UNIVERSIDAD DE CASTILLA-LA MANCHA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS

UN ENFOQUE ESTRUCTURADO PARA EL DESARROLLO DE INTERFACES DE USUARIO 3D

A STRUCTURED APPROACH TO THE DEVELOPMENT OF 3D USER INTERFACES

TESIS DOCTORAL

D. JOSÉ PASCUAL MOLINA MASSÓ

DIRIGIDA POR

DR. D. PASCUAL GONZÁLEZ LÓPEZ

ALBACETE, DICIEMBRE 2007

Table of contents

Abstract

Three-Dimensional User Interfaces, or 3DUIs, refer to those interfaces where interaction takes place in three-dimensional space. This area of Human-Computer Interaction is not a novel one, as these interfaces have been studied for several years now, though they have often been presented under other names, such as interactive 3D graphics, Virtual Reality or Augmented Reality. In fact, one of the icons of these interfaces, the head-mounted display, was invented short after another popular icon, the mouse, this being of desktop interfaces. However, the latter have now a state of maturity that three-dimensional interfaces have not reached yet. Desktop interfaces rely on a set of well-known interaction styles and widgets, and their development is based on well-funded methods. Quite the opposite, 3D user interfaces lack of a standard set of interaction techniques and controls, and their development relies on the experience and intuition of the developer.

Despite that fact, progress has been done in identifying the "universal" tasks that can be found in almost every 3D user interface, and together with it, many interaction techniques have been proposed to perform such tasks in different environments, some of them becoming quite popular such as ray casting or go-go. However, experience remains the key to successful development, and in this sense programmers apply their knowledge in software engineering practices, user interface designers rely on their skills in human-computer interaction, and artists use their know-how in content creation. The problem is that many proposed methods approach this development from the view of just one of these roles, then offering a partial solution to that development, which is not only about code, interaction or content but all these things together. Thus, it is necessary to combine both an engineering approach with a creative one.

That is one of the objectives addressed by the TRES-D framework, which is presented in this doctoral dissertation, offering a structured approach that is oriented to task and interaction as well as to objects and content. This way, the proposal promotes the participation of user interface designers and programmers as well as artists and digital content creators. In addition to them, it also involves the client, the domain expert and, of course, the user. The whole process is characterized by two main phases. The aim of the first one –the previous study- is to propose a solution to a given problem and, only if it is accepted, then the second phase follows –the detailed study- to complete the proposed design, and to implement and deploy it. This division in two phases reminds of former approaches to software development that were meant to reduce risks, and that is

exactly its purpose here, to reduce the risks that the development of experimental interfaces involves.

All in all, TRES-D is not simply a process model. A new digital-virtual-real continuum is also presented, which is meant to describe the design space of three-dimensional interfaces. Three meta-models, one for objects, one for interaction and a last one for space, offer developers a common language to better understand these interfaces, bring their ideas to paper, and communicate them to other developers. In order to help the designer when selecting the right interaction technique, two decision trees are given as guidance tools, one for object selection/manipulation tasks and another one for text input, both for immersive environments. To help the programmer when translating existing interfaces to three-dimensional environments, a new library is introduced, named VUIToolkit, which transforms plain widgets into 3D ones by adding true depth to them. Finally, the TRES-D proposal is demonstrated with four case studies, each showing a different application of 3D user interfaces.

1
Introduction

1.1 Summary

This chapter provides an overview of the field where this doctoral dissertation is rooted, progressively narrowing the attention to the particular problem on which it focuses. That field is devoted to the User Interface or UI, which plays an important role in current software development since it was accepted that, once the main functional requirements are addressed, the most important factor in the success of an application relies on the ease of use of the interface. This field is known as Human-Computer Interaction (HCI) or Computer-Human Interaction (CHI), and it drags many scientific events every year for the dissemination of research on theoretical and applied areas related to this field. As a result, there is a huge bibliography on user interfaces, which denotes its current state of maturity. This is true, at least, for desktop interfaces.

That state of maturity can be mistaken for a lack of new ideas, as current desktop interfaces are not much different from those running on first windows environments. This is not the case, as there is an ongoing research effort to find next generation user interfaces that offer richer interaction and match much better the capabilities of each single person. As people live in a three-dimensional world, using the three dimensions of space as a medium to interact with computers may be the answer to that quest.

Three-dimensional User Interfaces (3DUIs) have been studied for many years now. However, it may seem a novel area because they have not been refered as such until recently, but instead as 3D interactive graphics, Virtual Reality or Augmented Reality, and many other emergent technologies. Three-dimensional interfaces have not reached, in contrast, the same state of maturity that can be observed in desktop interfaces. The former are significantly more difficult to design, implement and use than the latter. One reason for it is that the design space in 3D user interfaces is also significantly more complex than in 2D, and there is still a lot of space to explore. Most part of previous research effort has been paid to "universal" tasks and interaction techniques, and also to human factors.

Despite the steps taken in 3D user interface research, it remains the question of how to combine the outcomes from all those previous works in a single framework that guides the designer thoughout the development of such interfaces, as this process has traditionally depended on each developer and their own experience. Several methodologies have already been proposed, though. Being most of them for particular applications of this kind of interfaces, the question is whether they are suitable for the development, in general, of three-dimensional interfaces and, if they are not, which process should be followed then. These questions are addressed in this doctoral dissertation, first presenting an in-depth review of existing design and development methods, and then proposing a new methodological framework, named TRES-D.

This document is structured in eight chapters. Chapter two is meant to define what a 3D user interface is, its elements, applications and design space, and presents a new digital-virtual-real continuum. In chapter three, twenty-six approaches to the development of such user interfaces are described and analyzed, and a first proposal called IDEAS-3D is given. Chapters four to six describe the TRES-D framework, which relies on three meta-models that provide a language for designers, and a process model that structures the development. Chapter six also presents two design tools for selecting interaction techniques, and one toolkit for implementing virtualized user interfaces –named VUIToolkit-. In chapter seven, four case studies are given, three of them illustrating the structured development process, and the last one applying VUIToolkit to the design and prototyping of distributed user interfaces. Conclusions, publications and further work are collected in chapter eight. Finally, an appendix has been added to describe in depth the implementation details of VUIToolkit, a set of 3D widgets based on the CUI model of UsiXML language.

2
The 3D user interface and its design

2.1 Summary

In this second chapter, a first study of 3D user interfaces has been carried out, addressing, in first place, their definition and the identification of the elements that they comprise, and then looking in depth at different aspects of the design of these interfaces.

It has not been an easy task, however, to find a definition of this kind of interfaces, as they are not usually presented as such but using other names, as for instance post-WIMP, non-WIMP or post-PC, names that make reference to interfaces that break with the previous WIMP interface style, which dominates the field of human-computer interaction at present. They are also found as Virtual Reality interfaces or virtual environments, or even in a more general form when talking about 3D graphics or simply “3-D”. All these terms are related to the use of 3D technology as an interface between the user and the computer, either as an input or as an output technology, or both, a characteristic that has then been used to propose a formal definition of what a 3D user interface is.

With the proposed definition, the focus then moved onto the applications of these interfaces, proposing a list of eight different uses of 3D user interfaces. This list represents a compilation of the different forms in which this kind of interfaces has been taken into practice. Some of then are more successful than others as a consequence of the experimental character of this technology. This should also be interpreted as the list is not closed, and other uses will surely be added in the future. The importance of this list is derived from the fact that past experiences should be taken into account by designers of 3D user interfaces in every new project, so that they can benefit from advantages found in those past experiences and do not incur in their same mistakes.

Another way to present 3D user interfaces to designers consists in placing them in the context of the evolution of the human-computer interaction, starting from the first interfaces based on a single instruction or command line, that is, from interfaces that exhibit just one dimension. Thus, based on the Reality-Virtuality (RV) continuum by Milgram and Kishino, a new design space is then proposed, extending itself not just in one axis but in two axes, one accounting for the number of dimensions and the other one for the degree of immersion, illustrating the gradual evolution of the user interfaces from 1D to 3D, and then from the virtual worlds to the reality, resulting in a new digital-virtual-real continuum (Figure 2.1).

