Thesis submitted to the University of Castilla-La Mancha for the European degree of Doctor of Philosophy in Computer Science

Albacete, 29 February 2008

A Structured Approach to the Development of 3D User Interfaces

José Pascual Molina Massó

Ph.D. candidate

Dr. Pascual González López

Supervisor

I3A Instituto de Investigación en Informática de Albacete

LoUISE Laboratorio de Interacción con el Usuario e Ingeniería del Software

DSI Departamento de Sistemas Informáticos

UCLM Universidad de Castilla-La Mancha

Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

Contents

- **Introduction**
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

Introduction

- The **User Interface** (UI) is considered the most important factor in the success of an application.
- **Graphical User Interfaces** (GUIs) based on **Windows, Icons, Menus and Pointer** (WIMP) have become a *de facto* standard, their ease of learning and ease of use has made PCs accessible to a huge number of novel users.
- Nowadays, well-founded methods and practices allow a rapid development, highly automated, of WIMP systems.

Introduction

- However, current desktop interfaces are not much different from those running on first windows environments.
- It may seem they are good enough, but there is an ongoing research effort to find **Next Generation User Interfaces** that are not only based on 2D widgets such as menus, forms or toolbars [van Dam, 97], but offer richer interaction and match much better the characteristics and abilities of people [Furness, 01].
- As people live in a 3-D world, using the three-dimensions of space as a medium to interact with computers may be the answer to that quest.

Introduction

- **Three-dimensional User Interfaces** (3DUIs) are more difficult to design, implement and use [Herndon, 94].
- They are as old as WIMP systems but have not reached, in contrast, the same state of maturity that can be observed in desktop interfaces [Bowman, 01].
- Progress has been done in identifying the so-called *universal* tasks, proposing interaction techniques for them and understanding human factors [Bowman, 06].

Introduction

- In spite of that progress, **intuition and experience remain the key** to successful development of 3DUIs.
- 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.
- Many methods have been proposed, the problem is that they approach the development from the point of view of a single role, thus offering a partial solution.
- **It is not only about code, interaction or content but all these things together.**

Introduction

- Motivation and scope of this Thesis:
  - The doctoral dissertation tackles **the problem of developing 3DUIs** by first questioning whether proposed methods are suitable for the development, in general, of these interfaces and, if they are not, which process should then be followed.
  - Thus, a critical review of existing design and development methods has been carried out.
  - Then, a new methodological framework, named **TRES-D**, has been proposed. In addition to that, a couple of advise tools and a 3D widget toolkit are also presented.
  - But, before going into details, a better understanding of 3DUIs and their design is needed.

Contents

- Introduction
- **The 3D user interface and its design**
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

The 3D user interface and its design

- A formal definition for 3DUI
- Applications of 3DUIs
- A new continuum
- Elements of 3DUIs
- Design of 3DUIs

The 3D user interface and its design

- A formal definition for 3DUI
  - 3DUIs are not usually presented as such, but using other names, as for instance *post-WIMP*, *non-WIMP*, *post-PC*, *Virtual Reality* (VR), *Virtual* Environments (VE), *Augmented Reality* (AR), *interactive 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.

The 3D user interface and its design

- A formal definition for 3DUI (cont'd)
  - As a definition, a **3D user interface** is…

> *…a human-computer interface in which the language used by the user to introduce commands and information into the computer, and/or the language used by the computer to present information to the user, are based on the physical space and its three dimensions*.

  - This general definition is intended to cover the different ways that a 3D user interface can adopt, which can be found in different kinds of applications.

The 3D user interface and its design

- Applications of 3DUIs
  - 3D technology has not yet found its **killer application** outside of video games and other specialized areas [Leavitt, 01]. Its use is sometimes forced because it's cool, rather than useful.
  - For that reason, 3DUIs have been accused of being ***a solution looking for a real problem***, and not matching expectations have brought headlines such as ***the failure of gloves and goggles***.
  - However, leaving hype aside, it must be recognized that some applications of 3DUIs have been more successful than others as a consequence of the experimental character of this technology.
  - Instead, it is better to identify which uses are appropriate for 3D technology based on past experiences, so we can benefit from their findings and do not incur in the same mistakes.

The 3D user interface and its design

- Applications of 3DUIs (cont'd)
  - Taking into account the classifications given by [Stuart, 01], [Shneiderman, 02] and [Sutcliffe, 03], and adding other uses not found in them, the following list has been produced:
    - Virtual Reality interfaces
    - Realism to bring computing to more users
    - 3D to overcome the physical limits of the screen
    - 3D to improve the organization of data and tasks
    - Information visualization
    - Creation and manipulation in three dimensions
    - Entertainment and fun
    - Augmented Reality interfaces

The 3D user interface and its design

- Applications of 3DUIs (cont'd)
  - Taking into account the classifications given by [Stuart, 01], [Shneiderman, 02] and [Sutcliffe, 03], and adding other uses not found in them, the following list has been produced:
    - Virtual Reality interfaces
    - **Realism to bring computing to more users**
    - 3D to overcome the physical limits of the screen
    - 3D to improve the organization of data and
    - Information visualization
    - Creation and manipulation in three dimens
    - Entertainment and fun
    - Augmented Reality interfaces

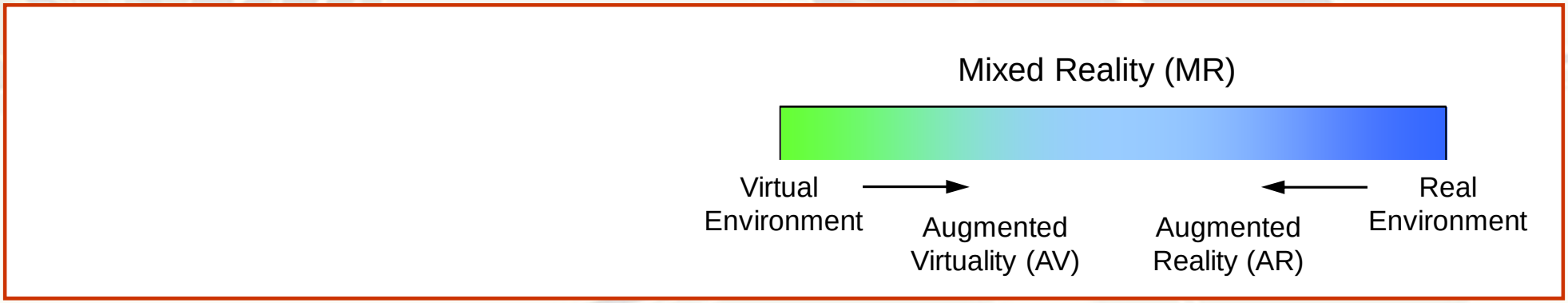
IBM RealPlaces [IBM, URL]

The 3D user interface and its design

- A new continuum
  - Another way to present 3DUIs to designers is to place them in the context of the evolution of human-computer interaction.
  - Systems have went through an evolution from 1D (CRT stream) to 2D (desktop GUI) [Fei, URL1]. Now, the third dimension is being progressively added to windows environments, while support for 2D GUIs is being added to VEs, resulting in hybrid 2D/3D UIs.

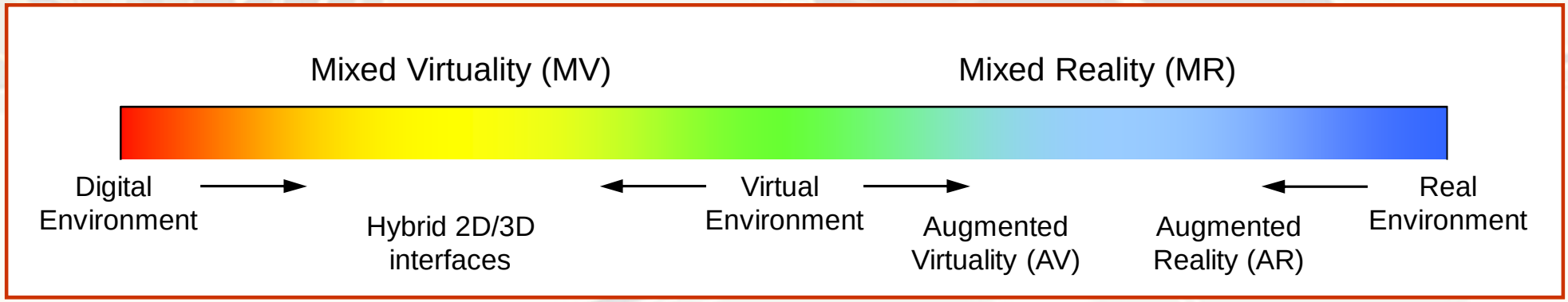
The 3D user interface and its design

- A new continuum (cont'd)
  - The idea is to extend the **Reality-Virtuality continuum** introduced by Milgram and Kishino [Milgram, 94]…



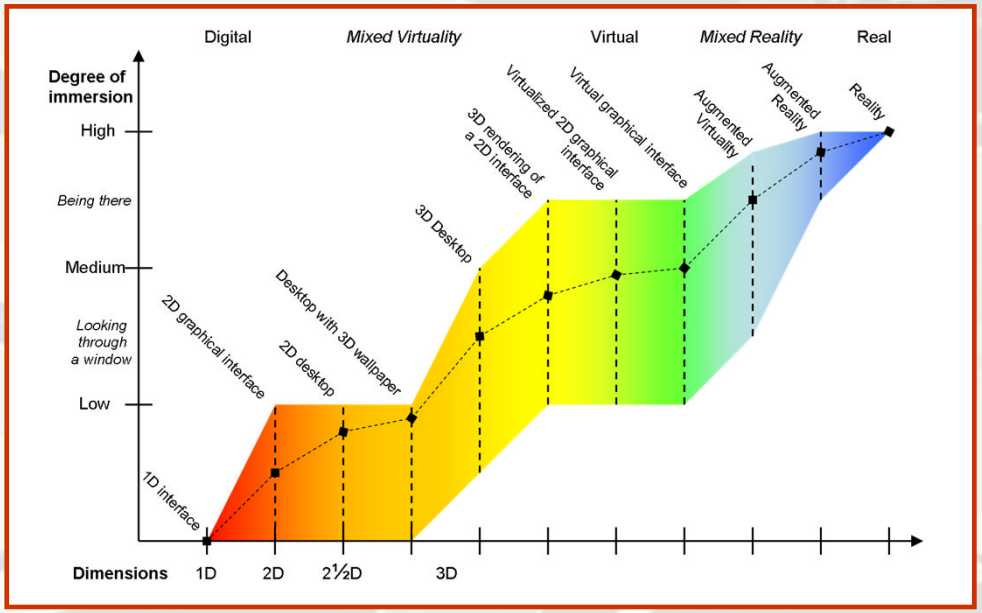
The 3D user interface and its design

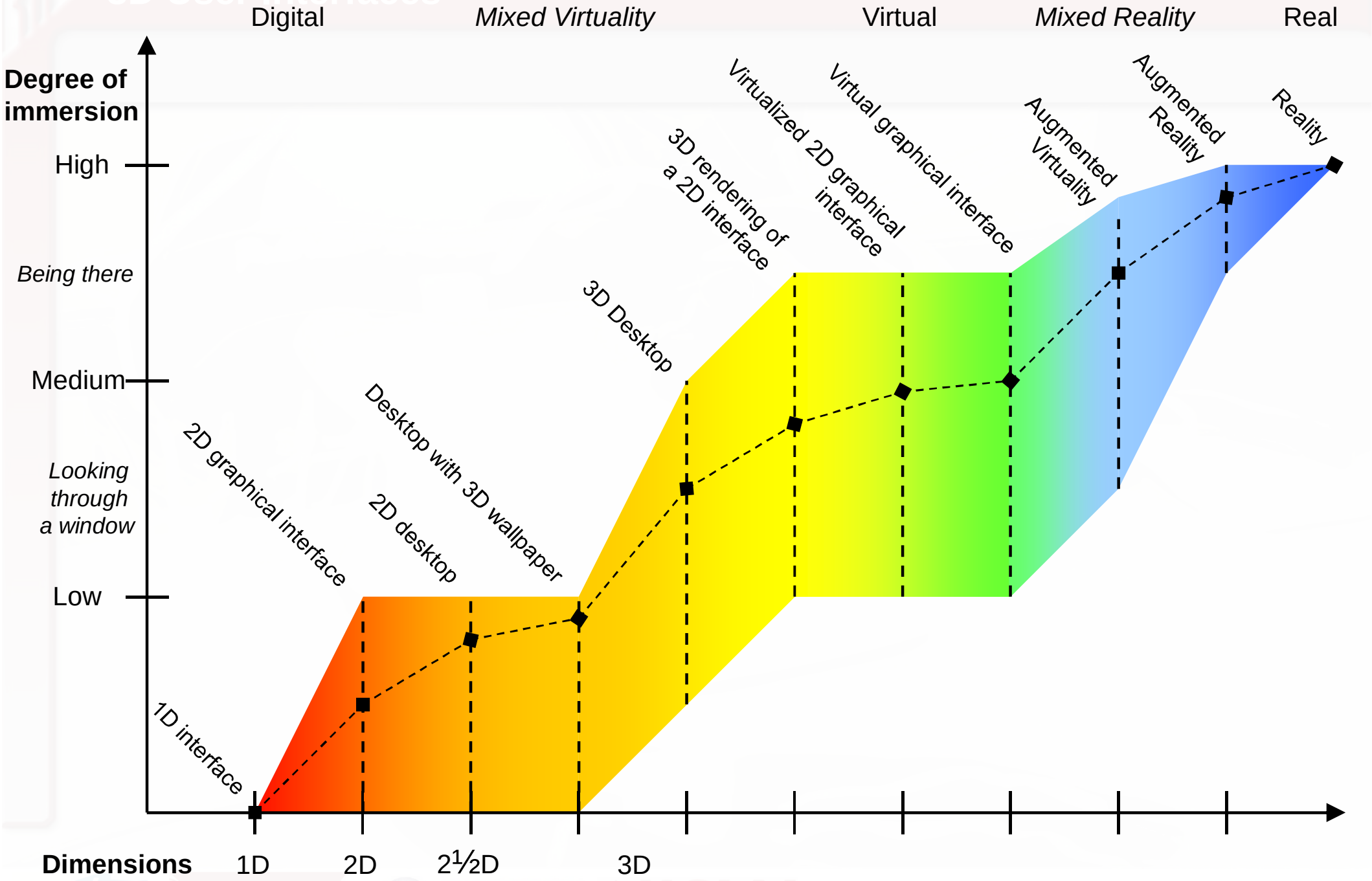
- A new continuum (cont'd)
  - The idea is to extend the **Reality-Virtuality continuum** introduced by Milgram and Kishino [Milgram, 94] …
  - … from the Virtual Environment endpoint to a new *Digital* one.
  - The space between both points is named as *Mixed Virtuality*, where 2D and 3D meet.

