Publicaciones en Ciencias y Tecnología. Vol 5, 2011 N^0 2, pp.43–52, ISSN:1856-8890,Dep.Legal pp200702LA2730

A Service-Oriented approach for interactive computing systems

*Jorge Luis Pérez Medina *Sophie Dupuy-Chessa *Dominique Rieu

Recibido: 20 de Junio 2011, Aceptado:10 de Diciembre 2011

Abstract

The introduction of new technologies leads to a more and more complex interactive systems design. In order to describe the future interactive system, the human computer interaction domain uses specific models, design processes and tools in order to represent, create, store and manipulate models. The aim of our work is to provide a theoretical conceptual approach to facilitate the work of model designers and project managers by helping them in choosing processes, models and modeling environments adapted to their specific needs. This paper details the use of a service-oriented approach for model management. Our proposals are related to three different abstract levels: the operational level to choose the appropriate modelling tool, the organisational level to select a design process and the intentional level to define modelling goals.
Keywords: HCI, model, service, model management, modeling tools, modeling services.

Un enfoque basado en servicios para sistemas informáticos interactivos
Resumen

La introducción de nuevas tecnologías conduce a un diseño de sistemas interactivos cada vez más complejo. Para describir el futuro sistema interactivo, el dominio de la interacción humano-computador usa modelos específicos, procesos de diseño y herramientas para representar, crear, almacenar y manipular modelos. El objetivo de nuestro trabajo es proporcionar un enfoque teórico-conceptual para facilitar el trabajo de los diseñadores de modelos y jefes de proyectos, ayudándoles en la selección de procesos, modelos y ambientes de modelado adaptados a sus necesidades especificas. Este artículo detalla el empleo de una propuesta a base de servicios para la gestión de modelos. Nuestras proposiciones se fundamentan sobre tres niveles de abstracción: el nivel operacional para escoger la herramienta de modelaje adecuada, el nivel organizacional para seleccionar un proceso de diseño y el nivel intencional para los objetivos de modelaje.
Palabras clave: Interacción humano-computador, gestión de modelos, herramientas de modelaje, servicios de modelaje.

Introduction

Model-based approaches aim at helping developers understand user needs and design solutions in an effective, users oriented way. In the Human Computer Interaction (HCI) domain, interactive systems are increasingly complex: they can use everyday life objects to propose tangible interfaces; they can couple the virtual and the physical worlds in augmented reality systems; they can adapt themselves to the user context, etc. Then they are increasingly difficult to design. Nowadays substantive efforts have been devoted to the definition and use of models, and extensive development of software support has been achieved. The HCI community uses different models to support the design of interactive systems. In particular, the HCI design is often based on task analyses, which are classically represented by task trees. In addition, the HCI community proposes specific models such as the Adaptor, System, User, Real

* *CNRS,LIG 385 rue de la bibliothque, University of Grenoble , BP 53 F-38041 GRENOBLE cedex 9, FRANCE, jorge-luis@imag.fr, sophie.dupuy@imag.fr, dominique.rieu@imag.fr*

object notation (ASUR) [9] and the Interacting with Real and Virtual Objects (IRVO) [6], that take into account augmented reality interactions in the context of a user task.

Moreover the use of these models can be guided by specific processes. Many interactive system design methods are proposed. They are often based on task analysis [17, 21, 23]. Many efforts are also related to contextual design [2, 11] and scenario-based approaches [7, 19]. The choice of such processes is a strategic decision that depends on the goals expected by the model designers (e.g. design an Augmented Reality system). It needs to be adapted to the specifications and features of projects. In addition, the user interface design is a process in which professionals feel the need for flexibility in their work to overcoming the growing complexity of interactive system [22] since models and process are no consensual [15]. To face these needs of adaptation and flexibility in the design, we propose a theoretical-conceptual approach based on services to helping model designers and project managers in the choice of models and processes according to their modeling goals. The choice of a process determines the models to use, and then their modeling environments. For instance, selecting the process proposed by [8] to design an Augmented Reality system lead to the use of the ASUR language and its tool support GUIDE-ME [24]. Our approach is based on theoretical concepts for reuse of existing processes, models and tools in order to find solution to the goals of designers and managers. It uses the notion of services [3] for the model management. Modeling services are defined to express modeling goals and automate software development steps using models and tools [15]. So, our proposals are based on three abstract levels: the operational level presented in [15], the organizational level and the intentional level: 1) operational services carry out automated operations (e.g. editing a model); 2) organizational services propose fragments of design methods i.e. processes; 3) the intentional services correspond to the goals proposed by any person or organization handling models. This paper focuses on the intentional and the organization levels which are the main contributions of our work. This paper is organized as follows. Section 2 describes two experimental scenarios. These examples are based on interactive system design methods. Section 3 presents our approach based on service-oriented models management. Models of the intentional layer and the organizational layer are presented. Finally, section 4 contains our conclusions and perspectives.