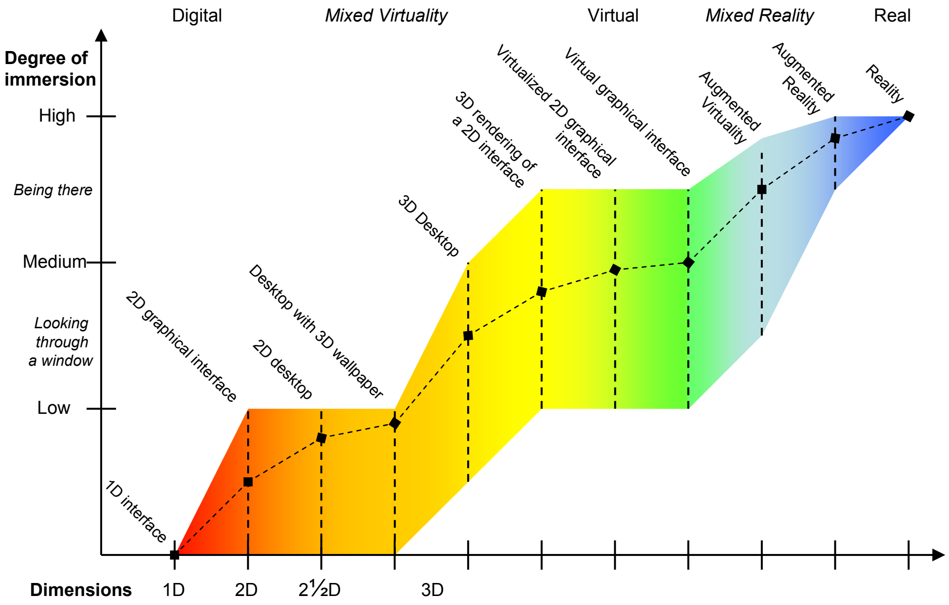


Figure 2.1 Digital-virtual-real continuum for user interfaces

Once the 3D user interface, its applications and its place in the human-computer interaction evolution have been defined, the next step taken in this chapter was the identification and description of the different elements of these interfaces. This way, four basic elements were addressed: the 3D space, the objects, the behaviours and the interaction. The latter was then studied in more depth. The study of all these elements found a huge number of terms and proposed taxonomies, but more important it showed significant differences among authors, which makes then necessary to put some order in all this mess, particularly in the model of objects and the model of interaction.

Following those previous studies, the design of this kind of interfaces was finally addressed. That section, first referred to the pillars that, according to Shneiderman, support the successful design of the user interface, in particular as derives from 2D interfaces experiences. However, all along that section it was shown that, despite the programming libraries, notations and tools that already exist, the design of 3D user interfaces is much more complex and has not reached that state of maturity yet, even worst, it is still in *incunabula* state.

All in all, even if the details of 3D user interfaces are known and the appropriate tools for their development are available, the design of such interfaces, as in every other artefact, is more difficult without a plan, without a method that guides the work of the designer, a key issue that will be addressed in the following chapter.

3 Methods for the development of 3D user interfaces

3.1 Summary

All along this chapter, it has been described and analyzed a total amount of twenty-six methodologies. They have been classified into seven different categories, as follows: one production pipeline for 3D animated movies, seven proposals for creating virtual worlds for PC, two methods that support a more participative development, four that make use of task analysis, three more that are based on software engineering, five methods for the development beyond the PC, and four methodologies for the development of conventional 2D user interfaces. The study has been completed with the description of a last methodology, IDEAS-3D, as the first proposal of the author of this Thesis.

In addition to that list of methodologies, it could be accounted one more, as at the beginning of the chapter it was given a first method based on three processes that are executed in sequence –requirements, design and implementation-, following the guidelines given by Shneiderman. Leaving aside that plain guidelines, most of the proposals collected in this chapter are methods for the creation of virtual worlds or Augmented Reality applications. That is, rather than methodologies for the development, in a general way, of 3D user interfaces, the authors of these proposals bring their attention to particular applications of this kind of interfaces. Besides, it is common to approach the development from the point of view of a particular discipline, resulting in different coverage of each role's activities, which is summarized in Table 3.1. This is also reflected in the kind of tools that are found in each proposal, as Table 3.2 shows.

	UI designer	Sw engineer	Programmer	Artist	Content creator
3D animated film production					
Pixar production pipeline, according to (Reddy, 2005)	.	.	/	X	X
Creating virtual worlds for PC					
Current VR application development, according to (Kim, 1998)	.	.	/	.	/
Current VE development, according to (Smith, 2001)	.	.	/	.	/
EON Studio workflow (EON Reality Inc.)	.	.	.	.	/
Current approaches to VE design, according to (Kaur, 1998)	.	.	/	.	X
VE design process (Kaur, 1998)	X	.	.	.	.
Creating an interactive 3D product using VRML (Hay, URL)	X	.	/	/	X
Building X3D worlds (Daly, 2002)	X	.	.	/	X
Participative development of virtual worlds					
VE user-centred development process (Neale, 2001)	X	.	.	.	.
3D content-centred methodology (Celentano, 2001)	X	.	.	.	X
Task analysis in VE development					
Sequential evaluation of VEs (Gabbard, 1999)	X	/	.	.	.
VE development (Pereira, 2000), (García, 2001)	X	/	.	.	.
VEDS (Eastgate, 2001), (Wilson, 2002), (D'Cruz, 2003)	X	/	X	/	X
Creating virtual worlds using X3D (Polys, 2005)	X	/	/	.	/
Development based on software engineering					
A prototype design methodology for VEs (Fencott, 1999)	X	X	/	X	.
CLEVR (Seo, 2001)	X	X	X	.	.
SENDA (Méndez, 2001), (Sánchez, 2001, 2003, 2005)	X	X	X	.	/
Beyond PC worlds					
VE design process –INQUISITIVE project- (Smith, 2001)	X	/	X	.	.
Interaction/content-driven design (Parés, 2001)	X	.	X	.	.
VRID (Tanriverdi, 2001)	X	/	.	.	.
MPIUA (Granollers, 2002)	X	/	.	.	.
VR design process (Sutcliffe, 2003)	X	X	.	/	.
Conventional user interface development					
Design methodology (Foley, 1996)	X	/	.	.	.
LUCID (Cognetics Corp.)	X	.	.	.	.
OVID (Bardon, 2001)	X	X	.	.	.
IDEAS (Lozano, 2001)	X	X	.	.	.

Table 3.1 Coverage of roles' activities by each approach: . - poor, / - partial, X - broad

Pixar production pipeline (Reddy, 2005)	Storyboards, art, sketches, clay, 3D modelling sw.
Current VR app. dev. (Kim, 1998)	CAD tools, programming libraries
Current VE dev. (Smith, 2001)	3D modelling sw., VE development sw.
EON Studio workflow (EON Reality Inc.)	3D authoring sw.
Current VE design (Kaur, 1998)	Reference material, VE development sw.
VE design process (Kaur, 1998)	Design advice hypertext tool
Creating an interactive … (Hay, URL)	Proj. documentation, storyboard, 3D authoring sw.
Building X3D worlds (Daly, 2002)	Sketches, storyboards, 3D modelling sw.
VE user-centred dev. (Neale, 2001)	Brainstorming, storyboards, reports
3D content-centred methodology (Celentano, 2001)	Classes of 3D experiences, customized 3D authoring sw.
Sequential eval. of VEs (Gabbard, 1999)	User task scenarios
VE dev. (Pereira, 2000), (García, 2001)	Dataflow diagrams, storyboards
VEDS (Eastgate, 2001), (Wilson, 2002), (D'Cruz, 2003)	Interviews, focus groups, scenarios, ethnographic techniques, storyboards, task lists, ref. material, taxonomies, guidance tools, VE development sw.
Creating virt. worlds using X3D (Polys, 05)	Scenarios, Task-Knowledge Structures (TKS)
A prototype design methodology for VEs (Fencott, 1999)	Art, storyboards, plans, use cases and class diagrams, Perceptual Maps
CLEVR (Seo, 2001)	Message sequence and dataflow diagrams, CSL language, statecharts, Visual Object Specification (VOS), sw. components, VE development sw.
SENDA (Méndez, 2001), (Sánchez, 2001, 2003, 2005)	Use cases, use concepts, class and transition diagrams, event traces, questionnaires, view maps, a set of forms
VE design process –INQUISITIVE project- (Smith, 2001)	Storyboards, annotated scene graph, flownets, VE development sw., VE toolkits
Interaction/content-driven design (Parés, 2001)	Taxonomies, 3D modelling sw., VE development sw.
VRID (Tanriverdi, 2001)	Natural lang. desc., taxonomies, PMIW: dataflow and state transition diagrams, VR development sw.
MPIUA (Granollers, 2002)	Ethnographic techniques, focus groups, Hierarchical Task Analysis (HTA), Puntos de Observación (PO), State Transition Notation(STN)
VR design process (Sutcliffe, 2003)	HTA, TKS, use cases, scenarios, questionnaires, reference material, sketches, taxonomies
Design methodology (Foley, 1996)	State diagrams, storyboards, style guides, UI toolkits
LUCID (Cognetics Corp.)	Envision sta., sketches, personas, scenarios, key-screen prototype, style guide, UI development sw.
OVID (Bardon, 2001)	Stories, UML diagrams, CASE tool, sketches, UI development sw.
IDEAS (Lozano, 2001)	Use cases, task templates, sequence and class diagrams, dialogue and component specification diagrams, UIDL, CASE tool

Table 3.2 Representative tools found in each approach

The first methodology to be described and analysed, however, did not aim at the development of any user interface, but a 3D animated movie instead. In spite of that, it should be remarked the order followed to bring scenes from paper to computer, the tools used, and the continuous revision of reels showing progress made in each scene.