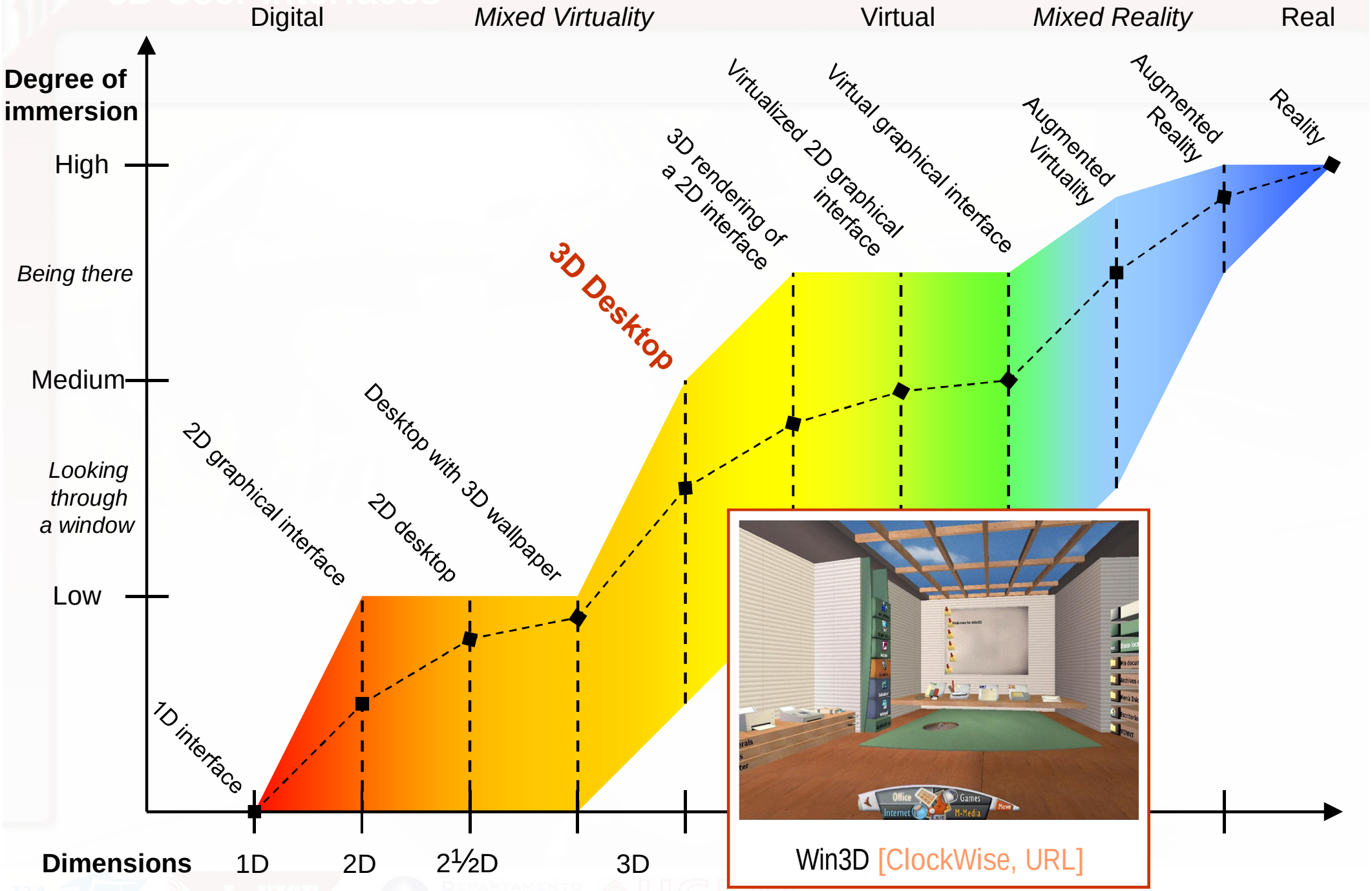


The 3D user interface and its design

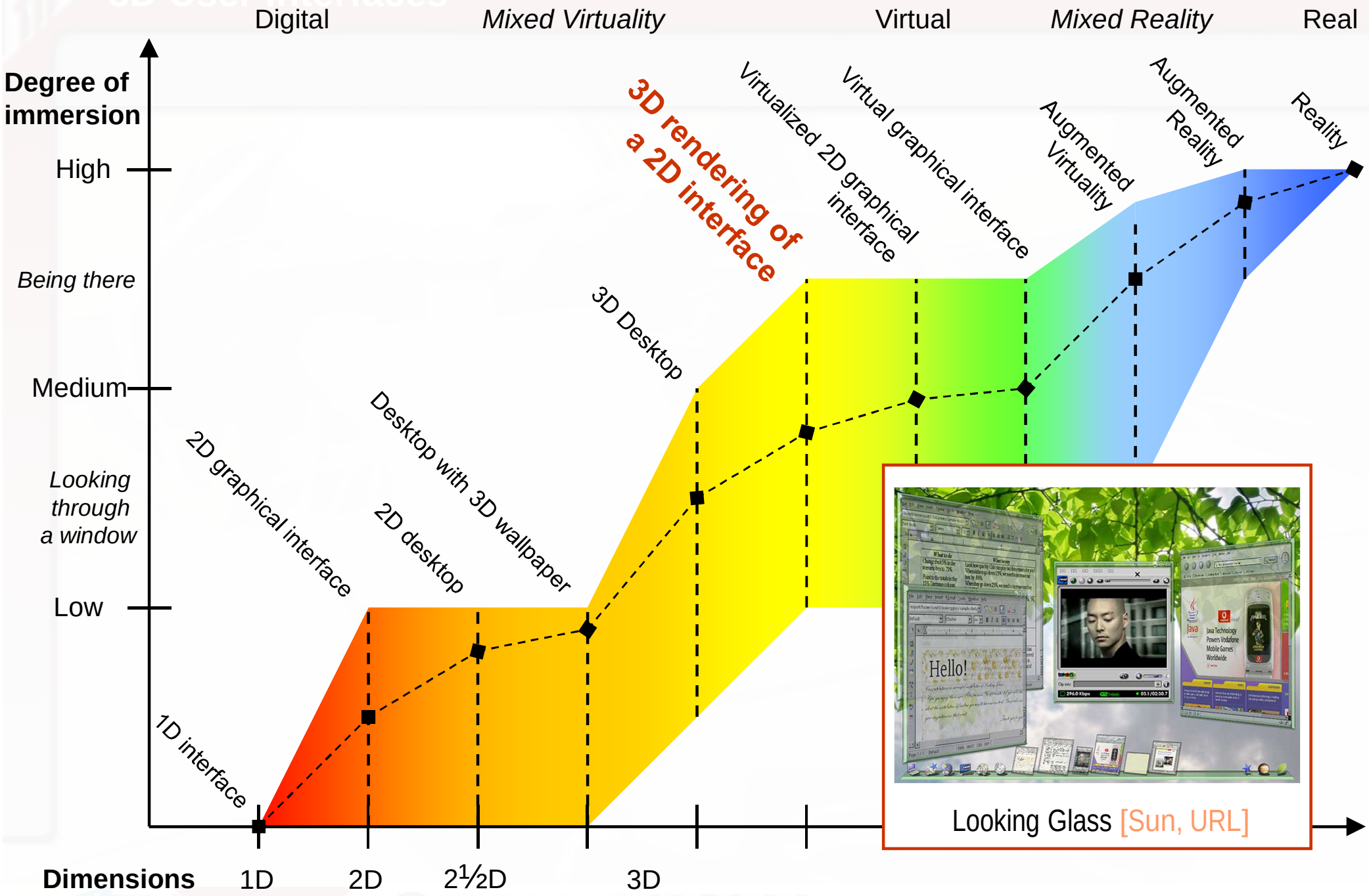
- A new continuum (cont'd)
  - The new **digital-virtual-real continuum**
    - One axis for the number of dimensions (1D, 2D, 2½D, 3D)
    - The other one for the degree of immersion, from *looking through a window* to *being there* (zero, low, medium, high)

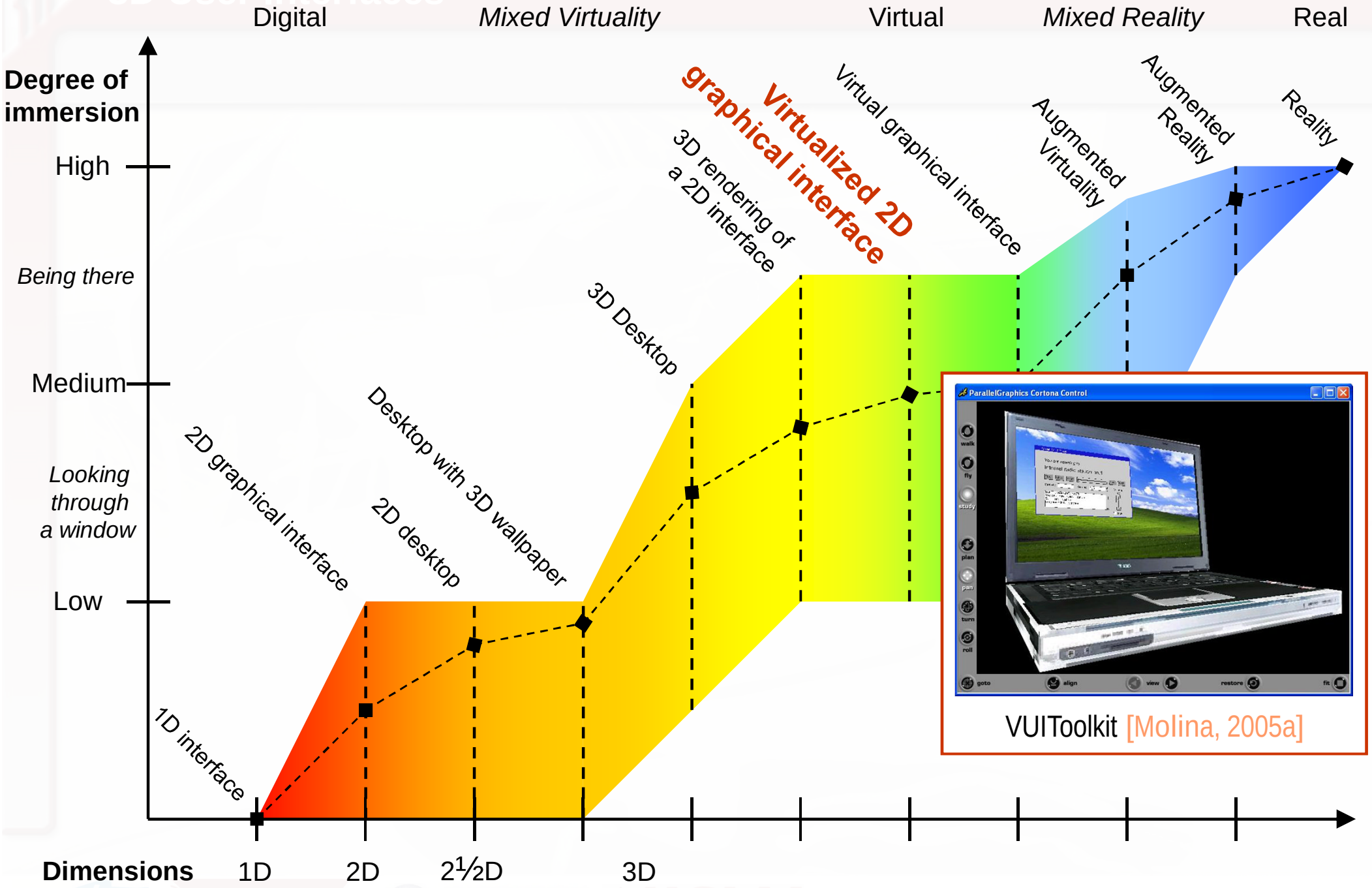



Digital
Mixed Virtuality
Virtual
Mixed Reality
Real
Degree of immersion
High
Being there
Medium
Looking through a window
Low
1D interface
2D graphical interface
2D desktop
Desktop with 3D wallpaper
3D Desktop
3D rendering of a 2D interface
Virtualized 2D graphical interface
Virtual graphical interface
Augmented Virtuality
Augmented Reality
Reality
Dimensions
1D
2D
2½D
3D



Win3D [ClockWise, URL]


Digital
Mixed Virtuality
Virtual
Mixed Reality
Real
Degree of immersion
High
Being there
Medium
Looking through a window
Low
1D interface
2D graphical interface
2D desktop
Desktop with 3D wallpaper
3D Desktop
3D rendering of a 2D interface
Virtualized 2D graphical interface
Virtual graphical interface
Augmented Virtuality
Augmented Reality
Reality
Hello!
Looking Glass [Sun, URL]
Dimensions
1D
2D
2½D
3D


Digital
Mixed Virtuality
Virtual
Mixed Reality
Real
Degree of immersion
High
Being there
Medium
Looking through a window
Low
1D interface
2D graphical interface
2D desktop
Desktop with 3D wallpaper
3D Desktop
3D rendering of a 2D interface
Virtualized 2D graphical interface
Virtual graphical interface
Augmented Virtuality
Augmented Reality
Reality
Dimensions
1D
2D
2½D
3D
ParallelGraphics Cortona Control
VUIToolkit [Molina, 2005a]

The 3D user interface and its design

- Elements of 3DUIs
  - Having defined the design space, the next step is to identify and describe the different elements that compose these interfaces.
  - Four basic elements were described:
    - 3D space
    - Physical and virtual objects
    - Behaviour
    - Interaction

The 3D user interface and its design

- Elements of 3DUIs
  - Having defined the design space, the next step is to identify and describe the different elements that compose these interfaces.
  - Four basic elements were described:
    - 3D space
    - Physical and virtual objects
    - Behaviour
    - **Interaction**

The 3D user interface and its design

- Elements of 3DUIs: Interaction
  - According to [Foley, 96], four are the basic elements of UIs, which can be compared with natural language:

Interactive dialogues	Sentences
Interaction tasks	Meaning of words
Interaction techniques	Words
Actions with input devices	Letters

  - This elements have been used to gain insight of interaction. Later, they will serve as a basis for a new meta-model.