Interactive system design methods

To explain our approach, we focus on two interactive system design methods. The reasons for which we use these methods are that we know well and they are part of the researches realized in our research team. Their goal is to demonstrate how our service-oriented approach can be applied to interactive system design.

The first method is an extension of the Symphony design method [12], used as a medium for merging HCI and software engineering development processes.

Symphony is a user-oriented, business component-based development method originally proposed by the UMANIS Company. It has been extended lately to integrate the design of Augmented Reality systems [10]. In this article, we concentrate on the phase of the "Organizational and Interactional Specifications of Requirements" (Figure 1). This phase aims to analyze a business process at organizational level ("who does what and when" of the future system), and to envisage a satisfactory interaction to carry out activities of different stakeholders. Concerning the HCI aspects (box of Figure 1), the activities proposed by the design method are: **description of task models** to clarify the interactions of the internal actors with the system, **external specification of the interaction** to define the user interface (UI) and its supporting devices and finally **structuration of Interaction concepts into a specific model** composed of reusable components **called Interactional Objects (IO)**. These actions must be driven by the ergonomics and the HCI specialist. **The second method [23] allows to define flexible methods for model-driven user interface development** that uses usability goals as a starting point. This approach defines a model-based user interface development (UID) method according to the problem domain or context of the project by analyzing goals and activities. The proposed process derives User Interfaces (UIs) from business process. For this purpose, a set of UI models is used to interact

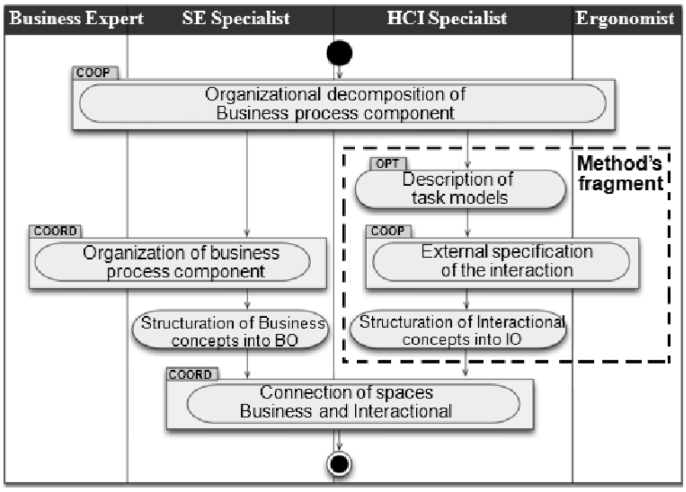

Figure 1: A phase of the Symphony method [10]

with each other according to the Cameleon Reference Framework [5]. The goal of this Framework is to build methods and environments supporting design and development of highly usable context-sensitive interactive software systems. This framework is composed of four development steps: create conceptual models (e.g. task model, data model, user model), create Abstract UI (AUI), create Concrete UI (CUI), and create Final UI (FUI). Figure 2 illustrates a global vision of this process. Concerning the HCI aspects carry out for the User Design Specialists, the activities proposed by this design method are: **Create Context of Use Model** to design user-centered UIs, Create AUI to specify objects in a UI independent from devices, **Transform into Task Model** to automate the generation of specification of UIs (receive AUI as input and generate task model), and finally **Transform into Domain Model** to automate the generation of UIs focused on the application domain for describe the manipulated data.