The inclusion of the previous methodology is not arbitrary, as the tools and techniques used in 3D animation are fairly similar to those employed in the creation of virtual worlds, if not the same, as it was then confirmed in the second group of reviewed methodologies. The main difference is the interactive nature of virtual worlds, which introduces the need for reaching the proper performance and avoiding usability problems. The first issue is tackled by an optimization process. In this process, the level of detail of the virtual world is adjusted, based on the performance of the graphics hardware of the target platform, a reference that some authors point out that should be elicited as a requirement in a plan or design stage prior to construction. The second issue is addressed by following some guidelines at the design stage, such as the ones proposed in (Kaur, 1998).

The methodologies of the previous group seem to assume that just one developer carries out the whole process. There is only one proposal that considers more than one, that is, a team of developers. Even more, they also seem to assume that the developer knows well the content of the virtual world and the preferences of the future user, though it is more probable that they are not familiar with that content, and that they will probably incur in a great mistake thinking that the user has the same preferences as they do. That led us to the third group of methodologies, characterized by promoting a higher involvement of both the user and the domain expert in the development. Thus, in order to obtain a product that satisfies the user, these methods proposed that the user should participate at the beginning of the development, during the course of it, and at the end of it. In addition, in order to obtain a product with the right content, the domain expert should also participate at the beginning and the course of the development, assisting, supervising or creating as any other author.

Up to this moment, the reviewed methods were oriented to the content of the virtual world, assuming that the main task of the user in that world is nothing but exploring it and finding how the objects react to their actions, without carrying out any specific task beyond that. On the contrary, there are other environments where the user does have to perform certain tasks and, therefore, the development must include the analysis of them. In that context, the fourth group of proposals takes into account this need, and from the study of such proposals there is one important thing to remark, which is the difference between the real world tasks and the virtual world tasks. This means that not one, but two are the analysis processes that must be carried out in the same development. However, there are still some doubts regarding the way tasks are related to content, an issue that in some methodologies, such as VEDS, seems not to be a problem as the tasks are nothing but a simple list of actions on objects, as it can be read in (Eastgate, 2001).

In such cases where the list of possible actions is short, only few objects react to them and the behaviours that objects exhibit are limited, seems to leave the programming task, and the software engineering with it, in a second place. The fifth group of methodologies demonstrated, quite the opposite, that techniques and tools coming from software engineering can also be applied to the structural aspects of the virtual environment, and more obviously to the components of such environment that have to be programmed, having a starring role when the development relies on a set of software libraries and not just on an integrated authoring environment. Nevertheless, some aspects of the virtual environment require the introduction of new diagrams, such as the concepts of use proposed in SENDA, and there are others that are just out of the scope of such practices, as the aesthetics of the environment according to (Fencott, 1999).

Even with their differences, the methodologies of the previous groups shared a common characteristic, that is, all of them assume that the user will face the computer through simple techniques based on the standard keyboard and mouse of a PC. The use of non-conventional devices implies different interaction techniques that must be carefully detailed at design, but also software to support such devices. The methodologies of the sixth group were then oriented, some of them, to the development of virtual environments ranging from desktop to immersive setups, and the rest of them, to the development of Augmented Reality interfaces. The result is a higher attention to the interface, in some cases focusing on theory aspects of it, as a physical and logical medium between the user and the application objects, and in other cases offering more practical solutions, as for instance the flownets proposed in the INQUISITIVE project.

The last group of methodologies was also oriented to the development of the user interface, but this time a flat interface, not a multimedia interface that could be related to the creation of virtual worlds, but interfaces for business applications. Attention, this way, is paid to analyse user tasks, model the user, design the navigation paths that lead from each screen or window to others, and specify the components that are part of each of them. Besides, as these methods rely in well-known components, they also have the advantage of tools that allow a rapid prototyping of the interface, as for instance in LUCID, and even the compilation of the interface from models that describe the application and its interaction in an abstract way, as in IDEAS.

In this sense, and with the aim of bringing the advantages of these last methodologies to development of 3D user interfaces too, the IDEAS-3D methodology was then proposed, bridging the gap between 2D and 3D as it is described in the digital-virtual-real continuum introduced in the previous chapter. However, that experience mainly showed the difficulty that represents to extend a method such as IDEAS to cover the whole design space of 3D user interfaces. Doing so, and according to the lessons learned in this chapter, requires greater changes in the process model, which are addressed in the next chapter by introducing a new methodology.

4
Towards the TRES-D methodology

4.1 Summary

As a response to the problems identified in the creation of the 3D user interfaces, which has been pointed out in the previous chapters, in this Thesis it is proposed the development of a new methodology. Here, a name has been given to it, TRES-D, and along with it the scope of that methodology has been established, listing among other objectives not only to give an ordered set of steps for the development of these interfaces, but also to provide a number of models that allow an easy understanding of such interfaces, and also a framework where design and construction practices and tools can find their right place.

From now on, this TRES-D methodology will be developed, having as a distinctive characteristic to put together two different approaches for the development of user interfaces, one oriented to tasks and the other one to content, in one single structured process. However, the details of this process will not be given yet. Before that, three models will be introduced, one for the objects, one for the elements of interaction, and the last one for the space. With this models or, better said, "meta-models", it is the intention to build a language that allows an easy communication among developers of such interfaces, using a vocabulary that they find familiar as it is based on literature from the field, but with the necessary changes in terms and meanings to solve ambiguities and contradictions found in such literature, and establishing a set of relationships that help developers in understanding the concepts and their use within the TRES-D methodology.

5
Meta-models for a new methodology

5.1 Summary

In this chapter, three models have been described, introduced with the purpose of clearing ambiguities and contradictions in the vocabulary that is used by 3D user interfaces researchers and developers, refining the meaning of some terms or introducing new others when needed, and confirming or redefining the relationships between some concepts and others. The aim of these models is to help developers building their own mental model of such interfaces, modelling the different elements that comprise them, and communicating those models to other people. In that sense, rather than models, what this chapter has described are “meta-models”.

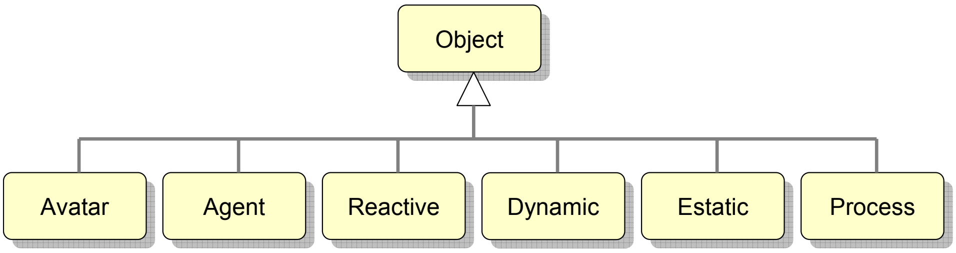


Figure 5.1 Proposed object classification

Starting with the object meta-model, a new classification of objects (Figure 5.1) was introduced with it, putting together in a simple hierarchy many different classes identified by other authors, distinguishing between process, static object, dynamic object, reactive object, agent and avatar. The purpose was to help developers in their task of working out the design requirements for each object, but once they are classified the developer will also need to know what to specify and how. In that sense, and after a new revision of the different models proposed by previous authors, for the TRES-D methodology it was decided a composition of function, behaviour and a set of sub-models (Figure 5.2). Among these sub-models, the graphics one was given as an example but remarked as an important one, identifying in it four new elements: structure, geometry, appearance, and perception.