The 3D user interface and its design

- Elements of 3DUIs: Interaction (cont'd)
  - As regards interaction tasks, different lists of **universal tasks** can be found. For instance, these are for VR applications:

Bowman y Hodges, 99	Boyd y Sastry, 99	Sutcliffe, 03
▪ Travel ▪ Selection ▪ Manipulation Other: ▪ Release ▪ System control	▪ Navigation ▪ Selection ▪ Manipulation ▪ Data input	▪ Movement and navigation ▪ Manipulation and interaction of objects ▪ Conversation with other agents ▪ Non-VR features

The 3D user interface and its design

- Elements of 3DUIs: Interaction (cont'd)
  - As for the **interaction techniques**, it is agreed that each interaction task can be carried out using a number of interaction techniques.
  - **Widget** seems to be another term to name interaction techniques. Thus, [Conner, 92] introduces the term "three-dimensional widget", and gives as an example the virtual sphere. This example is also given as an interaction technique in [Foley, 96]. And, according to the latter, interaction techniques are called widgets in the X Windows system.

The 3D user interface and its design

- Elements of 3DUIs
  - The study of all these elements found a huge number of terms and proposed taxonomies, but more important it showed significant differences among authors.
  - **It will be necessary to put some order in all this mess,** particularly in the model of objects and of interaction.

The 3D user interface and its design

- Design of 3DUIs
  - According to [Shneiderman, 98], three are the pillars that support the successful design of the UI: guidelines documents & processes, UI software tools, and expert reviews & usability testing.
  - In the case of 2D, there are guidelines, APIs and other tools describe and provide the basic elements for desktop interfaces.
  - However, despite the programming libraries, notations and tools that already exist, the design of 3DUIs is much more complex than 2D, and have not reached the same state of maturity [Bowman, 01].
  - Even worst, it is still in *incunabula* state [Fencott, 01].

The 3D user interface and its design

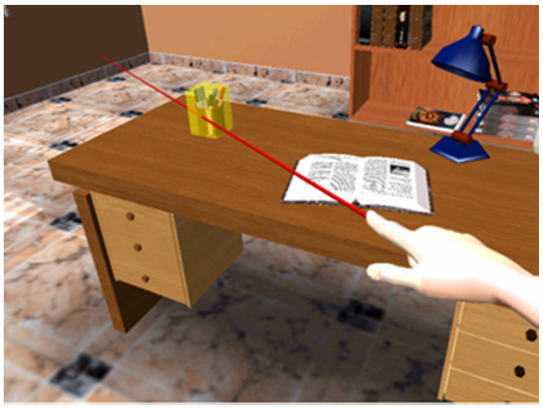
- Design of 3DUIs: Issues
  - Physical devices
  - Interaction techniques
  - Controls
  - 3D objects
  - Programming libraries
  - Prototypes and specification
  - Notations
  - Software tools
  - Mark-up languages
  - Guidelines, usability and test

The 3D user interface and its design

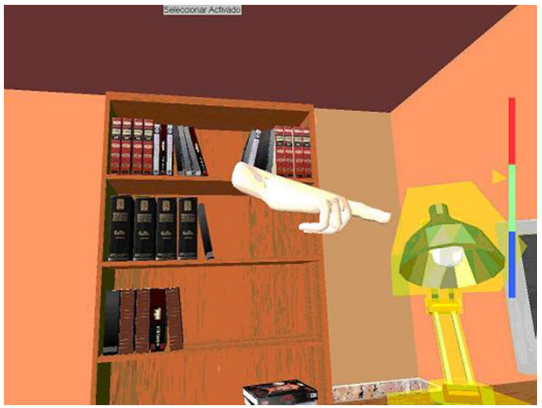
- Design of 3DUIs: Issues
  - Physical devices
  - **Interaction techniques**
  - **Controls**
  - 3D objects
  - Programming libraries
  - Prototypes and specification
  - **Notations**
  - Software tools
  - Mark-up languages
  - Guidelines, usability and test

The 3D user interface and its design

- Design of 3DUIs: Interaction techniques and controls
  - **There is no standard set of interaction techniques or controls.**
  - There are, however, some well-known interaction techniques.



Ray casting [Mine, 95]



Go-go [Poupyrev, 96]

  - As for the controls, some attempts have been made, such as the VRML Widgets Working Group [VRML WWG, URL], or the CONTIGRA project [Dachselt, 02]

The 3D user interface and its design

- Design of 3DUIs: Notations
  - The first asset in making designs is a good notation to record and discuss alternate possibilities [Shneiderman, 98].
  - State-transition diagrams are a common notation in desktop interfaces, where interaction is based on a turn-taking dialogue.
  - **Interaction in 3DUIs does not usually follow that model**, though. Instead, they are considered as hybrid systems of continuous devices and digital control.
  - Several authors have proposed different notations to represent data flow and control, such as the diagrams used in [Jacob, 96] or the *flownets* introduced in [Smith, 99].

The 3D user interface and its design

- Design of 3DUIs
  - All in all, even if the details of 3DUIs 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.

Contents

- Introduction
- The 3D user interface and its design
- **Methods for the development of 3DUIs**
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

Methods for the development of 3DUIs

- 26 methods have been described and analyzed.
- They have been classified into 7 categories:
  - Animation Studios methods (1)
  - Building PC-based virtual worlds (7)
  - Participative approaches (2)
  - Task analysis for building VEs (4)
  - Software engineering-oriented methods (3)
  - Beyond PC-based virtual worlds (5)
  - Methodologies from HCI (4)
- The study has been completed with the description of a last methodology, IDEAS-3D, as a first proposal.

Methods for the development of 3DUIs

1) Animation Studios methods
   - Only one method has been reviewed:

Pixar production pipeline	[Reddy, 05]

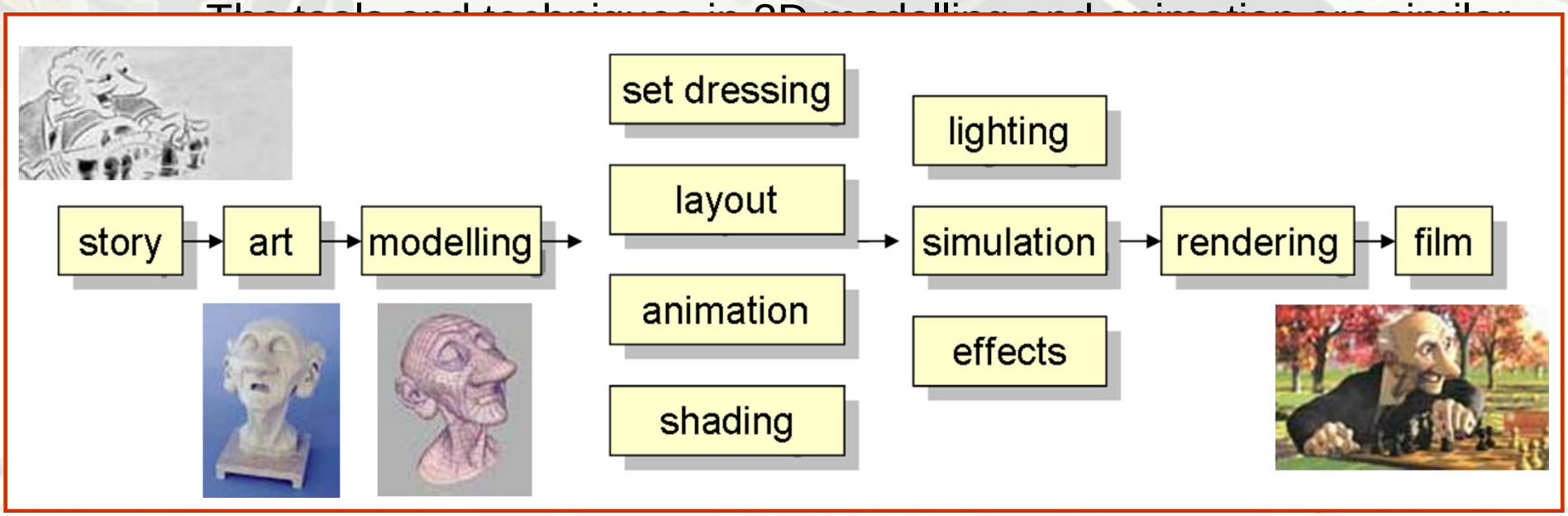
   - Things to remark:
     - The order followed to bring scenes from paper to computer
     - The tools and techniques in 3D modelling and animation are similar to those employed in the creation of virtual worlds. The main difference is interaction
     - The continuous revision of reels showing progress made in each scene

Methods for the development of 3DUIs

1) Animation Studios methods
   - Only one method has been reviewed:

Pixar production pipeline	[Reddy, 05]

   - Things to remark:
     - The order followed to bring scenes from paper to computer



Methods for the development of 3DUIs

1) Animation Studios methods
   - Only one method has been reviewed:

Pixar production pipeline	[Reddy, 05]

   - Things to remark:
     - The order followed to bring scenes from paper to computer
     - The tools and techniques in 3D modelling and animation are similar to those employed in the creation of virtual worlds. The main difference is interaction
     - The continuous revision of reels showing progress made in each scene

Methods for the development of 3DUIs

2) Building PC-based virtual worlds
   - Seven methods have been reviewed:

Current VR application development, according to	[Kim, 98]
Current VE development, according to	[Smith, 01]
EON Studio workflow	[EON, URL]
Current approaches to VE design, according to	[Kaur, 98]
VE design process	[Kaur, 98]
Creating an interactive 3D product using VRML	[Hay, URL]
Building X3D worlds	[Daly, 02]

Methods for the development of 3DUIs

2) Building PC-based virtual worlds (cont'd)

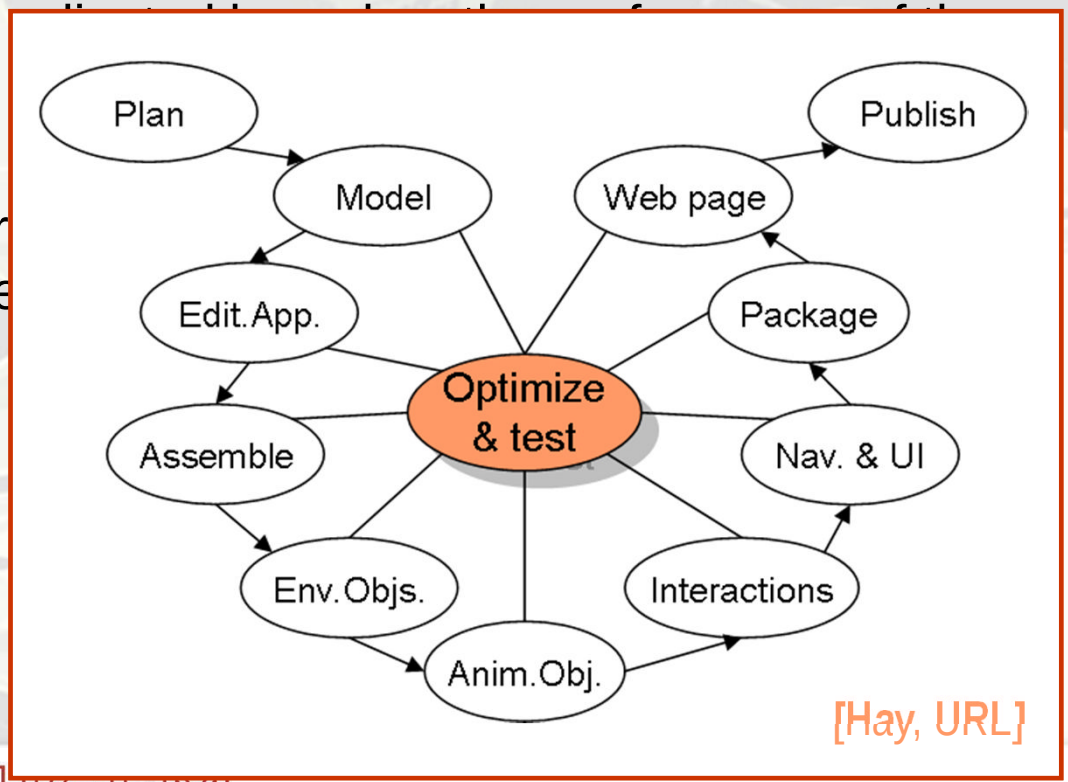
– Things to remark:

- The interactive nature of virtual worlds introduces the need for reaching the proper performance and avoiding usability problems.
- The first issue is tackled by an **optimization process**, the level of detail (LOD) of objects is adjusted based on the performance of the graphics hardware. That reference should be elicited in a plan or design stage previous to construction.
- The second issue is addressed by following some **guidelines** at the design stage, such as the ones proposed in [Kaur, 1998].