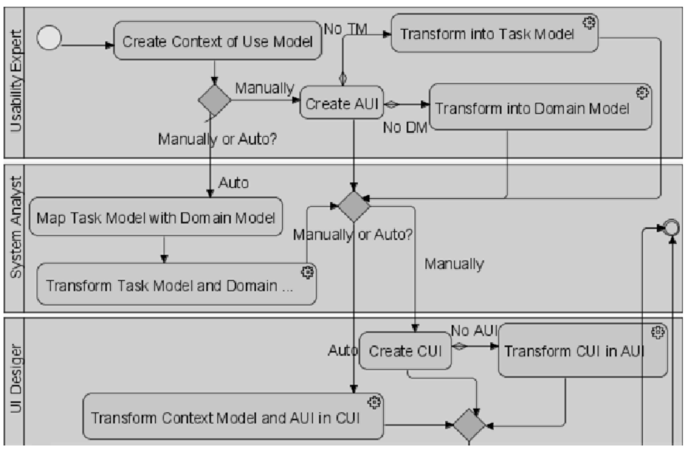

Figure 2: An interactive system modeling process [22]

General approach

Introduction

This section proposes theoretical concepts of our service-oriented approach for models management. In our approach, modeling services are defined in order to express modeling goals and to automate software development steps using models (e.g. model edition, model transformation, etc).

The basic service-oriented architecture

Service-Oriented Computing (SOC) [13] is a development approach that speeds the application development and facilitates the composition of distributed applications. A service is a set of self-contained software modules and auto-descriptive applications that can be described, published, discovered, composed, and negotiated on demand by a customer. Services perform functions, which can be anything from simple requests to complicated business processes [13]. To build integration-ready applications based on business processes, the service model relies on the service oriented architecture (SOA) [4, 13] (Figure 3). SOA is a logical way of designing a software system to provide services to either end-users or applications. This approach defines an interaction between software actors as an exchange of messages between service requesters (customers) and service providers. Customers are Model Designers or software actors that request the execution of a service. Providers are software agents that provide the service. Agents can be simultaneously service customers and providers. Providers are responsible for publishing a description of the service(s) they provide. Customers must be able to find the description(s) of the services they require and must be able to bind them.

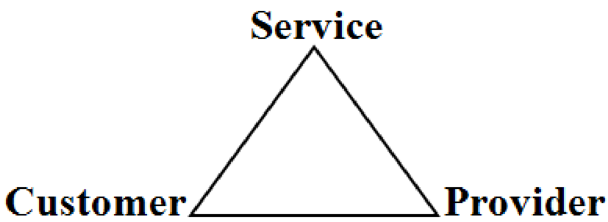

Figure 3: Service approach

Three levels of services

Our approach based on services relies on three modeling levels (see Figure 4) where providers, clients and services are different. The first level corresponds to the **operational layer**. This layer offers services for model designers, to facilitate the building of their modeling environment. The client is a model designer who wants to manage models in an individual or collaborative way (with other designers). So, he needs to define and to adjust his modeling environment to the functions he needs, in terms of models management. For example: an "HCI designer" can need a modeling environment that offers support for editing "task models" and transforming these models into a "concrete user interface" (CUI) for a specific device. **The organizational layer**enables the modeling of the design processes. This layer offers a support based on services for project groups. In this layer, the roles and activities are expressed in terms of simplified development processes. The objective is to capitalize fragments of methods in order to provide them to designers, who have a role in the project group. A method fragment is modeled by a set of activities that can be carrying out by different roles. The activities are described in terms of actions about models. The organizational layer uses the operational services in a coordinated way. Customers are, in this case, projects managers which need to define and manage roles and activities in their

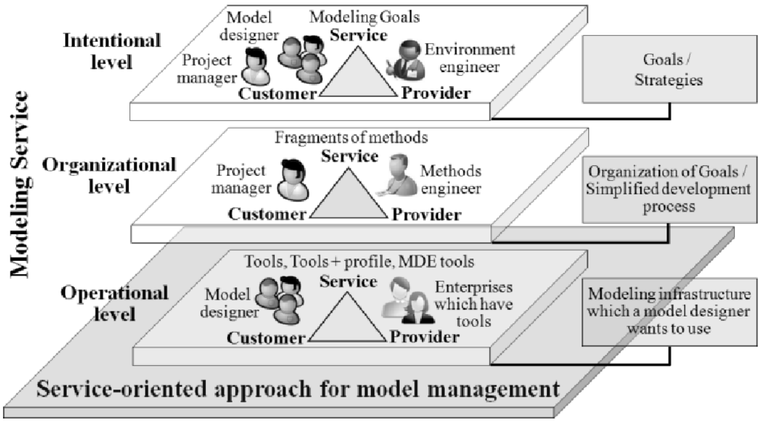

Figure 4: Three levels of service

development process. Project managers can choose some organizational services (part of design process) that require the implementation of operational services for model management. Thus, they create the model management environments for designers involved in their development process. **The intentional layer** (Figure 4) deals with modeling goals. It conceptualizes strategic modeling needs required by a specialist, a group of specialists, a unity of work or any organization involved in the development process. So, this layer uses the organizational services. The provider corresponds to the environment engineer who plays a new role in charge of the administration and management of the service platform. The customers are still those of the organizational and operational layers, e.g. the models designers and the project managers. For these customers, the services are the goals proposed by the environment engineer (e.g."Specify an Interactive System").