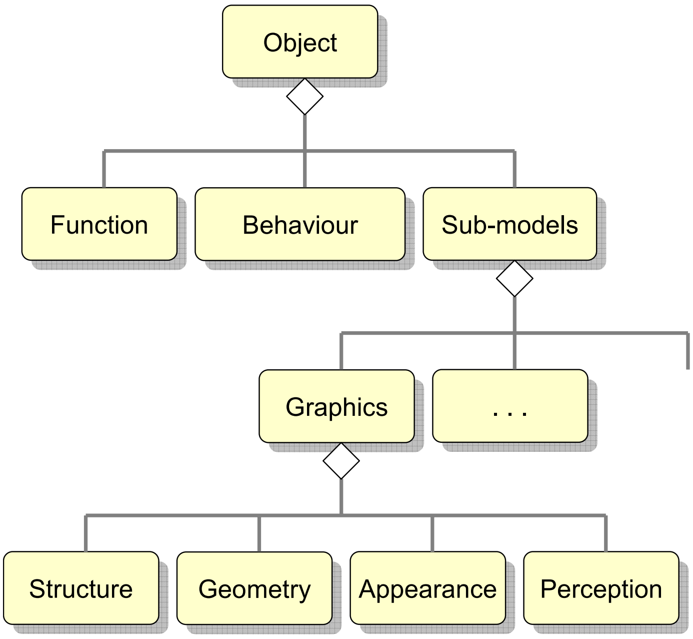


Figure 5.2 Proposed object model

The object meta-model was then followed by the one for the interaction elements, based in part on the approach given by Foley *et al.*, but making a great effort to rework some of their model to clarify some of the terms that are widely used in the literature, but authors do not agree in their meaning. The concepts that are included in this model are: dialogue, tasks, operations, interaction tasks, interaction techniques, actions, controls and physical devices. From them, it should be remarked the distinction, when talking about tasks, between high-level tasks and sub-tasks, and within the latter the lowest level ones are the operations, followed by the information units they require. The leaves then represent what is called Basic Interaction Tasks or BITas, and when going upwards in the hierarchy then it is found what is called Composite Interaction Tasks or CITas. It is also remarkable the distinction between interaction techniques and controls, the latter being part of the former, so that a widget, as a control, it is not considered as an interaction technique here, but only a part of it (Figure 5.3). Besides, the concept of widget 3D was clarified and, at the same time, a new concept was introduced, named as obget. And, establishing a parallelism with the interaction tasks, it is also proposed the terms Basic Interaction Techniques or BITes, and Composite Interaction Techniques or CITes. With these acronyms, the intention is to avoid the usage of IT letters alone, so much used in the literature, but so many times leading to confusion as they can refer to both interaction tasks or techniques if not specified.

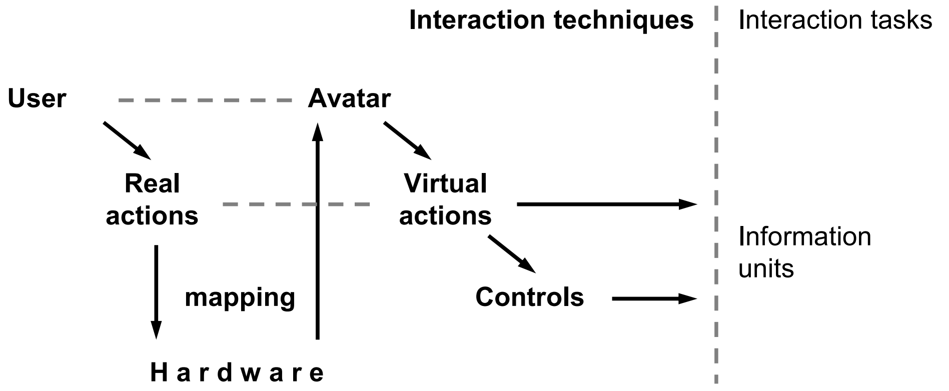


Figure 5.3 Interaction techniques: real and virtual actions, controls, and relationship with tasks

The third and last meta-model to be developed was the space one that, based on the digital-virtual-real continuum presented in the second chapter, distinguished between the 2D digital space, the 3D virtual space, and the 3D real space. In the last two spaces, it was highlighted the concepts of zone, place or locus, as well as the point of view or point of observation, as important elements in the ordering of such spaces. In all three it was remarked the relationships that link them together, as embedding a virtual world in a 2D interface through a viewport, immersing a 2D interface in a 3D environment, and connecting all them with the real world by means of physical devices that act as interaction surfaces and volumes (Figure 5.4). The model was finally completed with the definition of four types of navigation: inter-window, intra-window, inter-worlds –through portals- and trans-spatial.

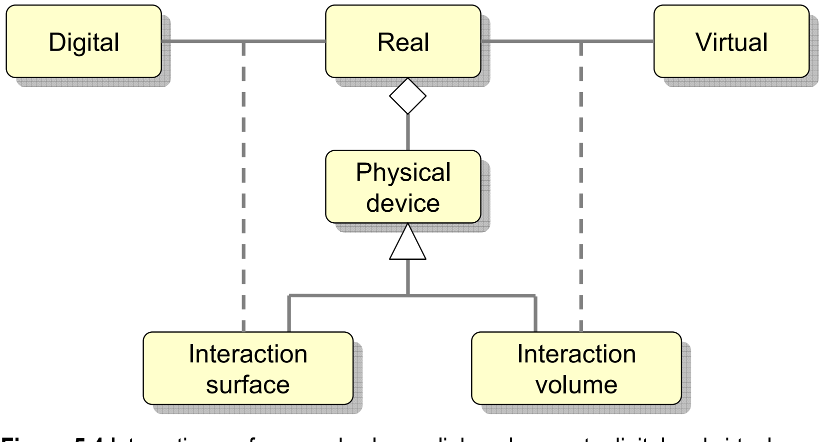


Figure 5.4 Interaction surfaces and volumen link real space to digital and virtual ones

All in all, and as it may be expected, the three meta-models are also related with each other. Thus, the tasks are decomposed in particular actions that are performed on some objects, and both tasks and objects are tightly related to space, something not surprising in this kind of user interfaces. This triangle relationship will strongly define the process model of the TRES-D methodology, as it will be shown in the next chapter.

6 The process model of the TRES-D methodology

6.1 Summary

The process model of the TRES-D methodology is supported on the concepts and the terms provided by the three meta-models presented in the previous chapter. This process model gives shape to a framework where different practices and tools for the general development of 3D user interfaces are gathered, achieving with all this the objectives that were established for this methodology.

The principles that characterize this process model are the following: first, to reduce as much as possible the risks that the developer may face in a field still far from being mature; second, to distinguish between design and implementation, and within the former to differentiate between an implementation-independent design and a dependent one; third, to orient the development to both tasks and interaction as well as objects and content, that is, to both user interface designers and programmers as well as artists and digital content creators; last, to involve not only these cited professionals, but also the client, domain experts and, of course, the user.

The process model is then divided in two main phases (Figure 6.1). The first one is devoted to understand the problem and then propose a solution, and the second one is aimed at developing such solution up to its final deployment and further maintenance. Each phase is also composed of three stages. Thus, the first phase, or initial study, consists of the problem introduction, the problem analysis, and the solution proposal. The second phase, or detailed study, is composed of design –design I and design II-, implementation, and deployment and maintenance.

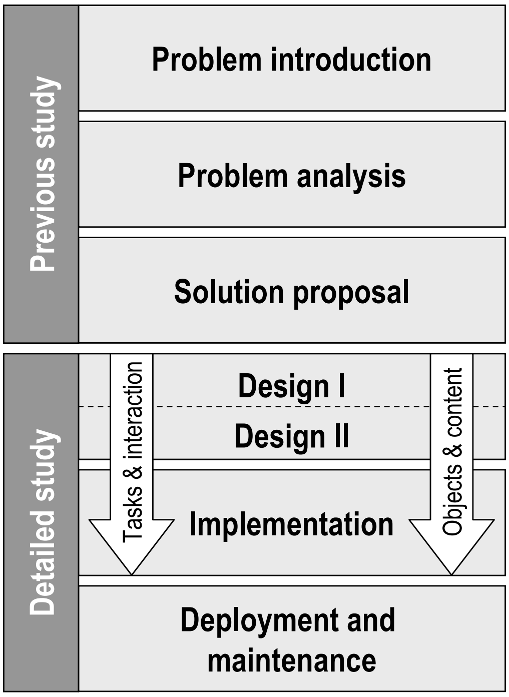


Figure 6.1 TRES-D process model

In each stage, it is possible to prepare in advance some work of subsequent stages, so that the developer can foresee the problems that may need to be faced. In fact, the stage of solution proposal is meant to tackle in advance part of the work that would belong to the phase of detailed study. An agreement with the client mediates between both phases, as it is the client who has to approve the proposed solution. Once at the second phase, the design is addressed at two different levels, one detached from the implementation details –design I-, and another one tightly related to that implementation –design II-. A decision on which software and hardware will support the solution divides both design levels. In any case, both design I and II, and later the implementation stage, are divided in two parallel activity chains. One of them is oriented to the interaction and the other one to the objects, following the proposed meta-models, translating the weight towards one or the other depending on each particular development, but in any case crossing both lines at several points of the development, specially due to their relationship with the third model, the space model. The last stage, deployment and maintenance, can not be forgotten in the development of this kind of interfaces, which is quite often characterized by their non-conventional hardware.