Methods for the development of 3DUIs

2) Building PC-based virtual worlds (cont'd)

- Things to remark:
  - The interactive nature of virtual worlds introduces the need for reaching the proper performance and avoiding usability problems.
  - The first issue is tackled by an **optimization process**, the level of detail (LOD) of objects is
    graphics hardware. That
    design stage previous to
  - The second issue is addr
    design stage, such as the



[Hay, URL]

Methods for the development of 3DUIs

2) Building PC-based virtual worlds (cont'd)

– Things to remark:

- The interactive nature of virtual worlds introduces the need for reaching the proper performance and avoiding usability problems.
- The first issue is tackled by an **optimization process**, the level of detail (LOD) of objects is adjusted based on the performance of the graphics hardware. That reference should be elicited in a plan or design stage previous to construction.
- The second issue is addressed by following some **guidelines** at the design stage, such as the ones proposed in [Kaur, 1998].

Methods for the development of 3DUIs

3) Participative approaches

- Methods of the previous group assumes that just one developer carries out the development. It also assumes that he or she knows well the content of the virtual world or the preferences of the future user.
- This third group promotes a higher involvement of both the **user** and the **domain expert** in the development, in order to obtain a product that satisfies the user, and with the right content.
- Two methods have been reviewed:

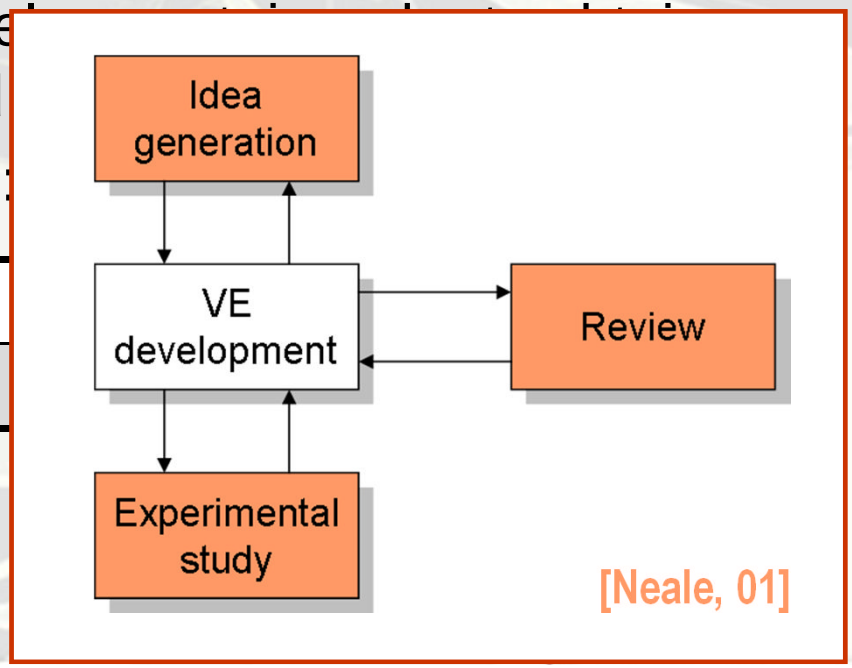
VE user-centred development process	[Neale, 01]
3D content-centred methodology	[Celentano, 01]

Methods for the development of 3DUIs

3) Participative approaches

- Methods of the previous group assumes that just one developer carries out the development. It also assumes that he or she knows well the content of the virtual world or the preferences of the future user.
- This third group promotes a higher involvement of both the **user** and the **domain expert** in the deve
  product that satisfies the user, and
- Two methods have been reviewed:

VE user-centred development process
3D content-centred methodology



Methods for the development of 3DUIs

4) Task analysis for building VEs
   - Up to this moment, reviewed methods were oriented to the content. However, there are other environments where the user does have to perform certain tasks and, therefore, the development must include the analysis of them.
   - Four methods have been reviewed:

Sequential evaluation of VEs	[Gabbard, 99]
VE development	[Pereira, 00] [García, 01]
VEDS: VE Development Structure	[Eastgate, 01] [Wilson, 02] [D'Cruz, 03]
Creating virtual worlds using X3D	[Polys, 05]

Methods for the development of 3DUIs

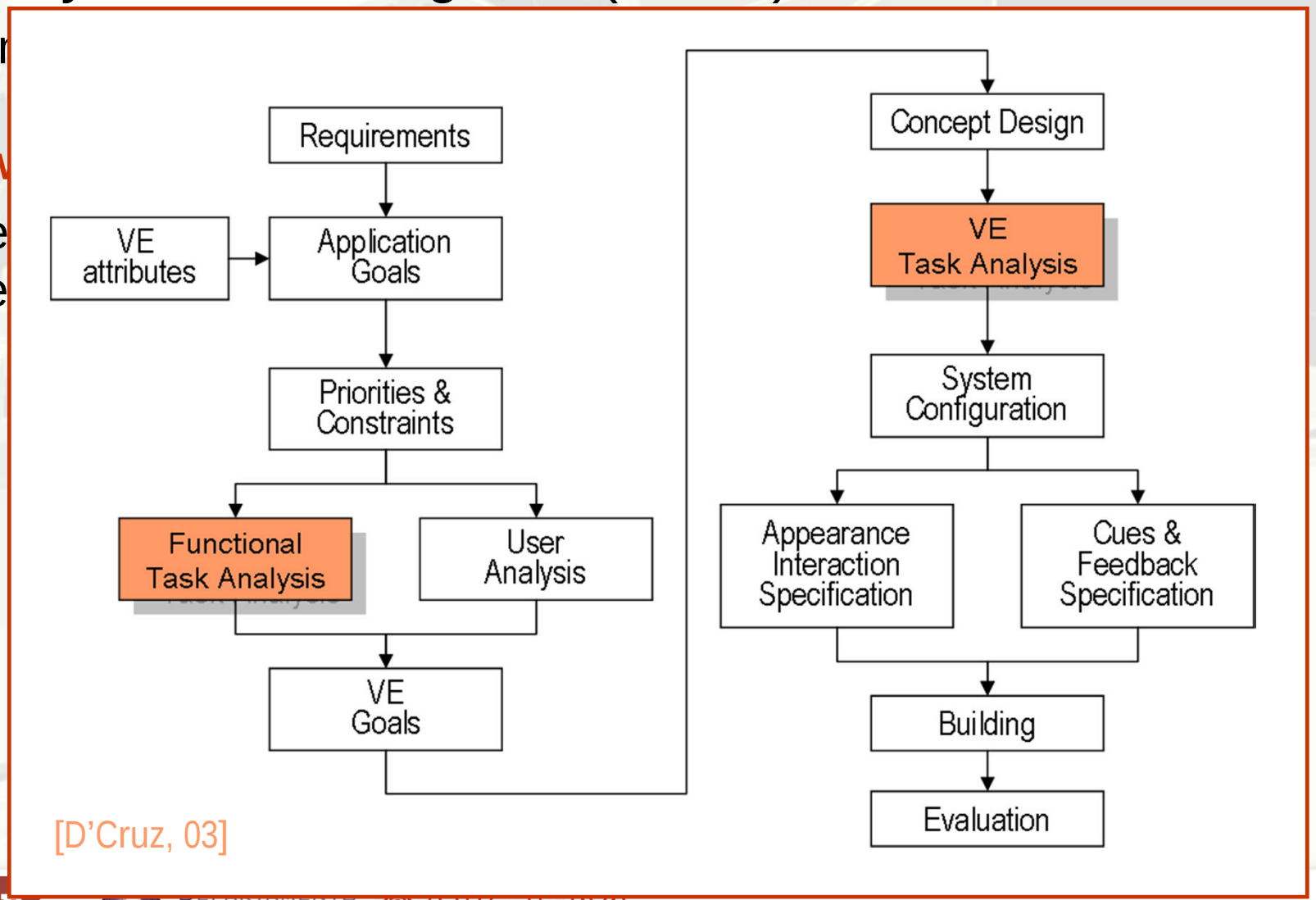
4) Task analysis for building VEs (cont'd)
   - One important thing to remark is the difference between the real world tasks and the virtual world tasks, which mean that not one, but **two task analysis** must be carried out.
   - There are still some doubts, however, regarding the way tasks are related to content.

Methods for the development of 3DUIs

4) Task analysis for building VEs (cont'd)

– One ir
world
but tw

– There
are re



[D'Cruz, 03]

Methods for the development of 3DUIs

4) Task analysis for building VEs (cont'd)
   - One important thing to remark is the difference between the real world tasks and the virtual world tasks, which mean that not one, but **two task analysis** must be carried out.
   - There are still some doubts, however, regarding the way tasks are related to content.

Methods for the development of 3DUIs

5) Software engineering-oriented methods
   - Many methods seem to leave the programming task, and the software engineering with it, in a second place.
   - Software engineering methods and programming practices are specially important when development relies on a set of software libraries and not just an integrated authoring environment.
   - Three methods have been reviewed:

A prototype design methodology for VEs	[Fencott, 99]
CLEVR: Concurrent and Level by level VR development of VR systems	[Seo, 2001]
SENDA: Metodología de desarrollo de mundos virtuales habitados	[Méndez, 01] [Sánchez, 01, 03, 05]

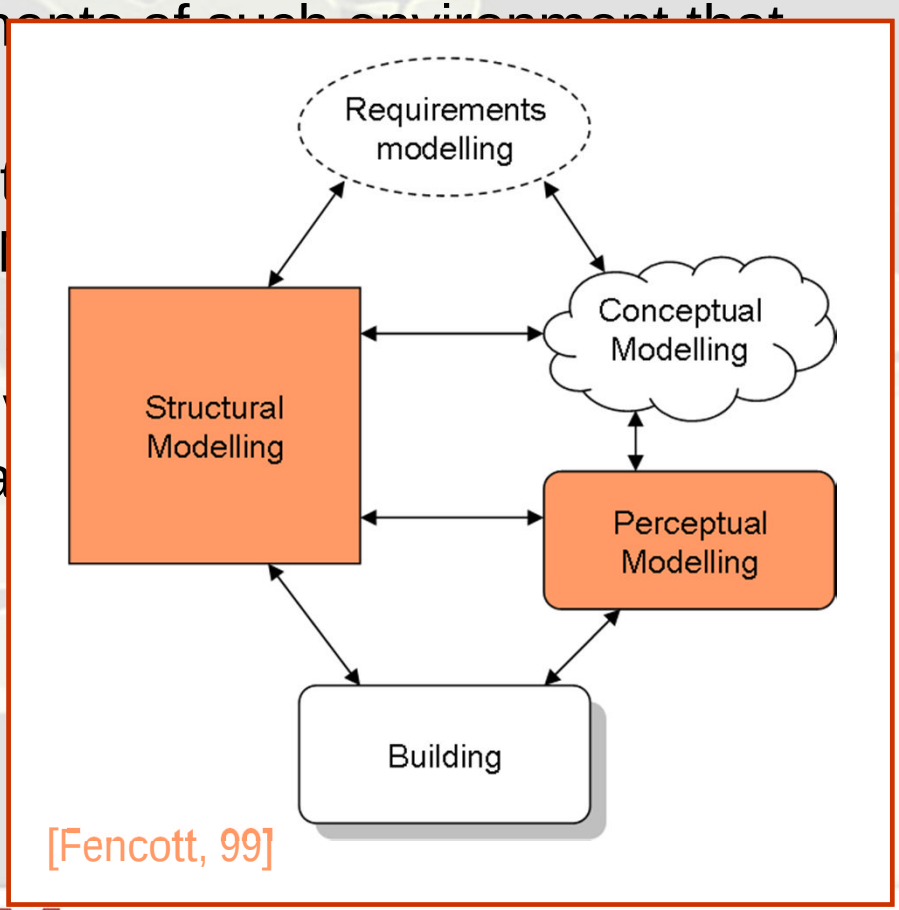
Methods for the development of 3DUIs

5) Software engineering-oriented methods (cont'd)
   - This group of methodologies demonstrates that software engineering can also be applied to the **structural aspects** of the VE, and obviously to the components of such environment that have to be programmed.
   - Some aspects of the VE require the introduction of new diagrams, such as the *Use Concepts* in SENDA.
   - Other aspects are just out of the scope of software engineering, such as the **aesthetics** of the environment, according to [Fencott, 99]. *Perceptual Maps* may the used in this case.

Methods for the development of 3DUIs

5) Software engineering-oriented methods (cont'd)

- This group of methodologies demonstrates that software engineering can also be applied to the **structural aspects** of the VE, and obviously to the compon
have to be programmed.
- Some aspects of the VE require t
such as the *Use Concepts* in SE
- Other aspects are just out of the
such as the **aesthetics** of the en
[Fencott, 99]. *Perceptual Maps* ma



Methods for the development of 3DUIs

5) Software engineering-oriented methods (cont'd)

– Diagrams proposed by Fencott follow the work by [McIntosh, 00], based on the SBF model, which can be compared with [Kim, 98], a work previous to CLEVR.

