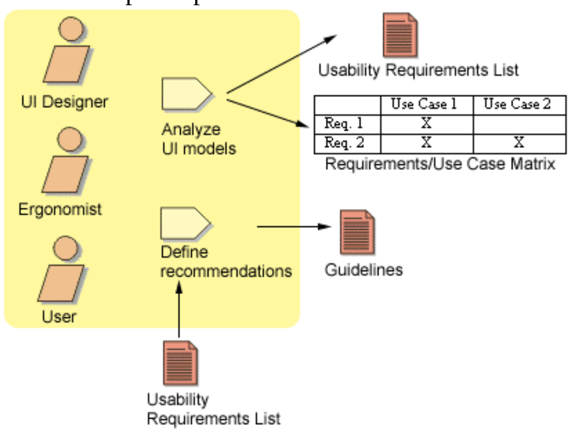


**Figure 6:** Extra Activities in the Refine the System Definition Workflow Detail

Guidelines definitions are important to provide consistency in the UI, increasing user productivity, decreasing number of errors, among other aspects related to usability. Guidelines defined during the requirements workflow are related to the system conceptual level, and are used as a basis for the definition of guidelines related to other workflows. For example, the guideline "use metaphors for operations" is useful to define the guideline "use a printer image for the print operation", necessary for the UI generation during the implementation workflow. Guidelines are defined by UI designers based on usability requirements. For instance, a use case "system maintenance" for any system that needs internal control, it could have "maintenance speed" as one usability requirement. Thus, the generating the following semantic guideline: use metaphors that group together actions that are familiar to users.

## 5 The Analysis and Design Workflow

The RUP analysis and design workflow has the purpose of transforming requirements into a design specification useful for the implementation workflow. This workflow is depicted in figure 7, in which the software architect is responsible for defining, and refining the system architecture. The system architecture is based on components, which can be tested individually and integrated gradually to compose a complete system.

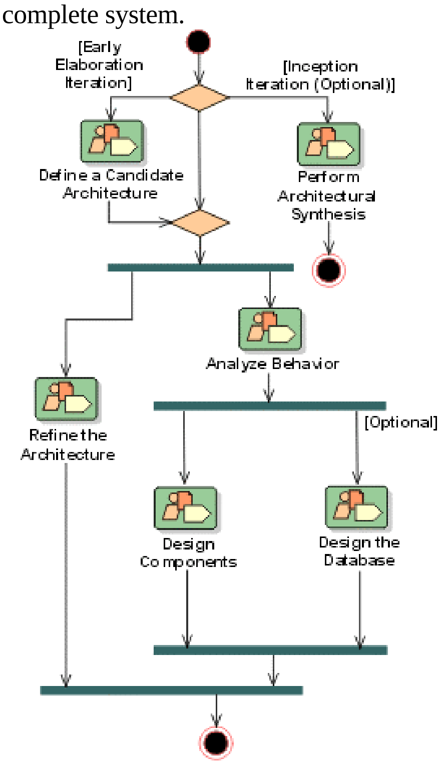


**Figure 7:** The RUP Analysis and Design Workflow from (RUP, 2003)

The UPi analysis and design workflow uses the models prepared in the requirements and business modeling workflows in order to generate more usable UIs. With this goal in mind, HCI activities are included in three existing workflow details:

- In the "Analyze Behavior" workflow detail, the software architect is responsible for analyzing and transforming use case behavioral descriptions, and the interactive characteristics of tasks (e.g. sequence and repetition control) associated to the use case model into elements useful for design.

In the UPi, we recommend the participation of the UI designer in order to refine the task model, generated in the requirements workflow (figure 8). Such refinement may be done by considering characteristics of different kinds of users and different interaction styles.

This way, the software architect is identifying the system classes, and detailing use-case flow of events in terms of the system functionality, but also, taking into consideration the possibilities of system adaptation based on users specific characteristics.