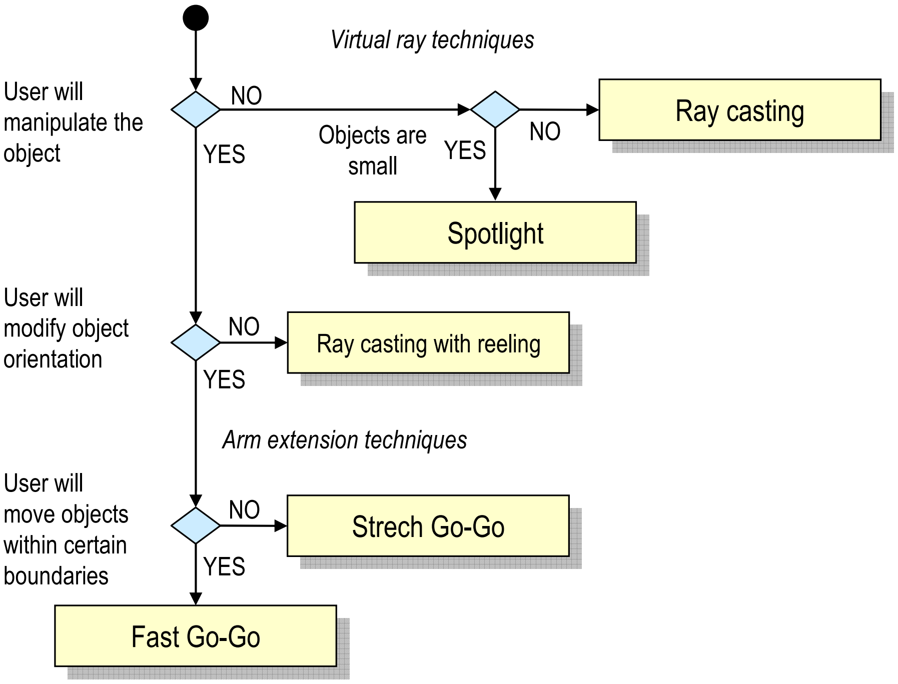


Figure 6.2 Design guidance tool for object selection and manipulation techniques

As a framework, apart from dividing the process in phases and stages, the model also recommends different tools for each activity that takes place along the development process. Some of these tools serve the developers as a guide in their decisions, others help them to document their ideas, and finally there are others that make possible to turn that designs into real products. Regarding the first ones, two more have been added to those that are already known, in particular these two tools are two decision trees meant for selecting interaction techniques, one for selecting and manipulating objects (Figure 6.2), and the other one for text input, both in the context of immersive virtual environments. As regards the second kind of tools, it has been referred to existing ones, some of them as for instance the UML diagrams are well-known by software engineers, others such as drafts, storyboards and layouts are part of the daily practice of 3D model creators and animators. In spite of that, and taking into account the lack of standard sets of controls for 3D user interfaces, it has been introduced the VUIToolkit library as a set of pre-defined 3D widgets (Figure 6.3), and the details of its implementation have been given, highlighting the relationship between the elements of its architecture and the meta-models for objects and interaction elements.

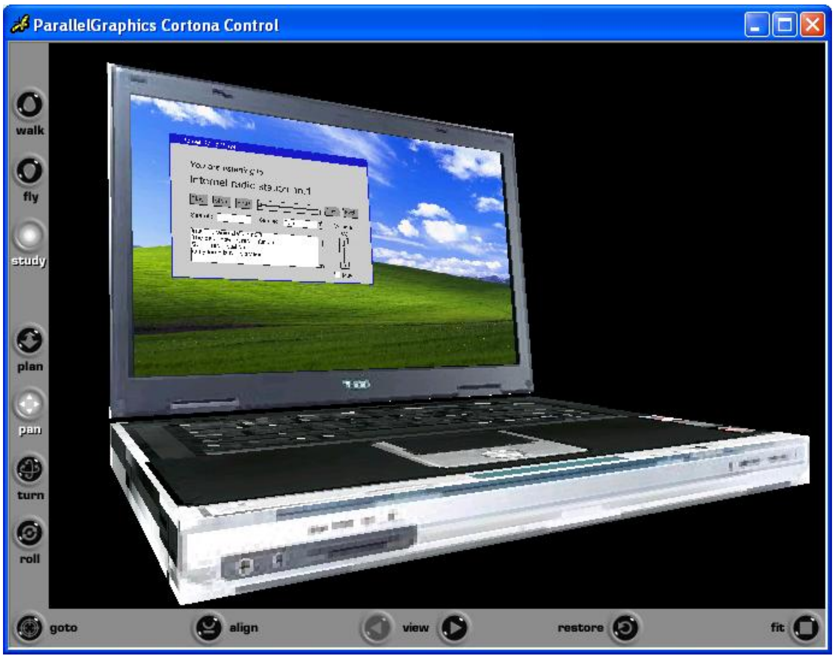


Figure 6.3 Virtual laptop showing VUIToolkit's 3D widgets on its screen

All in all, the main purpose when developing the TRES-D methodology was to put together new solutions with the best of the studied proposals in an iterative and incremental creation process that could be adapted to the variable complexity found in the different developments of 3D user interfaces, process that will be illustrated with practical cases in the next chapter.

7
Case studies

7.1 Summary

This chapter describes several case studies that bring to practice the proposals introduced in this Thesis. In total, four case studies are detailed, three of them showing the structured development of 3D user interfaces that the TRES-D methodology proposes, and the forth one showing the use of the VUIToolkit library, which is one of the contributions that has been presented together with the mentioned methodology.

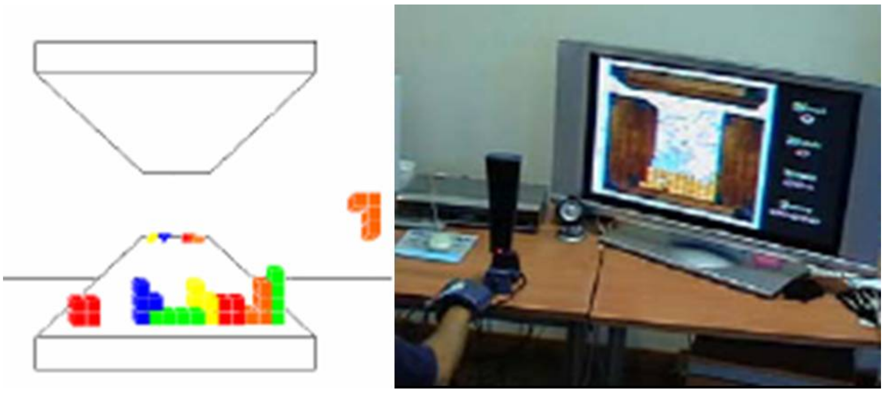


Figure 7.1 TTristero first sketch (left) and final user interface (right)

Starting with the first three ones, these case studies had not only in common the development of a 3D user interface with the TRES-D methodology, but also the use of datagloves as input interface. However, each project was devoted to a different application, each of them with its own particular requirements, which led to the selection of different data glove models. One of these projects had the objective of creating a videogame for PC desktop that allowed the user to get into the game using 3D stereo graphics and a dataglove, and so the interest was to find a low cost solution (Figure 7.1). The second of these projects pursued the virtualization of a block-based construction game in an immersive Virtual Reality system, looking this time for the highest degree of naturalness (Figure 7.2). Finally, the third one was aimed at reproducing the user interface shown in “Minority report”, the movie, using all available means for this purpose (Figure 7.3).