	Kim et al., 98	McIntosh, 00
Function	Dataflow diagrams (DFDs) and CSL language	Use cases, sequence, collaboration diagrams
Behaviour	Statecharts and Message Sequence Diagrams (MSDs)	Sequence diagrams and statecharts
Form / Structure	Visual Object Spec. (VOS)	Entity-Relationship diagrams
Class diagrams	Before modelling views	After SBF modelling

Methods for the development of 3DUIs

5) Software engineering-oriented methods (cont'd)

	Fencott	CLEVR	SENDA
Use cases diagram	Structural modelling, user-VE relationships		Analysis, requirements definition, jointly with use concepts
Sequence diagram		Scenarios, specify interactions among objects	
Object diagram	Structural modelling, nodes in the scenegraph		
Class diagram	Structural modelling, programmed components	After the sequence diag., before modelling views	Static modelling
Dataflow diagram		Function, jointly with CSL language	
Statechart		Behaviour	After SBF modelling

Methods for the development of 3DUIs

6) Beyond PC-based virtual worlds
   - All previous methodologies assume that the user will face the computer through simple techniques based on the keyboard and mouse of a commonplace 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.
   - Thus, the methods of this sixth group pays higher attention to the interface in Virtual and Augmented Reality.
   - Some focus on theory, other offer more practical solutions, such as the flownets proposed in the INQUISITIVE project.

Methods for the development of 3DUIs

6) Beyond PC-based virtual worlds (cont'd)

– Five methods have been reviewed:

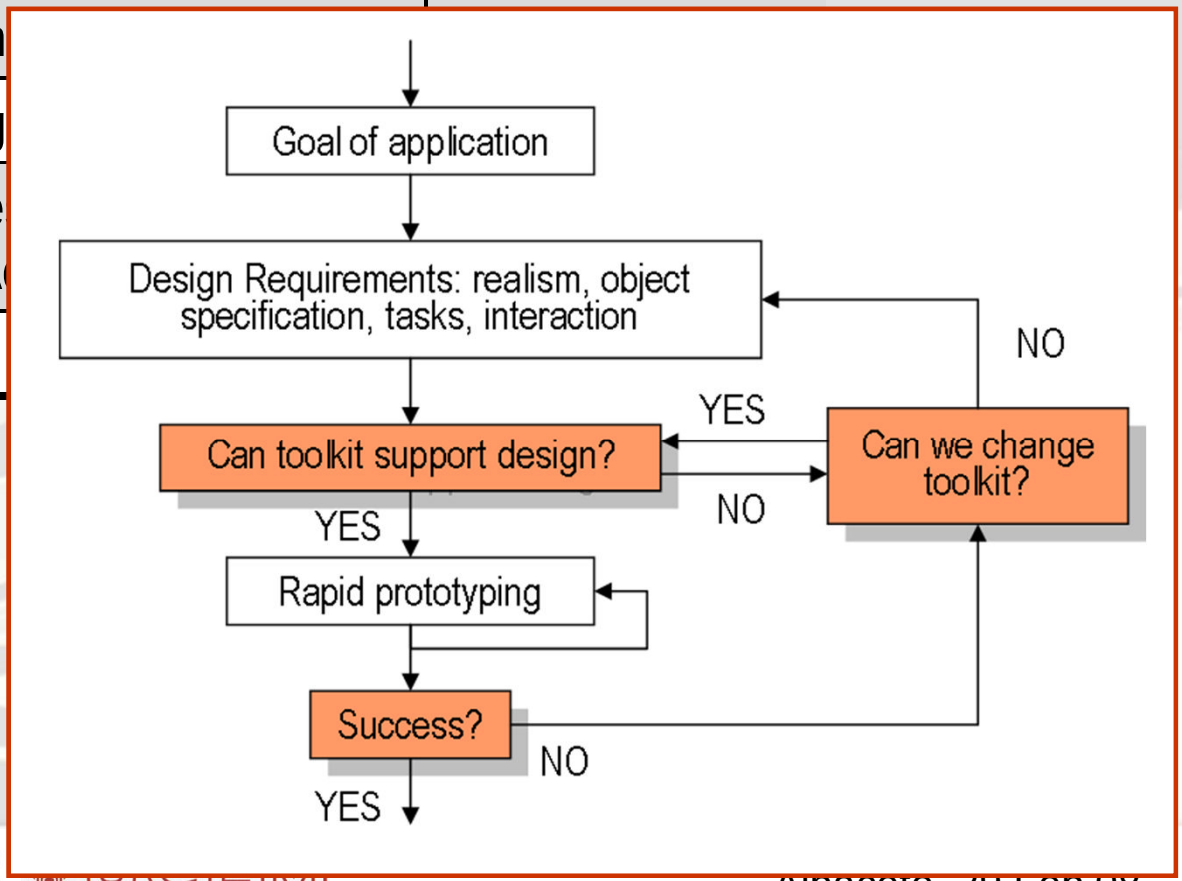
VE design process – INQUISITIVE project	[Smith, 01]
Interaction/content-driven design	[Parés, 01]
VRID: VR Interface Design	[Tanriverdi, 01]
MPIUA: Modelo de Proceso de la Ingeniería de la Usabilidad y de la Accesibilidad	[Granollers, 02]
VR design process	[Sutcliffe, 03]

Methods for the development of 3DUIs

6) Beyond PC-based virtual worlds (cont'd)

– Five methods have been reviewed:

VE design process – INQUISITIVE project	[Smith, 01]
Interaction/content-driven	
VRID: VR Interface Desig	
MPIUA: Modelo de Proce de la Usabilidad y de la A	
VR design process	



Methods for the development of 3DUIs

6) Beyond PC-based virtual worlds (cont'd)

– It is interesting to compare the proposal made by Tanriverdi *et al.* with previous approaches.

Kim et al., 98	McIntosh, 00	Tanriverdi et al., 00
▪ Form ▪ Function ▪ Behaviour	▪ Structure ▪ Behaviour ▪ Function	▪ Graphics ▪ Behaviours ▪ Interactions ▪ Communications

Methods for the development of 3DUIs

6) Beyond PC-based virtual worlds (cont'd)

– A comparison of notations and tools reveal interesting too.

	VRID	CLEVR	INQUISITIVE
Dataflow	Dataflow graph	Dataflow Diagram (DFD)	Extending Petri nets to dataflow
Control	State-transition diagram	Statecharts	Petri nets
Notation	Related through links and conditions	A DFD may have a corresponding statechart	Combined in one single notation: flownets
Visual tool	VRED Editor	P-VoT	MARIGOLD Toolset

Methods for the development of 3DUIs

7) Methodologies from HCI
   - Last group is oriented to interfaces for business applications, mainly desktop and Web systems.
   - Four methods have been reviewed:

Design methodology	[Foley, 96]
LUCID: Logical User-Centred Interaction Design	[Cognetics, URL]
OVID	[Bardon, 01]
IDEAS	[Lozano, 01]

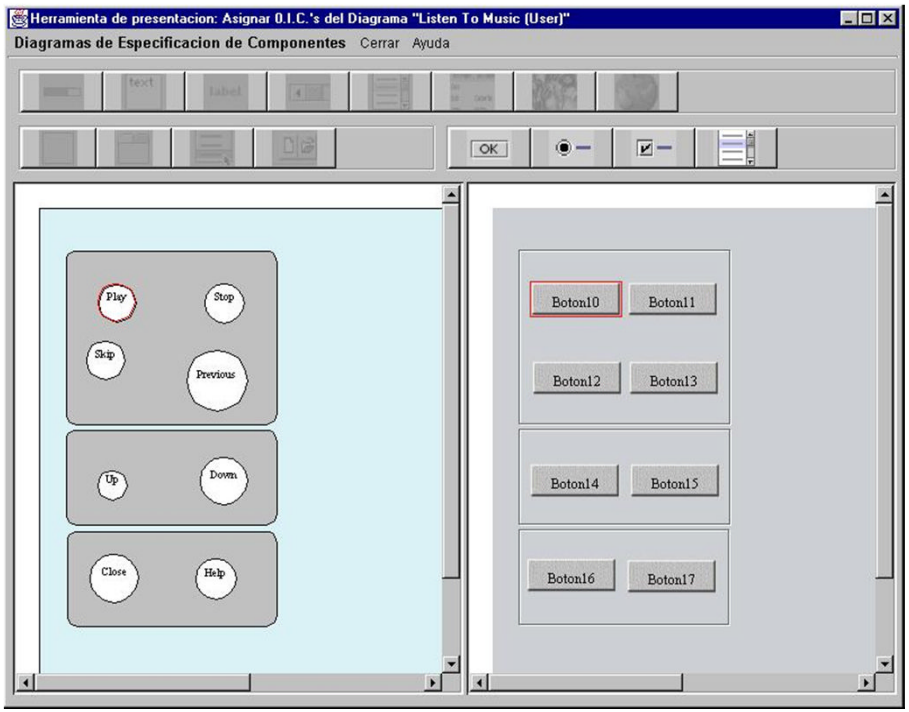
Methods for the development of 3DUIs

7) Methodologies from HCI (cont'd)
   - Attention is paid to analyze user tasks, model the user, design the navigation paths that connects windows, and specify the components that are part of each of them.
   - These methods rely on well-known components, and take advantage of tools that allow a **rapid prototyping** of the interface, even its **compilation** from abstract models, as in IDEAS.

Methods for the development of 3DUIs

7) Methodologies from HCI (cont'd)

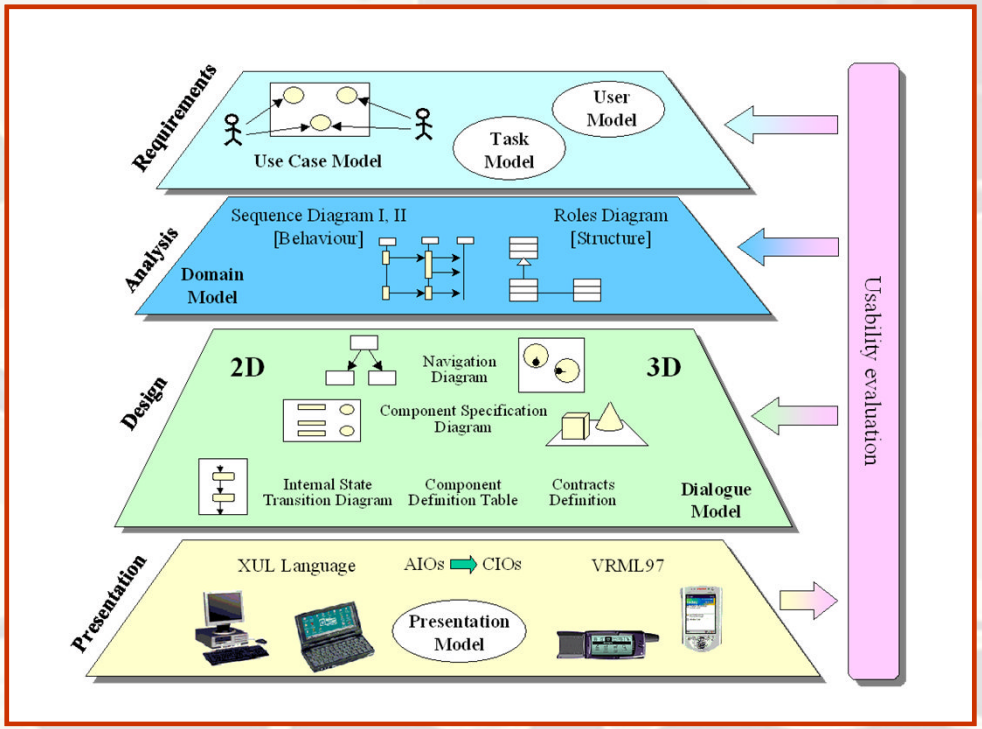
- Attention is paid to analyze user tasks, model the user, design the navigation paths that connects windows, and specify the components that are part of each of them.
- These methods rely on well-known components, and take advantage of tools that allow
even its compilation from ab