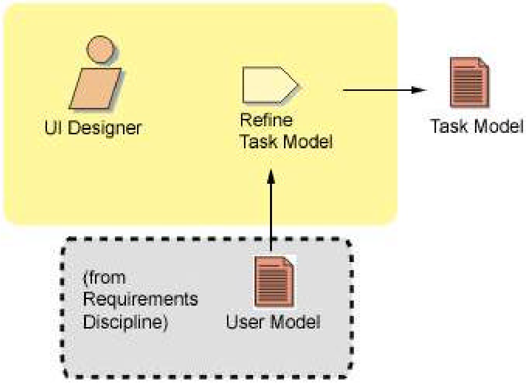


**Figure 8:** Extra Activities in the Analyze Behavior Workflow Detail

- In the "Refine the Architecture" workflow detail, the software architect is responsible for defining the architeture by focusing on implementation and deployment.

In the UPi, we recommend the participation of the UI designer in order to prepare the interface conceptual model, and analyze UI design options (figure 9). The Interface Conceptual Model (MIC) (Furtado, 2002) that represents the interactive part of the system by graphically representing the navigational structure of the system. The MIC is generated based on the task model, on the context of use model, and on a set of guidelines applied to translate the task model into the MIC. For instance, there is one guideline that states that if a task in the task model shall only be performed when a user validates it, then two interaction objects have to be created in the MIC (e.g. confirm and cancel). There is another one that states that each interruptible task in the task model shall originate an interactive space in the MIC. The production of the MIC can generate design options because the UI designer is able to prepare this model for different kinds of users, e.g. experts and novices. The design options are generated using the MIC. This analysis is performed by the association of usability requirements to design options of a specific task in order to help the UI designer when trying to choose, along with the users, the best option.

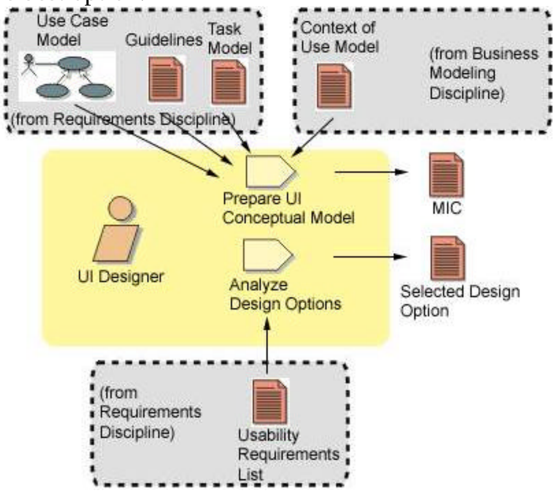


**Figure 9:** Extra Activities in the Refine the Architecture Workflow Detail

As a result of the participation of the software architect and the UI designer, we have a system that is well-structured in terms of its functionality, and usability.

- In the "Design Components" workflow detail, the class characteristics and relationships are completely specified.

In the UPi, we recommend the class model to be enhanced by being specified in a n-tier architecture (figure 10). In such a class model, the first layer represents the interface, the second one is the control layer, the third is the business logic layer, and the fourth one is the data layer. This approach allows the accordance to the independence dialogue principle (Lafon, 1991), which means modeling the system application independent from the system UI. The main goal is to facilitate changes in these layers. This way, a change in the UI will not affect the application, and vice-versa.

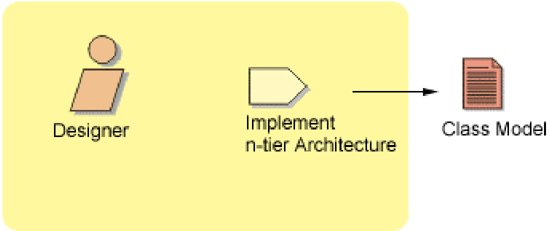


**Figure 10:** Extra Activity in the Design Components Workflow Detail

# 6 Conclusion

This research work presented three UPi workflows, the business modeling, requirements, and analysis and design workflow, focusing on how to integrate HCI and SE workers, activities, and artifacts. The Business Modeling Workflow is enhanced by focusing on users´ characteristics and environment. The Requirements Workflow is enhanced by focusing on model-based UI generation, on usability goals, and with the application of model guidelines. The Analysis and Design Workflow is enhanced by focusing on different kinds of users and interaction styles, dialogue independence, and usability requirements.