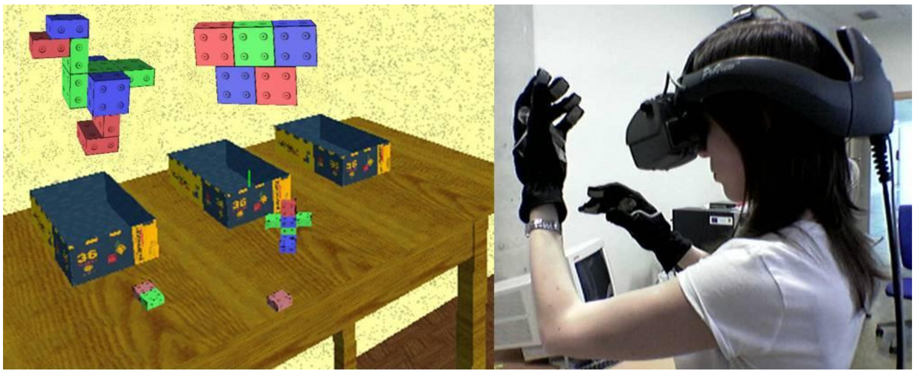


Figure 7.2 VRPrismaker virtual playground (left) and user (right)

The three projects were tackled in two phases, following the TRES-D methodology, starting with a preliminary study that results in a proposal where specific requirements for the 3D user interface are included –highlighting this time the selection of the dataglove model-, then followed by a detailed study where the design is completed and the implementation is carried out –highlighting in this case the separation between interaction and content, which is characteristic of the TRES-D methodology-. The three proposed meta-models –object, interaction elements and space- are present throughout the development of the these projects, as well as the use of tools and guidelines that were gathered in the methodological proposal for the development of this kind of interfaces.

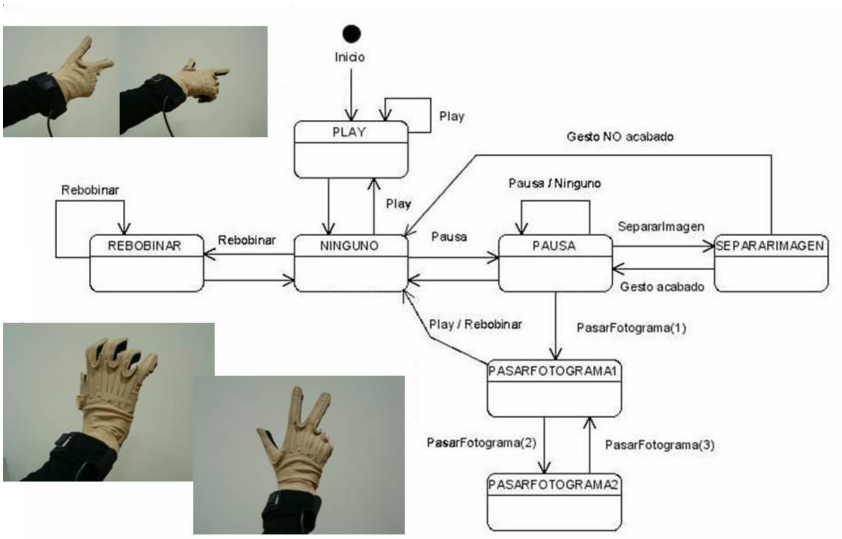


Figure 7.3 State diagram and gestures associated to different operation

The forth and last case study was not centred in the process model as the former three but, as it has been said before, it shows the usage of the VUIToolkit library. This toolkit, whose main characteristic is to transform the widgets from plain interfaces into a true 3D representation, that is, into 3D widgets, is presented in this case study as a tool for designing and prototyping distributed user interfaces –DUIs-. Thus, starting from a UsiXML specification that describes the user interface based on the four models of the Cameleon framework, together with the description of the context of use that includes the physical environment as a VRML97/X3D scene, then a virtual simulation of such environment is generated and can be viewed with a Web3D browser. That simulation also renders the platforms and the user interfaces of the applications that are executed in these plataforms, using the VUIToolkit to represent such interfaces and allow their manipulation and distribution across different platforms, all with prototyping purposes (Figure 7.4).

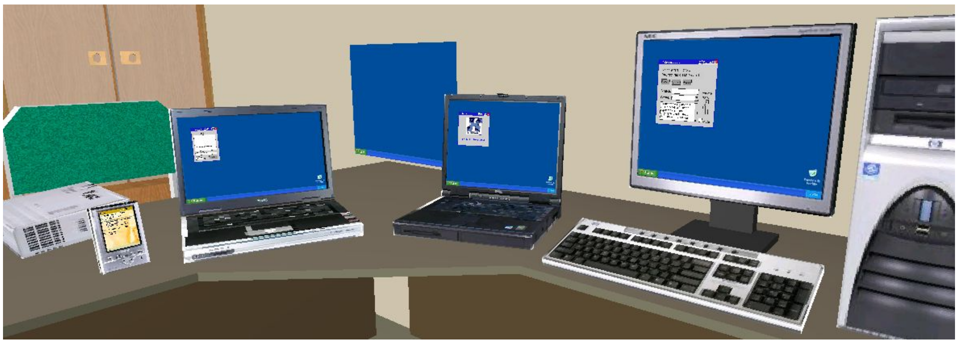


Figure 7.4 Overview of the virtual environment, showing platforms and user interfaces

With the description of these four case studies, this doctoral work reaches its end, whose main contributions are then summed up in the next and last chapter, highlighting the conclusions that are derived from this research, and also the future work that can then start from it.

8
Conclusions

8.1 Summary

Three-dimensional user interfaces have not reached the same state of maturity that can be observed in desktop interfaces, where well-founded methods and practices allow a rapid development, highly automated. Starting from past experiences in conventional user interfaces, this doctoral dissertation then focused on existing approaches to the design and creation of three-dimensional interfaces. Many proposals were found, and many more may have been ignored but, beyond the number, attention must be paid to the different points of view from where each author approaches to the given problem. Following a careful review of all of them, it was concluded that a new proposal was needed. That proposal was named TRES-D, which stands for "ThRee dimEnsional uSer interface Development". However, this work is not simply a sequence of phases, stages and activities.

The following list sums up the contributions of this doctoral dissertation. Many of them have been presented in national and international scientific events, being published in their proceedings. It is worth mentioning the participation, every year since 2003, in Interacción, an international conference organized by AIPO, and the participation in other events such as Web3D, IUI and VRST, which are organized by ACM. All these publications are also cited in the next list, matching them with their related contribution:

- ✓ Many terms and definitions are related to 3D, so the first step to take was to define what a 3D user interface is. Together with the proposed definition, a list of past and current applications was given, some being successful, others not.

- ✓ The well-known reality-virtuality continuum by Milgram and Kishino has been transformed into a new digital-virtual-real continuum. A first version of this design space is found in [Molina, 2005a], introduced to explain the virtualization of user interfaces. This new continuum makes room for other applications of 3D user interfaces, such as 3D desktops.

- ✓ The different elements that compose a 3D user interface have been identified and described, being these the objects, their behaviour and interaction, and the space. Later, this work served as a base for the definition of the three meta-models included in the TRES-D proposal.

- ✓ The problems that the design of each element, and the whole interface, encompasses have been studied. On the one hand, interaction does not follow the turn-taking dialogue typically found in conventional interfaces. On the othe hand, there is neither standard set of interaction techniques nor controls. These findings encouraged the proposal of design tools and toolkits in the TRES-D framework.

- ✓ Twenty-six existing methodologies have been described and analyzed, grouping them in seven different categories according to some similarities found in these approaches. Most of them are aimed at the development of 3D user interface applications, such as virtual worlds. This large review is summarized in [Molina, 2005b].

- ✓ Based on the experience gained through applying the IDEAS methodology to the development of conventional interfaces, a first proposal, previous to TRES-D, was presented as an extension to that methodology, named IDEAS-3D. This is presented in [Molina, 2003a], [Molina, 2003b] and [Molina, 2003c].

- ✓ Leaving aside IDEAS-3D, and heading to the TRES-D framework, three new meta-models were then introduced to better understand 3D user interfaces and facilitate their design and implementation. One meta-model tackles objects and their classification, a second one defines interaction elements and their relationships, and the last one is devoted to space. From them, the interaction meta-model is described in detail in [Molina, 2006b].

- ✓ The TRES-D process is found in [Molina, 2005b], [Molina, 2006b], and [Molina, 2006d]. This process is characterized, first, by dividing the development into two main phases. At the end of the first phase –the previous study-, a proposal is presented, and only if it is accepted, then the second phase follows –the detailed study-. This second phase includes design and implementation, where user interface designers and programmers tackle tasks and interaction and, in parallel, artists and digital content creators focus on objects and content. Design is also divided in two horizontal abstraction levels, not only to differentiate between interaction tasks and techniques as in other approaches, but also between artists' impressions and object specs given to content creators.