In this section we introduced the principles of our service-oriented model management approach. In the following sections, we detail the intentional and the organizational levels on the interactive system design methods presented in section 2.

Modeling an intentional service

An **intentional service** is a business-oriented service described from an intentional point of view (e.g. **specify** an interactive system, **study** the usability). It corresponds to the modeling goals. It can be composed by other intentional services. Finally an intentional service is realized by organizational services that correspond to methods fragments (Fig. 5). The service is characterized by **a verb** that is associated with objects and complemented by parameters. We identify a set of verbs which describe specific intentions for models management (e.g. specify an interactive system, study interactive system usability, **design** UIs considering users mental models to perform their task, **automate** the generation of UIs considering many devices). **The object** is a modeling concept that is defined or reused by the verb (e.g. Interactional Object, Abstract UI, Task, Interactive System). A parameter is a feature that plays specific roles concerning the verb, for example: in order to "specify the software architecture with the choice of the interaction devices" the parameter "with the choice of the interaction devices" corresponds to the way to solve the goal achieved. We use a linguistic approach to formulate an intention. Our purpose is to express the intentional services defined by the meta-model presented in Figure 5. This approach relies on the structural declaration of an intention proposed by Rolland [18]. We have adapted this general statement to the needs of model management. So, in our context work, the structure of an intention is:

$$\text{Verb} < \text{objS} > [p_1, ..., p_N],\ \ [\text{objT} >,\ [p_1, ..., p_N]]$$

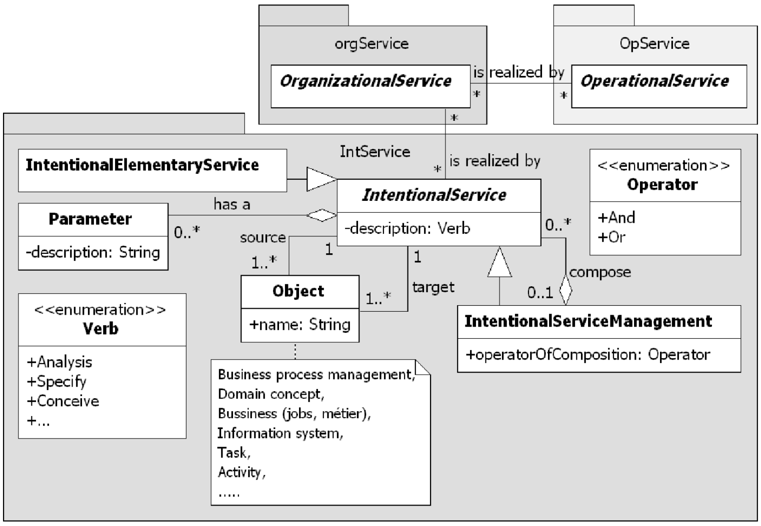

Figure 5: Intentional model service

The elements that are in hooks “[ ]” correspond to optional elements, the elements “$p_1, p2, , pN$” correspond to the terms: parameter1,, parameter$N$. ObjS and ObjT correspond to Object Source and Object Target respectively. The Object Source corresponds to the modeling concept reused by the intention. The Object target is the modeling concept that produces the intention. The general structure of the intentions corresponds to several cases. We present below some combinations followed by an example.

- **Intention 1**: Verb < ObjS >, For the intention: **“Specify an Interactive System”**, the general structure is: Verb(“Specify”) < ObjS(**“an Interactive System”)** >.

- **Intention 2**: Verb< Obj >,[p_1, p_2, p_N], For the intention: “Study the user interaction by task modeling”, the general structure is: Verb (“Study”) < ObjS(“$theuserinteraction''$) >, [p_1(“by task modeling$''$)].