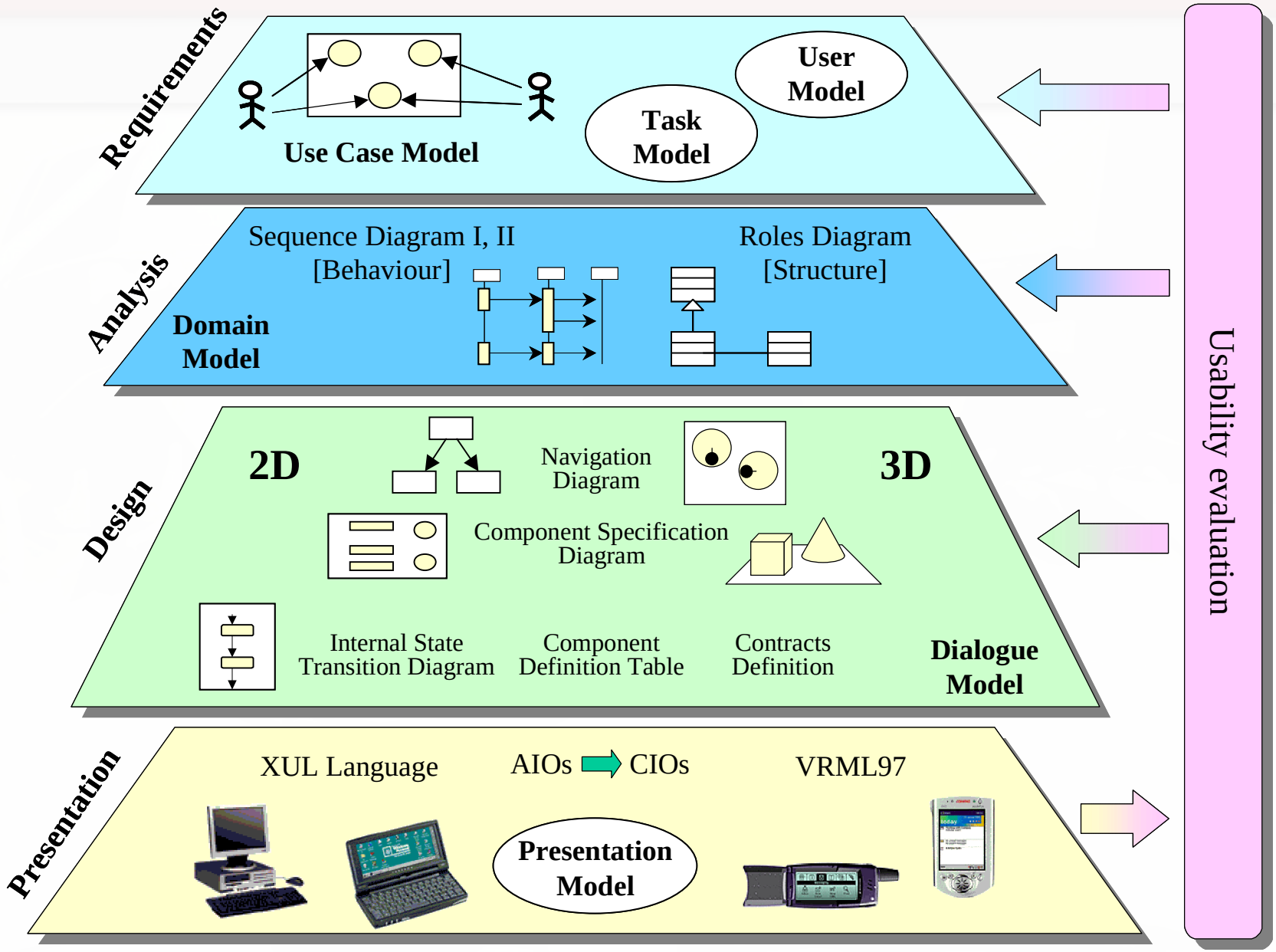


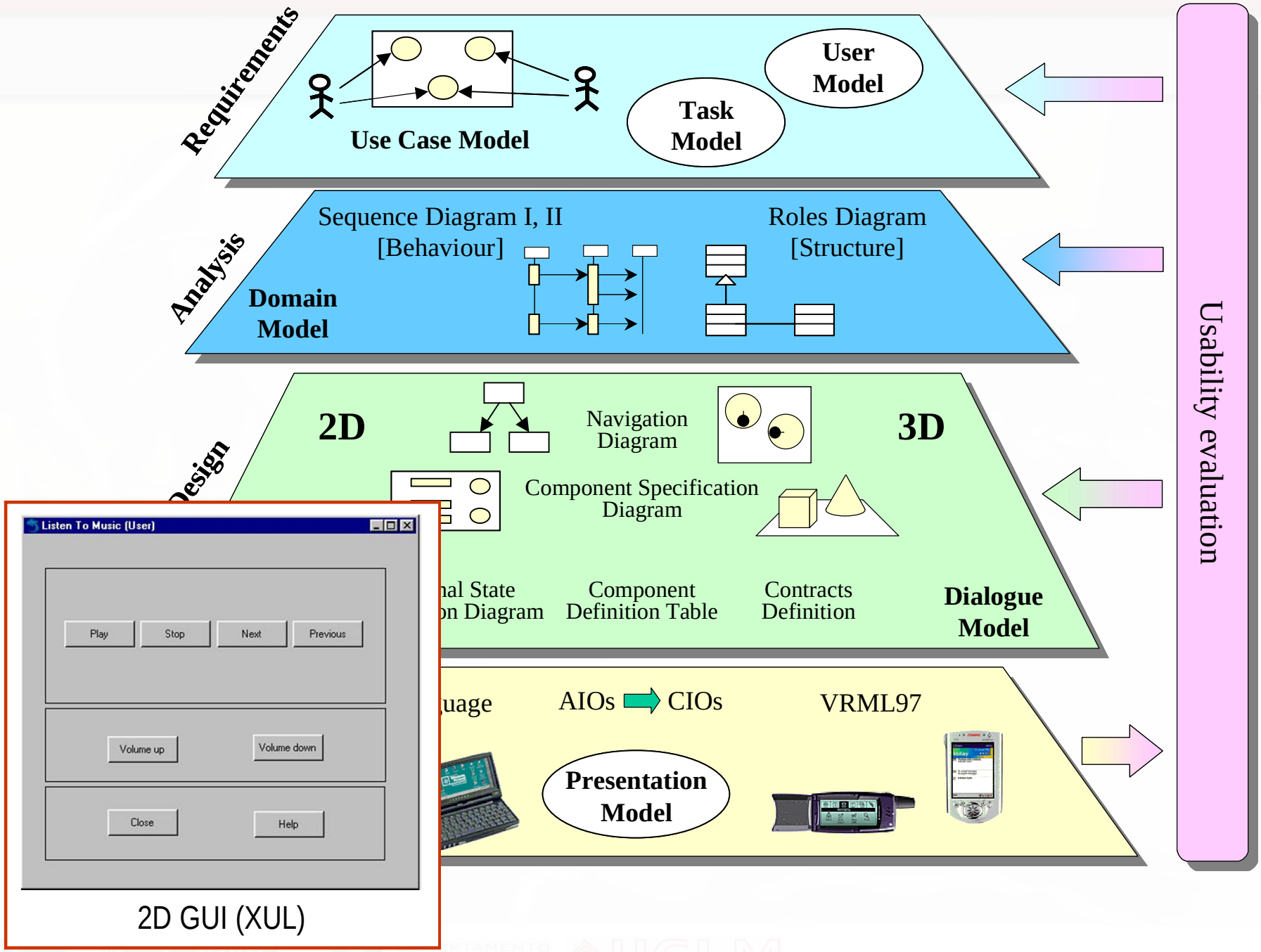
IDEAS CASE [Lozano, 01]

Methods for the development of 3DUIs

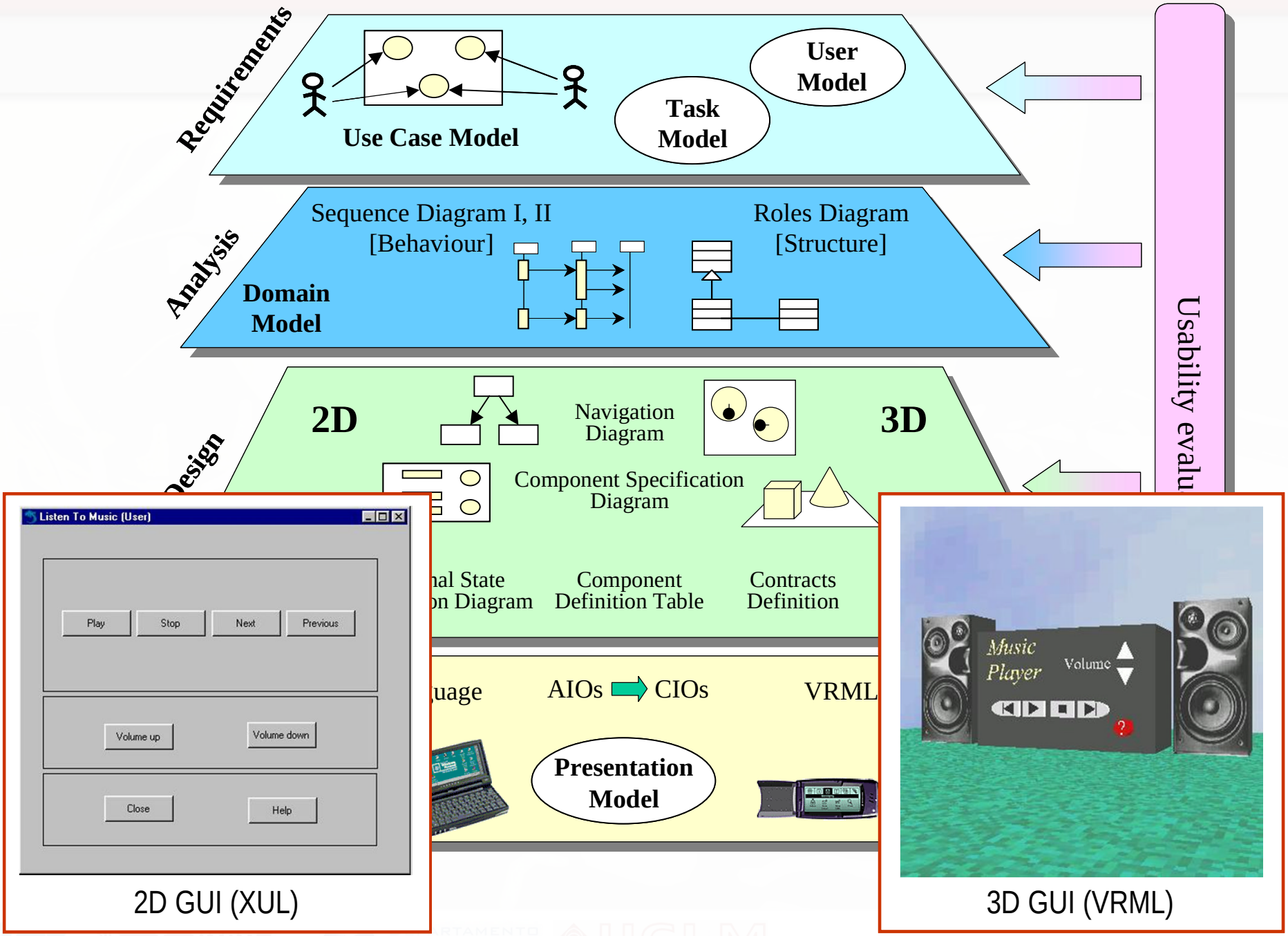
- IDEAS-3D
  - Based on IDEAS, and with the aim of bringing the advantages of this method to the development of 3DUIs, then IDEAS-3D was proposed, **bridging the gap between 2D and 3D**.




Requirements
Use Case Model
Task Model
User Model
Analysis
Sequence Diagram I, II
[Behaviour]
Roles Diagram
[Structure]
Domain Model
Design
2D
Navigation Diagram
3D
Component Specification Diagram
Internal State Transition Diagram
Component Definition Table
Contracts Definition
Dialogue Model
Presentation
XUL Language
AIOs
CIOs
VRML97
Presentation Model
Usability evaluation


Requirements
Use Case Model
Task Model
User Model
Analysis
Sequence Diagram I, II
[Behaviour]
Roles Diagram
[Structure]
Domain Model
Design
2D
3D
Navigation Diagram
Component Specification Diagram
Component Definition Table
Contracts Definition
Dialogue Model
AIOs
CIOs
VRML97
Presentation Model
Usability evaluation
Listen To Music (User)
Play
Stop
Next
Previous
Volume up
Volume down
Close
Help
2D GUI (XUL)

A Structured Approach to the Development of 3D User Interfaces



Methods for the development of 3DUIs

- IDEAS-3D
  - The experience mainly showed the difficulty that represent to extend a method such as IDEAS to cover the whole design space and applications of 3DUIs.
  - **Doing so requires greater changes in the models and in the process that guides the development.**

Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- **The TRES-D methodology**
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

The TRES-D methodology

- ThREe dimensional uSer interface Development.
- The **main purpose** is to put together new solutions with the best of the studied proposals in a creation process that can be adapted to the variable complexity of 3DUIs.

The TRES-D methodology

- **Th**ree dimensional u**S**er interface **D**evelopment.
- The main purpose is to put together new solutions with the best of the studied proposals in a creation process that can be adapted to the variable complexity of 3DUIs.
- **Objectives:**
  - Give an ordered set of steps for the development of 3DUIs
  - 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

The TRES-D methodology

- **Th**ree dimensional u**S**er interface **D**evelopment.
- The main purpose is to put together new solutions with the best of the studied proposals in a creation process that can be adapted to the variable complexity of 3DUIs.
- **Objectives:**
  - **Give an ordered set of steps for the development of 3DUIs**
  - 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

The TRES-D methodology

- **Th**ree dimensional u**S**er interface **D**evelopment.
- The main purpose is to put together new solutions with the best of the studied proposals in a creation process that can be adapted to the variable complexity of 3DUIs.
- **Objectives:**
  - Give an ordered set of steps for the development of 3DUIs
  - **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

The TRES-D methodology

- **Th**ree dimensional u**S**er interface **D**evelopment.
- The main purpose is to put together new solutions with the best of the studied proposals in a creation process that can be adapted to the variable complexity of 3DUIs.
- **Objectives:**
  - Give an ordered set of steps for the development of 3DUIs
  - 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**

The TRES-D methodology

- **Th**ree dimensional u**S**er interface **D**evelopment.
- The main purpose is to put together new solutions with the best of the studied proposals in a creation process that can be adapted to the variable complexity of 3DUIs.
- Objectives:
  - Give an ordered set of steps for the development of 3DUIs
  - 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
- Its **distinctive characteristic** is to combine **two different approaches** for the development of UIs, one oriented to tasks and the other one to content, **in one single structured process**.

Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - **Meta-models for a new methodology**
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

Meta-models for a new methodology

- The **purpose** of these meta-models is to fulfil one of the objectives of the TRES-D framework, that is, to offer a language to developers that allow them to…
  - Build their own mental model of 3DUIs
  - Model the different elements that comprise them
  - Communicate those models to other people
  - And relate the concepts and their use within the TRES-D proposal

Meta-models for a new methodology

- The **purpose** of these meta-models is to fulfil one of the objectives of the TRES-D framework, that is, to offer a language to developers that allow them to…
  - **Build their own mental model of 3DUIs**
  - Model the different elements that comprise them
  - Communicate those models to other people
  - And relate the concepts and their use within the TRES-D proposal

Meta-models for a new methodology

- The **purpose** of these meta-models is to fulfil one of the objectives of the TRES-D framework, that is, to offer a language to developers that allow them to…
  - Build their own mental model of 3DUIs
  - **Model the different elements that comprise them**
  - Communicate those models to other people
  - And relate the concepts and their use within the TRES-D proposal

Meta-models for a new methodology

- The **purpose** of these meta-models is to fulfil one of the objectives of the TRES-D framework, that is, to offer a language to developers that allow them to…
  - Build their own mental model of 3DUIs
  - Model the different elements that comprise them
  - **Communicate those models to other people**
  - And relate the concepts and their use within the TRES-D proposal

Meta-models for a new methodology

- The **purpose** of these meta-models is to fulfil one of the objectives of the TRES-D framework, that is, to offer a language to developers that allow them to…
  - Build their own mental model of 3DUIs
  - Model the different elements that comprise them
  - Communicate those models to other people
  - **And relate the concepts and their use within the TRES-D proposal**

Meta-models for a new methodology

- Then, starting from the four elements found in the preliminary study, which were *3D space*, *objects*, *behaviour* and *interaction*, **three meta-models** are now proposed:
  - Object meta-model
  - Interaction meta-model
  - Space meta-model

Meta-models for a new methodology

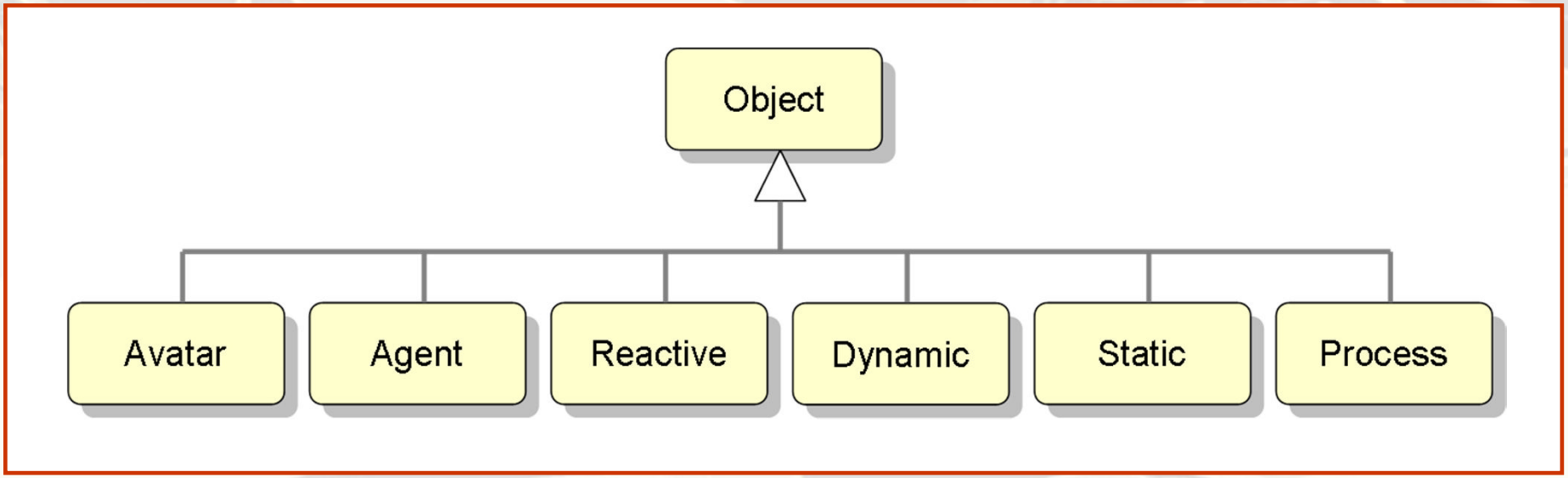
- Then, starting from the four elements found in the preliminary study, which were *3D space*, *objects*, *behaviour* and *interaction*, three meta-models are now proposed:
  - Object meta-model
  - Interaction meta-model
  - Space meta-model
- Their **vocabulary** is based on the literature from the field, but with the necessary changes in terms and meanings to solve ambiguities and contradictions found in such literature.

Meta-models for a new methodology

- Object meta-model
  - This first meta-model addresses two issues:
    - First, to help developers in their tasks of working out the requirements for each object
    - And second, once the objects are classified, to show developers what to specify and how

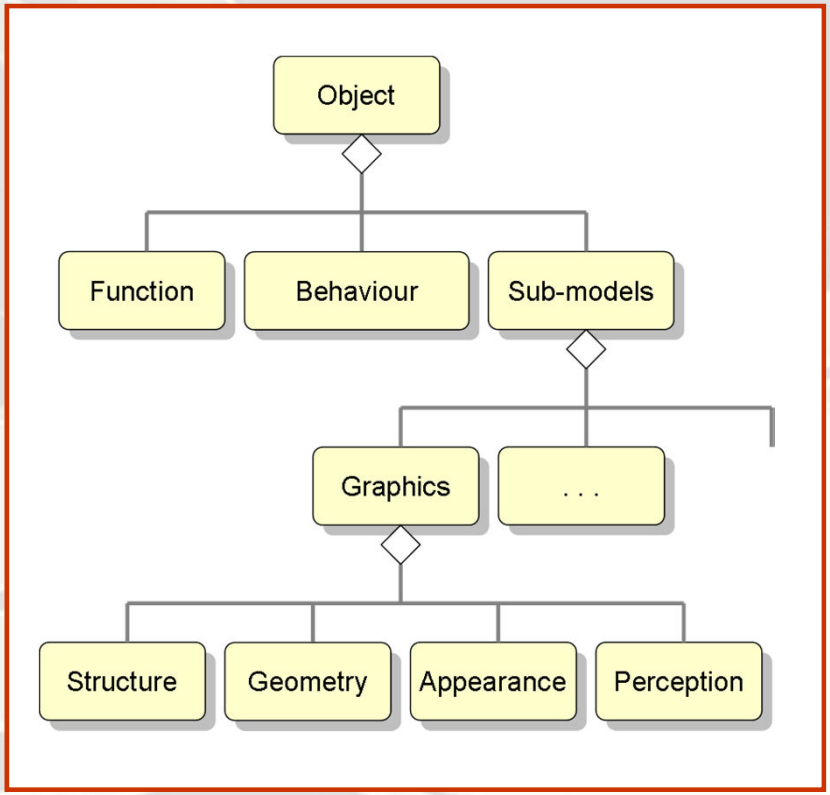
Meta-models for a new methodology

- Object meta-model (cont'd)
  - With the purpose to help developers when working out the requirements for each object, a new **classification of objects** puts together in a simple hierarchy many different classes identified by other authors.



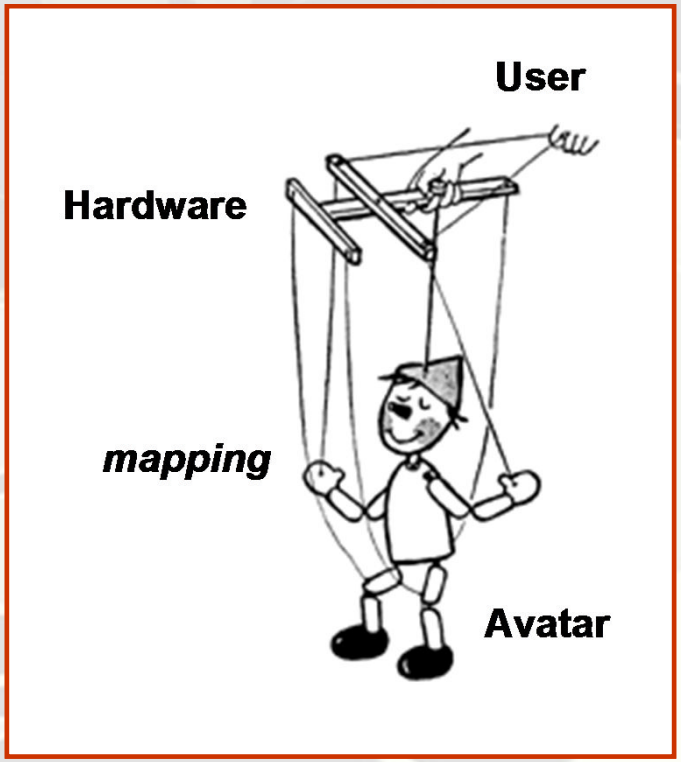
Meta-models for a new methodology

- Object meta-model (cont'd)
  - Once the objects are classified, the developer will also need to know what to specify and how.
  - After a revision of the different models proposed by previous authors, it was decided a composition of *function*, *behaviour* and a set of *sub-models*.
  - As for the graphics sub-model, four elements are identified: *structure*, *geometry*, *appearance* and *perception*.



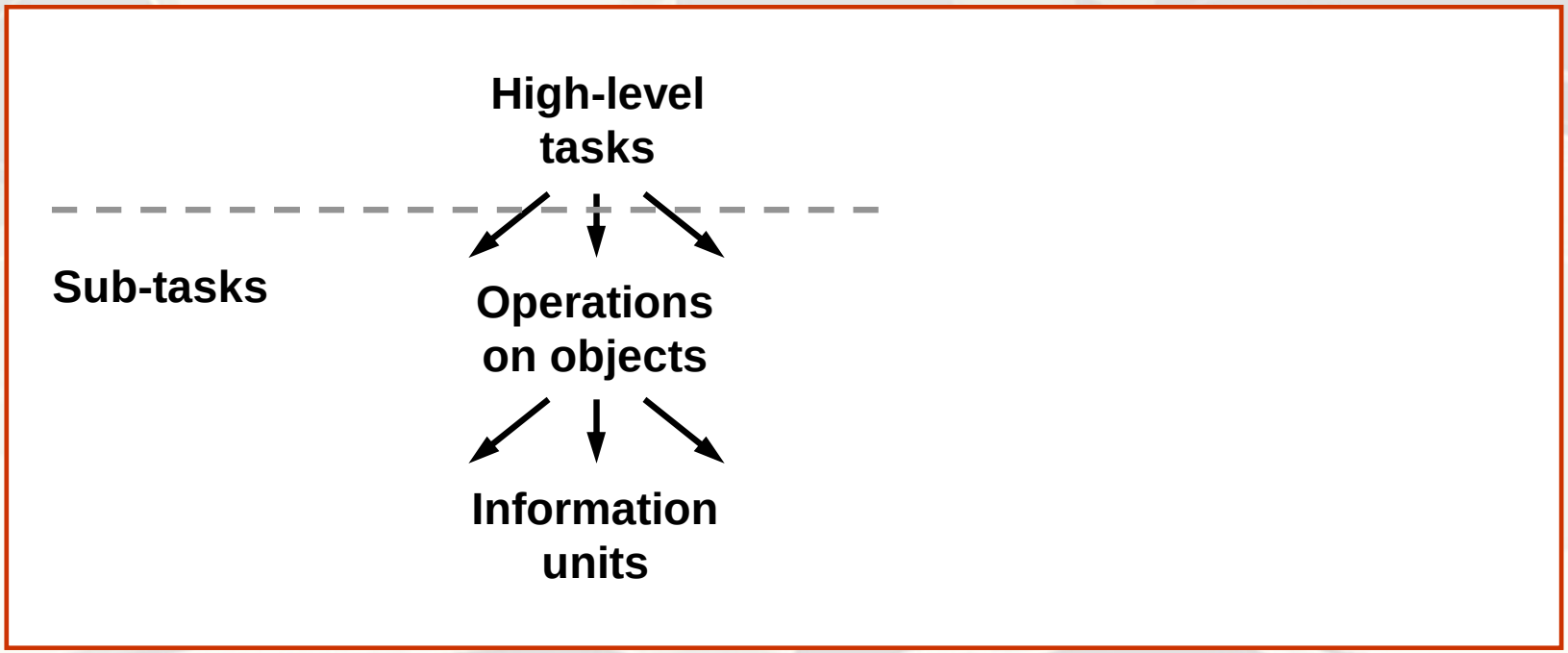
Meta-models for a new methodology

- Interaction meta-model
  - The structure of this second model starts from the explanations given in [Foley, 96], and it is based on the following concepts:
    - *dialogue*, *tasks*, *operations*, *interaction techniques*, *actions*, *controls* and *physical devices*