- ✓ As part of the framework, two guidance tools have been presented, being these two decision trees that help designers selecting interaction techniques. One of them guides towards the appropriate object selection/manipulation technique in immersive environments, and it is described in [García, 2005a] and [García, 2005b]. The other one guides towards the right text input technique in HMD-based environments, and it is described in [González, 2007].

- ✓ In addition to design time tools, a toolkit for programmers has also been included, to help them translating flat interfaces to three-dimensional environments. Its name is VUIToolkit, a set of 3D widgets meant for virtualized user interfaces, which was first introduced in [Molina, 2005a].

- ✓ Finally, the proposed framework has been brought to practice through four case studies. Three of them demonstrated the process of the TRES-D methodology and has been detailed in [Molina, 2006b] and [Molina, 2006d]. The forth one showed the application of VUIToolkit in the design and prototyping of distributed user interfaces, as seen in [Molina, 2006a], [Molina, 2006c] and [Vanderdonckt, 2007].

At the same time this doctoral work was carried out, it has been able to participate in regional and national projects funded by Junta de Comunidades de Castilla-La Mancha (PBC-03-003 and PAI-06-0093) and the Ministerio de Ciencia y Tecnología (TIN2004-08000-C03-01), tightly related to methodologies and virtual environments.

Part of this work has been done during a stay at the Univesité catholique de Louvain (UCL), Belgium, as a member of the Belgian Computer-Human Interaction (BCHI) group, and being supervised by professor Dr. Jean Vanderdonckt. This collaboration resulted in several publications co-authored with this professor. Thanks to him, our research group became fellow member of the SIMILAR Network of Excellence, funded by the 6th framework program of the European Commission, where the main aim is creating human-machine interfaces SIMILAR to human-human communication.

All in all, the proposed framework is not presented as a finished work but as a living being that must continuously evolve in order to accommodate the advances in the field of 3D user interfaces. Some ideas for further work are:

- The concept of sub-models is presented in the object meta-model so that each object can be described from different points of view, but sharing the same behaviour and function. However, only the graphics one has been given, others as the haptics one should be added.

- Supporting the proposed methodology with a CASE tool is desirable, as in IDEAS. In the simplest case, a Web site could guide developers in their work, as Kaur proposed, but even in this case the site could be enhanced with services to store and retrieve guides and tools, as a public repostitory.

- TRES-D presents a process model and recommends tools for each activity, but it does not achieve the automation that was aimed when the first proposal, IDEAS-3D, was presented. Further work on the proposed meta-models must be done in order to address that goal. The fact that VUIToolkit is based on the CUI model of UsiXML is one step in that direction, but there is still work to be done.

- New decission trees could be produced for other universal tasks, such as navigation and system control. Furthermore, it should be discussed the way these tools are presented, as a pattern format could be more useful to detail when each recommendation applies.

- As for the VUIToolkit, there are still CUI classes that can be added, such as the CUI Box. VUIToolkit could also be ported to other development environments beyond Web3D languages, such as VR Juggler, a widely-used programming library for VR applications.

- VR Juggler is also used at our lab, currently to develop Collaborative Virtual Environments or CVEs. For this kind of environments, the proposed methodology also applies, with avatars being considered an important class in the object meta-model. In habited virtual worlds, the authors of SENDA realized that specific tools were needed, and so they proposed the Use Concept. However, beyond conversational worlds, collaboration puts forward important challenges that open a whole new line of research.

Appendix A
The VUIToolkit library

A.1 Summary

There are many reasons to support 2D user interfaces in 3D environments, and many of them have been stated in this Thesis. At this point, it is assumed that such support must be given, and so the aim is to make integration possible, thanks in this case to the VUIToolkit library that is described in this appendix. VUIToolkit is a set of 3D widgets, having produced two different versions of that set, one based on VRML97 standard language and another one based on the third and most recently update of such language, known as X3D (*eXtensible 3D graphics*).

In order to immerse a plain interface in the three dimensions of the space, one idea is "to take applications as textures", and another one is "to virtualize user interfaces". This work is based on the latter, ensuring a smooth transition between 2D user interfaces and 3D environments, while taking into account specific characteristics of the 3D world. Thus, the depth cues usually represented in 2D user interfaces can be turned into true ones instead of being simulated. For instance, when a button is rendered, its depth could be a true one, not simulated, so that when the user presses or releases the button, it moves along the third dimension as the user would expect, instead of showing a predefined sequence of images that simulates those movements with 2D drawings. This way, the behaviour of the widget, being transformed into a 3D widget, matches much better the mental model of the user.

Instead of creating different versions of the same user interface for rendering it in two or three dimensions, each interface can be specified just once with UsiXML (*User Interface eXtensible Markup Language*) [UsiXML, URL] and then rendered, for instance, as a final user interface in 3D thanks to the VUIToolkit library. The UsiXML language supports the Cameleon Reference Framework (Calvary, 2003), which defines user interface development for multi-context interactive applications (Figure A.1). Four different levels are identified: tasks and concepts (T&C), abstract user interface (AUI), concrete user interface (CUI), and final user interface (FUI). The CUI model consists of a hierachical decomposition of CIOs, or concrete interaction objects, whose specification is toolkit-independent, that is, regardless it is finally rendered in Java, HTML, or even VRML97/X3D. This characteristic is usually referred as platform-independency.

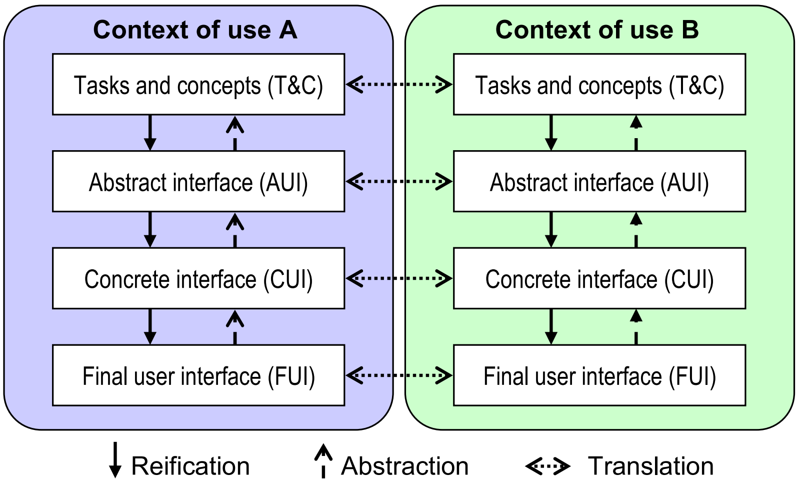


Figure A.1 Cameleon framework

The VUIToolkit was then developed starting from the object classes described in the concrete model of UsiXML language. However, neither VRML97 nor X3D offer elements that could be matched with defined CIOs. In this case, there is a need to fill this gap by introducing new widgets in the VRML97/X3D world by appropriate implementation. This is what happens with the CIOs that have been used as a starting point for the toolkit, for each of them a new widget has been included in the toolkit that can be used for the representation of the CIO in the final interface. VUIToolkit does not completely match the CUI model, though, as there are some CIOs and attributes that are defined in that concrete level of UsiXML but that are not used in the 3D world. There are also other elements and attributes which were added because they were necessary to describe the interface in the 3D world, such as those properties that allow the specification of the position and dimension of the widget.