Examples

At the intentional level, we must determine the goals of the two methods presented in section 2. These strategic goals are those required by Software Engineers (SE) and Human-Computer Interaction specialists (HCI specialists), who participate in the development process of interactive systems. Studying the two examples, we define the main goal specify an interactive system, which can be decomposed into other goals. So, based on the intentional model of the Figure 5, we create the appropriate intentional services to develop interactive systems. The representation of Figure 6 shows a composition of the goals required by the design process of an interactive system. In the figure above, rectangles correspond to the services and the links describe the composition of intentional services. For example: the service “Specify an Interactive System” is decomposed into three other services: “Specify functional aspects”, “Specify interactional aspects” and “Specify software architecture”. Similarly, the service “Specify interactional aspects” is decomposed intro three other services: “Study the usability”, “Specify the Interaction Process” and “Concretize the User Interface”. These services are linked to organizational services in order to propose a solution (in terms of processes) to the specified goals.

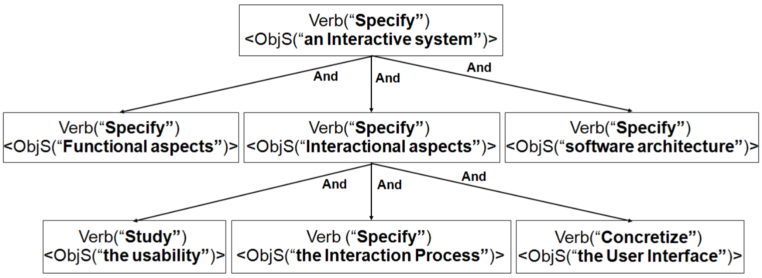

Figure 6: A composition of intentional services

Modeling an organizational service

This section presents the model of the organizational layer. Our organizational model service (Fig. 7) is inspired by the work of Ralyté et al. [16] who propose a method engineering process model approach, which allows to represent any method as an assembly of the reusable method fragments. In our work, we use the notion of service to support the construction of modeling processes by assembling method fragments. In our approach method fragments are considered like an action on the model that can be defined out by Method Engineers (ME). An organizational service consists in a composition of development method fragments that can be reused by model designers. A method fragment is represented by an organizational elementary service that is defined in terms of model manipulation. For example, in the definition of a transformation process to generate an Abstract UI, an aspect is the edition of the source meta-model. This activity consists in the production of a model that can be used by other method fragments. An organizational service is carried out by one or more roles. A model designer who plays a role of ME can define and reuse several organizational services. At this level, the collaboration term is used for coordination and cooperation tasks between designers. Another aspect considered by our organizational model is the fact that the operational services are the mechanisms for realize the organizational services. It means that organizational elementary services must use operational service to support the management of modeling activities [15].