The focus of HCI aspects in the UPi workflows brings benefits both to the HCI and SE experts. The first ones benefit from system functionality definition, leading to better precision useful for the UI definition. The latter benefit from usability, acceptability, and accessibility aspects, leading into user satisfaction.

As for future work, we intend to continue this closer integration in the other UPi workflows. Besides that, we intend to suggest a model that integrates the most important characteristics from the user, task, context of use, and use case model. In other words, we want to visually demonstrate how specific users intend to perform their tasks in the envirnonment where they are located.

# References


Constantine, Larry; Lockwood, Lucy. (1999), *Software for Use: A Practical Guide to Models and Methods of usage-Centered Design.* Massachusetts: Addison-Wesley.

Furtado, Elizabeth; Sousa, Kênia; Cólera, César. (2002), An Environment to Support Developers in Elaborating a Participatory and Evolutionary Help on Style Guide. *In: Seminário em Fatores Humanos para Sistemas de Computação, 2002, Fortaleza. Usabilidade: um direito.* Fortaleza: Banco do Nordeste, pp. 84 – 92.

Kruchten, Philippe. (2000), *The Rational Unified Process - An Introduction.* 2 ed. New Jersey: Addison-Wesley.

Kruchten, Philippe; Ahlqvist, S; Bylund S. (2001), *User Interface Design in the Rational Unified Process, in Object Modeling and User Interface Design.* Massachusetts: Addison-Wesley.

Lafon, M. (1991), Interfaces Homme-machine : Vue d'ensemble et perspectives. *Actes du congrès Génie logiciel & Systèmes Experts, Interfaces homme-machine*, Maquettage & prototypage. Vol. 24, pp. 4-16.

Lauesen, Soren. (2001), Task & Support - Task Descriptions as Functional Requirements. *In: Proceedings of AWRE*, pp. 83-91.

Massoni, Tiago; Sampaio, Augusto; Borba, Paulo. (2002), A RUP-Based Software Process Supporting Progressive Implementation. *In: 2002 International Conference of the Information Resources Management Association (IRMA'2002)*, pp. 480-483.

Nielsen, Jakob. (1993), *Usability Engineering.* Califronia : Morgan Kaufmann.

Phillips, Chris; Kemp, Elizabeth. (2002), In Support of User Interface Design in the Rational Unified Process. *In: Proceedings of the Third Australasian User Interface Conference*, pp. 21-27.

Pierret-Golbreich, Christine.; Scapin, Dominique. (1989), MAD: Méthode Analytique de Description des tâches. *Colloque sun língénierie des interfaces homme-machine.* Sophia-Antipolis.

Piveta, Eduardo Kessler; Devegili, Augusto Jun. (2002), Aspects in the Rational Unified Process' Analysis and Design Workflow. *In: Proceedings of the Aspect Oriented Design (ADO).*

Preece, Jenny et al. (1994), *Human-Computer Interaction.* England: Addison-Wesley.

RUP. (2003), *The Rational Unified Process.* Available in: <http://www.rational.com/products/rup/tryit/eval/gen_eval.jsp>. Accessed in 23 apr. 2003.

Shneiderman, Ben. (1998), *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* 3 ed. England: Addison-Wesley.

Sousa, Kenia; Furtado, Elizabeth. (2003a), Integration of Human-Computer Interaction in a Software Development Process. *In: Human-Computer Interaction International* – HCII'2003.

Sousa, Kenia; Furtado, Elizabeth. (2003b), RUPi – A Unified Process that Integrates Human-Computer Interaction and Software Engineering. *In: International Conference on Software Engineering – ICSE'2003*, pp. 41-48.

Sousa, Kenia; Furtado, Elizabeth. (2003c) A Unified Process for Interactive Systems. *In: Latin American Conference on Human-Computer Interaction – CLIHC'* 2003.