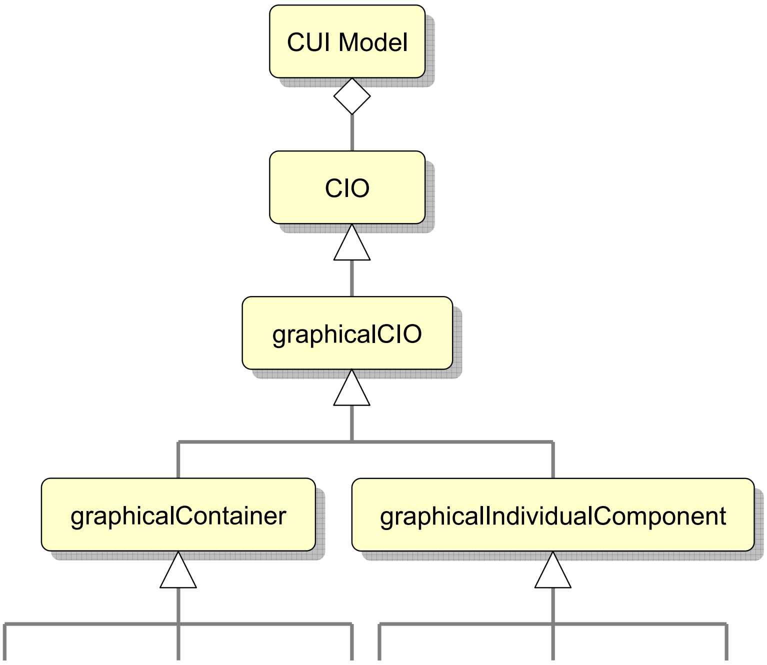
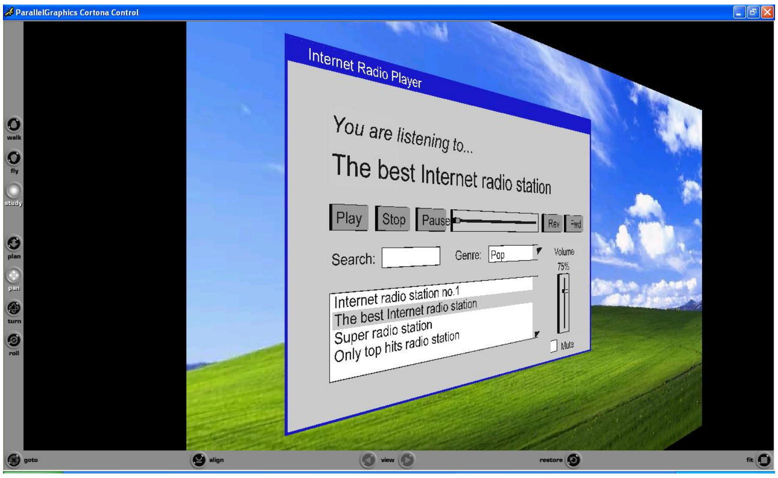


Figure A.2 Decomposition of a CUI model into concepts

The architecture of VUIToolkit was designed taking into account ideas from previous efforts, particularly those from the former VRML Widget Working Group [VRML WWG, URL], bringing them to practice thanks to the prototype creation mechanism or PROTOs. This mechanism can be found in both VRML97 and X3D languages, and is quite close to the concept of "class" offered by object-oriented programming languages. This way, both versions of the same library share the same architecture, and for each final 3D widget two prototypes were created, one for VRML97 and another one for X3D, each one consisting in: an interface base on a set of fields –for both setting initial values and sending/receiving events-, shapes that represent the geometry and appearance of the widget –customizable through the prototype's interface-, the sensors that make possible for the user to interact with the widget, an internal logic that defines the behaviour of each widget accordint to each state, and a set of routes that connect shapes, sensors and logic.



You are listening to...

The best Internet radio station

Genre: Pop

Volume 75%

Internet radio station no.1
The best Internet radio station
Super radio station
Only top hits radio station

Figure A.3 3D widgets of a final user interfaces renderer in Cortona player

Currently, the implemented prototypes are: *VUIWindow*, *VUIButton*, *VUIToggleButton*, *VUICheckBox*, *VUITextComponent*, *VUIImageComponent*, *VUISlider*, *VUICursor* and *VUIComboBox*. Besides, a prototype called *Screen* is used as an interaction surface where widgets can be rendered on in the 3D environment.

References

[García, 2005a] A.S. García, J.P. Molina y P. González. Exemplar VE design guidance tool for selection and manipulation interaction techniques. Proc. of 11th International Conference on Human-Computer Interaction, HCI International 2005, Las Vegas, Nevada, EE.UU., julio 2005. CD-ROM.

[García, 2005b] A.S. García, J.P. Molina y P. González. Aproximación a la evaluación de interfaces de Realidad Virtual. VI Congreso de Interacción Persona-Ordenador, Interacción'05, organizado dentro del primer Congreso Español de Informática, CEDI'2005, Granada, septiembre 2005.

[González, 2007] G. González, J.P. Molina, A.S. García, D. Martínez y P. González. Evaluación de técnicas para la introducción de texto en Entornos Virtuales Inmersivos. VIII Congreso de Interacción Persona-Ordenador, Interacción'05, organizado dentro del segundo Congreso Español de Informática, CEDI'2007, Zaragoza, septiembre 2007, pp. 313-321.

[Molina, 2002] J.P. Molina, P. González y M.D. Lozano. A Unified Envisioning of Future Interfaces. Proc. of the Second IASTED Int. Conf. Visualization, Imaging and Image Processing (VIIP 2002), Málaga, Spain, septiembre 2002, ACTA Press, pp. 185-190.

[Molina, 2003a] J.P. Molina, P. González, M.D. Lozano, F. Montero y V. López. Bridging the gap: developing 2D and 3D user interfaces with the IDEAS methodology. En: J. Jorge *et al.* (eds.), DSV-IS: Issues in Designing New-generation Interactive Systems, Proc. of the Tenth Workshop on the Design, Specification and Verification of Interactive Systems, Springer-Verlag Lecture Notes on Computer Science, 2003, pp. 303-315.

[Molina, 2003b] J.P. Molina, P. González y M.D. Lozano. Desarrollo de interfaces de usuario tridimensionales: un enfoque metodológico basado en IDEAS. Actas del Cuarto Congreso Internacional Interacción Persona-Ordenador, Interacción 2003, Vigo, junio 2003, Reprogalicia Edicións. CD-ROM.

[Molina, 2003c] J.P. Molina, P. González y M.D. Lozano. Developing 3D UIs using the IDEAS Tool: A case of study. En: J. Jacko and C. Stephanidis (eds.), Human - Computer Interaction, Theory and Practice, Proc. of 10th Int. Conf. on Human-Computer Interaction, HCI International 2003, Creta, Grecia, junio 2003, Part I, Volume 1, pp. 1193-1197.

[Molina, 2004] J.P. Molina y P. González. Model-based design and new user interfaces: current practices and opportunities. First Int.l Workshop on Making model-based user interface design practical: usable and open methods and tools, dentro de IUI / CADUI, Funchal, Madeira, Portugal, enero 2004. CEUR-Workshop Proceedings, Vol. 103. URL: http://CEUR-WS.org/Vol-103

[Molina, 2005a] J.P. Molina, J. Vanderdonckt, F. Montero y P. González. Towards Virtualization of User Interfaces base don UsiXML. Proc. of ACM Web3D 2005 Symposium, 10th Int. Conf. on 3D Web Technology, Bangor, Reino Unido, marzo-abril 2005, ACM, pp. 169-179.

[Molina, 2005b] J.P. Molina, A.S. García, V. López & P. González. Developing VR applications: the TRES-D methodology. First Int. Workshop on Methods and Tools for developing Virtual Reality Applications (MeTo-VR). 11th Int. Conf. on Virtual Systems and Multimedia (VSMM), Gante, Bélgica, octubre 2005. URL: http://wise.vub.ac.be/metovr/proceedings.html

[Molina, 2006a] J.P. Molina, J. Vanderdonckt & P. González. Direct manipulation of User Interfaces for Migration. Proc. of 10th ACM Int. Conf. on Intelligent User Interfaces (IUI’2006), Sydney, Australia, enero-febrero 2006, pp. 140-147.

[Molina, 2006b] J.P. Molina, A.S. García, D. Martínez, F.J. Manjavacas, V. Blasco & P. González. An Interaction Model for the TRES-D Framework. Proc. of 13th IEEE Mediterranean Electrotechnical Conference (MELECON 2006), special session "New interaction paradigms in Virtual Environments", Benalmádena, Málaga, mayo 2006. CD-ROM.

[Molina, 2006c] J.P. Molina, J. Vanderdonckt, P. Gonzalez, A.F. Caballero, & M.D. Lozano. Rapid Prototyping of Distributed User Interfaces. En: G. Calvary *at al.* (eds.), Computer-Aided Design of User Interfaces V, Proc. of 6th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2006, Springer-Verlag Information Systems Series, 2006, pp. 151-166.

[Molina, 2006d] J.P. Molina, A.S. García, D. Martínez, F.J. Manjavacas, V. Blasco, V. López & P. González. The development of glove-based interfaces with the TRES-D methodology. Proc. of 13th ACM Symposium on Virtual Reality Software and Technology (VRST), Limassol, Chipre, noviembre 2006. pp. 216-219.

[Vanderdonckt, 2007] J. Vanderdonckt, H. Mendonça y J.P. Molina. Distributed User Interfaces in Ambient Environment. Proc. of 1st Int. Workshop on Model Driven Software Engineering for Ambient Intelligence Applications, MDSE4AmI'07, Springer-Verlag Lecture Notes in Computer Science, 2007, pp. 44-52.