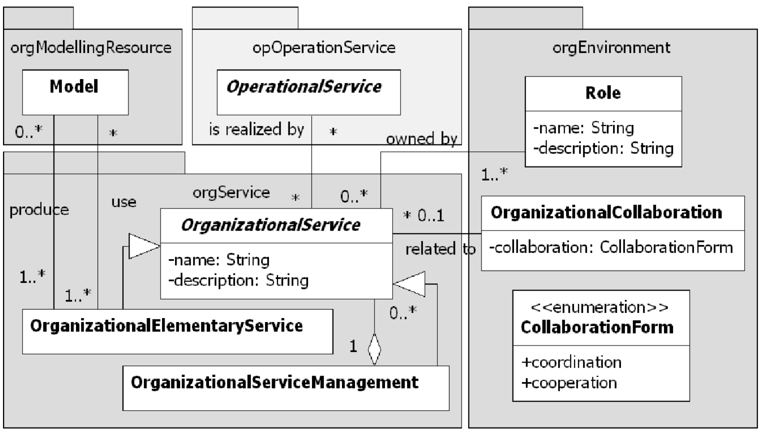

Figure 7: Organizational model service

Examples

As we have noted in previous section, we consider that an intentional service can be realized by several organizational services. Concerning the sequence of modeling activities of the intentional service "Study the usability" of an Interactive System, one of the possible processes that answers to this goal is a part of the phase "Organizational and Interactional Specifications of Requirements" as defined by the Symphony design method described in section 2. This method fragment (box of Fig. 1) must be defined in terms of organizational services. It corresponds to an organizational service composed of three sub-services (one by activity). So, the intentional level service: "Study the usability" is linked to this composite organizational service (Figure 8). The elementary services are described by actions that use or produce models. For example, the activity "description of task models" is expressed at the organizational level as an elementary service that carries out the action: "edition of task models". Moreover an organizational service can be supported for several tools defined in terms of operational services. The action "edition of task models" can be carried out by tools with support to task models among the well-knows are: CTTE [14], Teresa [1] KMAde[20].... Another possible process that answers to the goal "Study the usability" of an Interactive System is a part of the method engineering for model-driven user interface development proposed by [22]. This goal deals with the UI activities of the role "Usability Expert". They correspond to an organizational service composed of three elementary services: Create task model to describe task in a hierarchical manner, Create context of use model to describe the users characteristics, the platform used and environment; and Create domain model to describe the manipulated data. As previously, the action create task model needs operational services supporting task modeling. These operational services are those (i.e. services for CTTE, Teresa or KMade) which are already linked to the action "edition of task models". n this example, we have shown how an intentional service can be realized by organizational

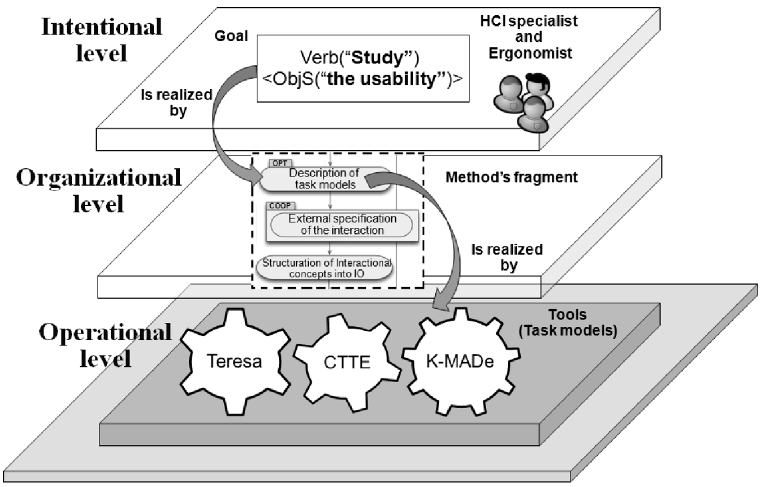

Figure 8: Relation between an intentional service and an organizational service

services. We also illustrate how an organizational service is composed of other organizational services, which can be linked to operational services. Through these two examples, we have showed the feasibility of our approach by illustrating the intentional services and their relationship with the organizational services. Thus, this approach allows to design model environments to facilitate the design of interactive systems.

Conclusion and perspective

This article presents the theoretical-concepts and principles used in our service-oriented models management approach designed to facilitate the work of model designers and project managers by helping them in choosing processes, models and modeling environments adapted to their specific needs. Our work is not yet limited just consider model management tools as services. It relies on three modeling levels: the operational layer to define the modeling environment for model designers; the organizational layer to enable the reuse of operational services in a coordinated way, but also the creation and the management of model process fragments; the intentional layer permits to define the modeling goals that can be implemented by design processes described at organizational level. We emphasized that an intentional service is defined by a verb associated with objects and parameters. Regarding those verbs, we notice that they are related by semantic relations. So, we consider that it is still necessary to create a corpus of knowledge relative to those verbs. This definition will make possible to write languages to implement taxonomies of goals for model management. On afterthought, we consider also necessary to conceive a modeling process language to express the organizational services defined by the meta-model presented in Figure 7. Our purpose is to unify the concepts of intentional and organizational levels (expressed in terms of models management) with operational services. Finally, we will consider realize a platform prototype for the implementation of our three levels of services. We intend to use the service-oriented tool DoCoSOC [25]. With this prototype, we will be able to easier evaluate the interest and the usability of our approach by conducing user experiments.

Acknowledgments

We are grateful to the Foundation "Gran Mariscal de Ayacucho", the LIG laboratory and the university "UCLA-Venezuela" for their financial support.

References

[1] Berti S., Correani F., Mori G., Paternó F., Santoro, C.: TERESA: A Transformation-Based Environment for Designing Multi-Device Interactive Applications, in Proc. of CHI04 (NY, 2004), ACM Press, 793-794.

[2] Beyer H.; Holtzblatt K.: Contextual Design. Defining Customer-Systems. (1998) San Francisco: Morgen Kaufmann.

[3] Bieber G., Carpenter J., Introduction to Service-Oriented Programming (Rev 2.1), Avril 2001.

[4] Blanc X., Gervais M.-P., Sriplakich P., Modeling Services and Web Services: Application of ModelBus, in Proc. of SERP05 (Las Vegas, 2005), 557-563.

[5] Calvary G., Coutaz J., Thevenin D., Limbourg Q., Bouillon L., Vanderdonckt J.: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers 15(3), (2003), 289308.

[6] Chalon R., David B-T.: IRVO: an Interaction Model for designing Collaborative Mixed Reality Systems, in Proc. of HCII05, (Las Vegas USA, 2005), 22-27.

[7] Constantine L., Biddle R., Noble J.: Usage-centered design and software engineering: Models for integration, in Proc. of the IFIP TC13 workshop on Closing the gaps: Software engineering and Human-Computer Interaction. (2003).

[8] Dubois E., Gauffre G., Bach C., Salembier P., Participatory Design Meets Mixed Reality Design Models - Implementation based on a Formal Instrumentation of an Informal Design Approach, in Proc. of CADUI06, (Bucharest Romania, 2006), 71-84.

[9] Dubois E., Gray P., Nigay L.: ASUR++: a Design Notation for Mobile Mixed Systems. Interacting With Computers 15(4) (2003) 497-520.

[10] Dupuy-Chessa S., Godet-Bar G., Pérez-Medina J-L., Rieu D., Juras D.: Technical and Functional Issues for Integrating Mixed Systems into Information Systems. Engineering of Mixed Reality Systems. Editors: Dubois E., Gray P., Nigay L. Springer, 2009. (To appear)

[11] Gulliksen J., Goransson B.: Usability design: Integrating user-centred systems design in the systems development process. In: Tutorial at CHI'05, (Portland USA, 2005).

[12] Hassine I., Rieu D., Bounaas F., Seghrouchni O., Symphony: a conceptual model based on business components, in SMC'02, IEEE International Conference on Systems, Man, and Cybernetics. Volume 2. (2002)

[13] Papazoglou M-P., Traverso P., Dustdar S., Leymann F., Service-Oriented Computing: A Research Roadmap, Dahstuhl Seminar 05462, 2005, 1-29.

[14] Paterno F.: ConcurTaskTrees: An Engineered Notation for Task Models. In: The Handbook of Task Analysis for Human-Computer Interaction, Lawrence Erlbaum Associates, 483-503, 2003.

[15] Perez-Medina J-L., Rieu D., Dupuy-Chessa S., A service-oriented approach for model management, in Proc. of 12th Iberoamerican Conference IDEAS09, (Medellin Colombia, 2009).

[16] Ralyté J., Rolland C., An Approach for Method Reengineering, in Proc. of ER01, (Yokohama Japan, 2001). Hunii H., Jajodia S., Solvberg A. (Eds.), LNCS 2224, Springer-Verlag, 471-484.

[17] Redish J., Wixon D.: Task analysis. The human-computer interaction, fundamentals, envolving technologies and emerging applications. L. Erlbaum Associates Inc., (NJ USA, 2003), 922-940.

[18] Rolland C.: Capturing System Intentionality with Maps, Conceptual Modeling in Information Systems Engineering, Springer-Verlag, Berlin Heidelberg, Germany, 2007, 141-158.

[19] Rosson M-B., Carroll J-M.: Usability Engineering: Scenario-based Development of Human-Computer Interaction. (2002) London: Academic Press.

[20] Scapin D-L., Lai-Chong Law E., Report and Refine Usability Evaluation Methods (R3UEMs), COST294-MAUSE 3rd International Workshop, (Athens, 2007).

[21] Sousa K., Furtado E.: From usability tasks to usable user interfaces, in Proc. of TAMODIA'05, (NY USA, 2005), ACM Press, 103-110.

[22] Sousa K., Mendona H.,Vanderdonckt J.: Towards Method Engineering of Model-Driven User Interface Development, in Proc. of TAMODIA07 (Toulouse France, 2007), 112-125.

[23] Tarby J-C., Barthet M-F.: The DIANE+ Method, in Proc. of CADUI96, (Namur, 1996), 95-120.

[24] Viala J., Dubois E., Gray P.: GUIDE-ME: Environement Graphique de Manipulation de la Notation ASUR, in ACM Proc. of Mobilite et Ubiquite 2004, (Nice - France, 2004), 74-78.

[25] Yu J., Lalanda P., Chollet S.: Development Tool for Service-Oriented Applications in Smart Homes, in Proc. of SCC08, (DC USA, 2008), 239-246.