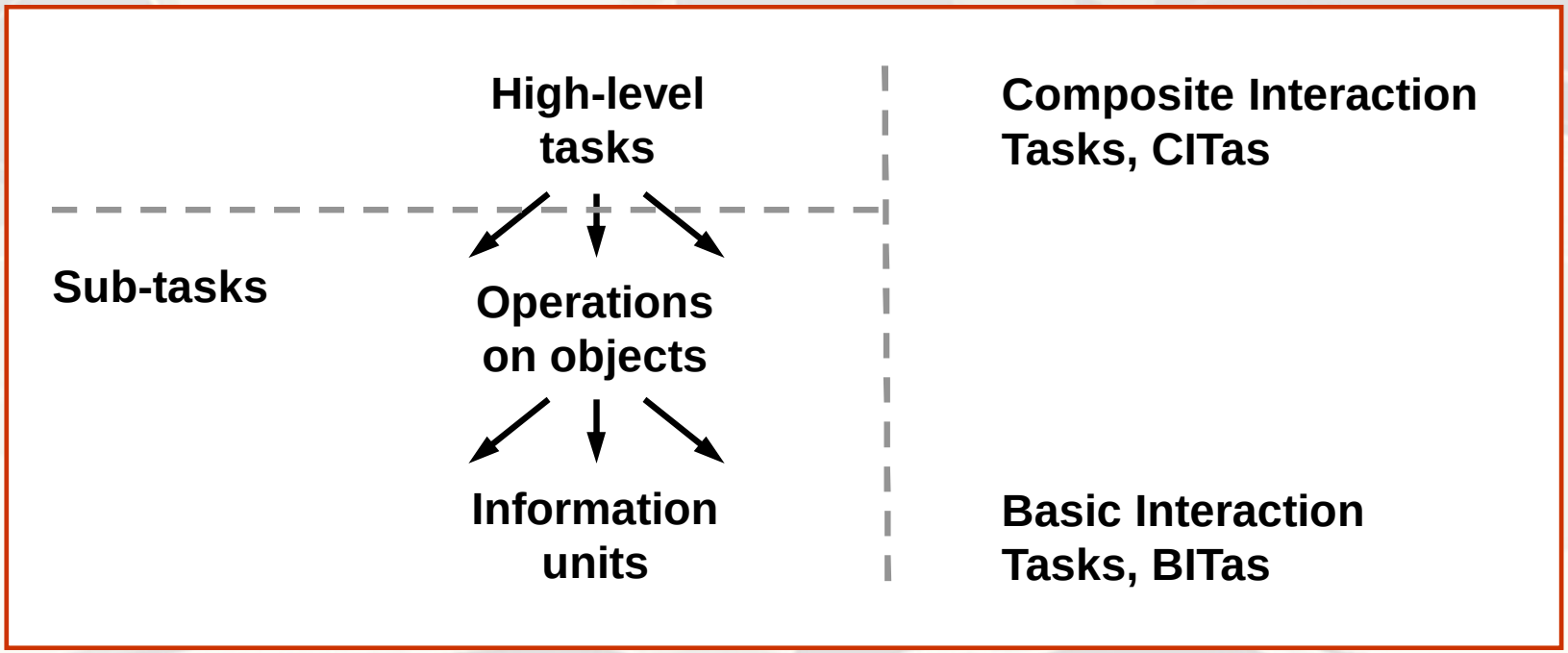
Meta-models for a new methodology

- Interaction meta-model: Tasks and operations
  - A distinction is made 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.



Meta-models for a new methodology

- Interaction meta-model: Tasks and operations
  - The leaves of this hierarchy are the *Basic Interaction Tasks* or **BITas**, the other nodes are *Composite Interaction Tasks* or **CITas**.

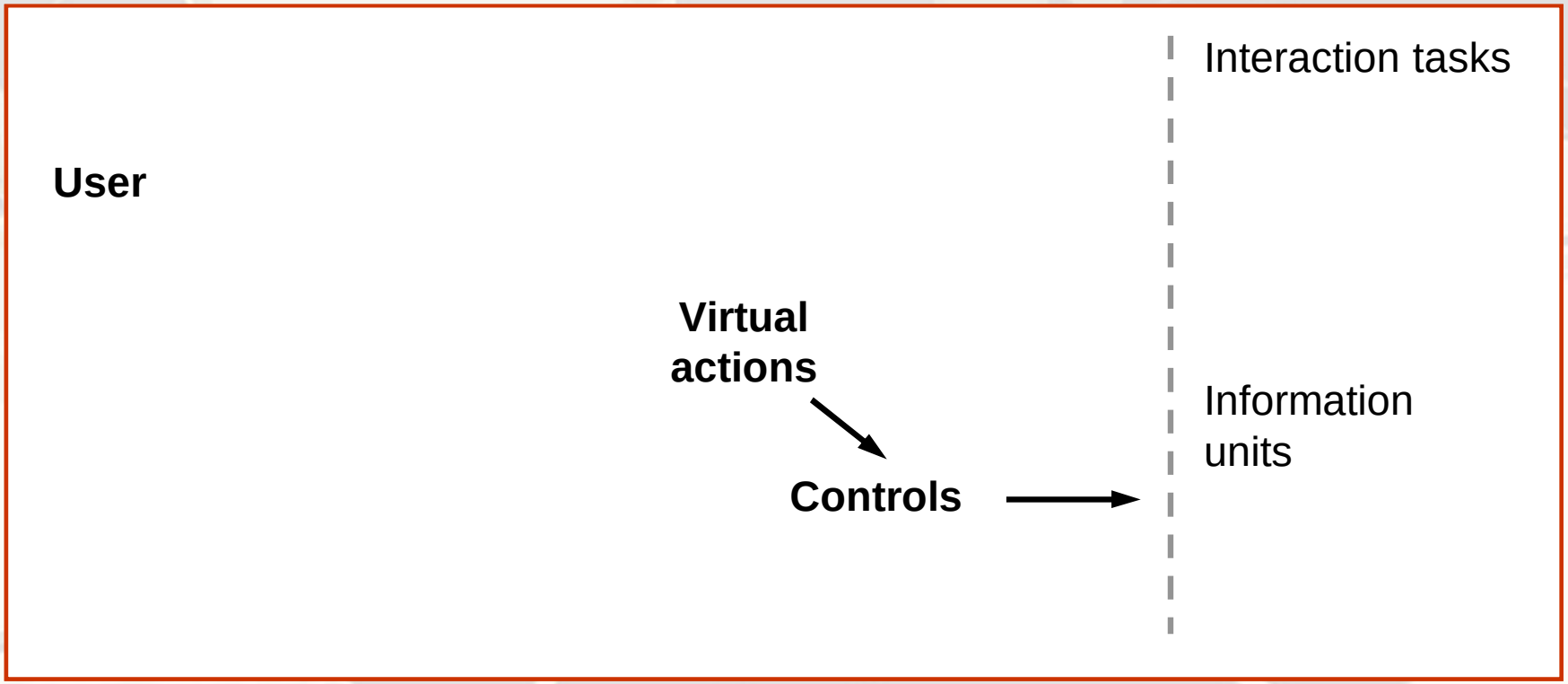


Meta-models for a new methodology

- Interaction meta-model: Interaxn. techniques and controls
  - A distinction is also made between interaction techniques and controls. A control, e.g. a widget, is no longer considered as an interaction technique here, but only a part of it.

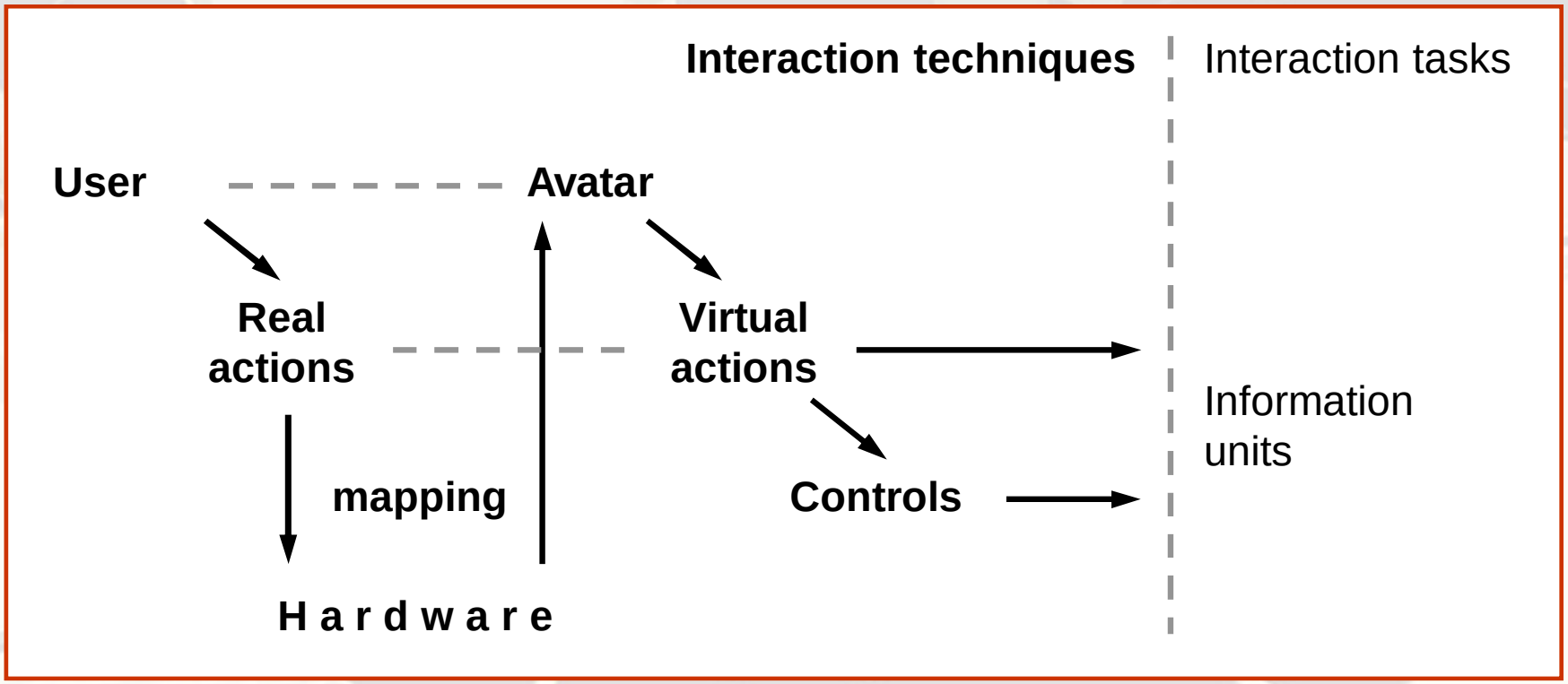
Meta-models for a new methodology

- Interaction meta-model: Interaxn. techniques and controls
  - A **control** is the object of user **virtual actions**, and generate the units of information that represents an interaction task. Controls can be combined to form **components**.



Meta-models for a new methodology

- Interaction meta-model: Interaxn. techniques and controls
  - An **interaction technique** (**ITe**) translates user **physical actions** to the virtual world as virtual actions, as a way to materialize an interaction task or to operate a control.

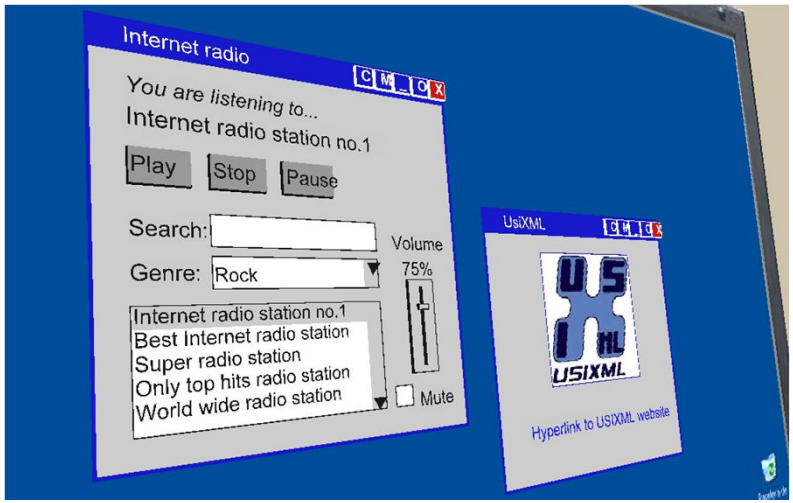


Meta-models for a new methodology

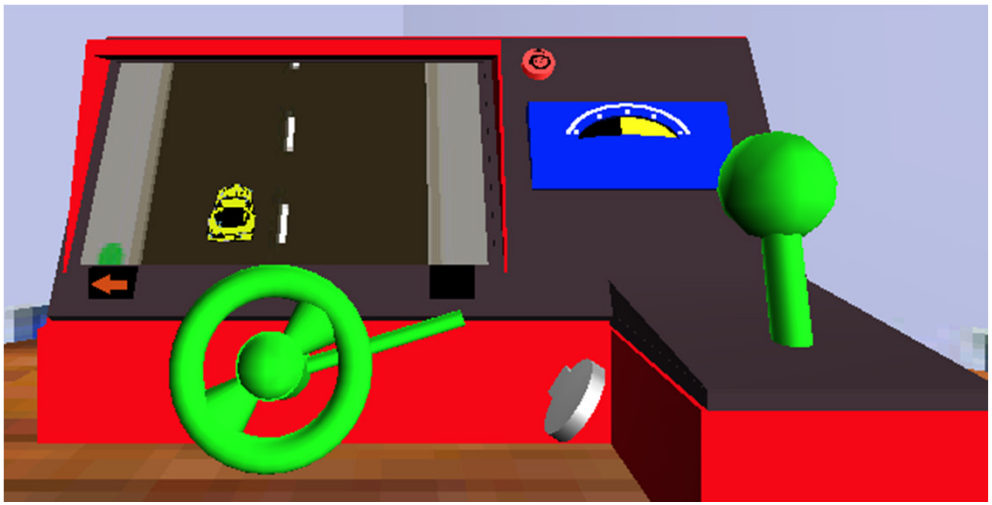
- Interaction meta-model: Interaxn. techniques and controls
  - In parallel to interaction tasks, it is also proposed the terms *Basic Interaction Techniques* or **BITes**, and *Composite Interaction Techniques* or **CITes**.
  - With this acronyms, the intention is to **avoid the usage of IT letters alone**, so much used in the literature, but leading to confusion as they can refer to both interaction tasks or techniques.

Meta-models for a new methodology

- Interaction meta-model: Interaxn. techniques and controls
  - With these definitions, the term **widget** simply refers to a class of controls, in particular to those found in desktop 2D GUIs.
  - A **3D widget** refers to the 3D representation of those elements.
  - Other examples of controls are the interactive elements of the virtual objects. For them, a new term is introduced, **obget**.



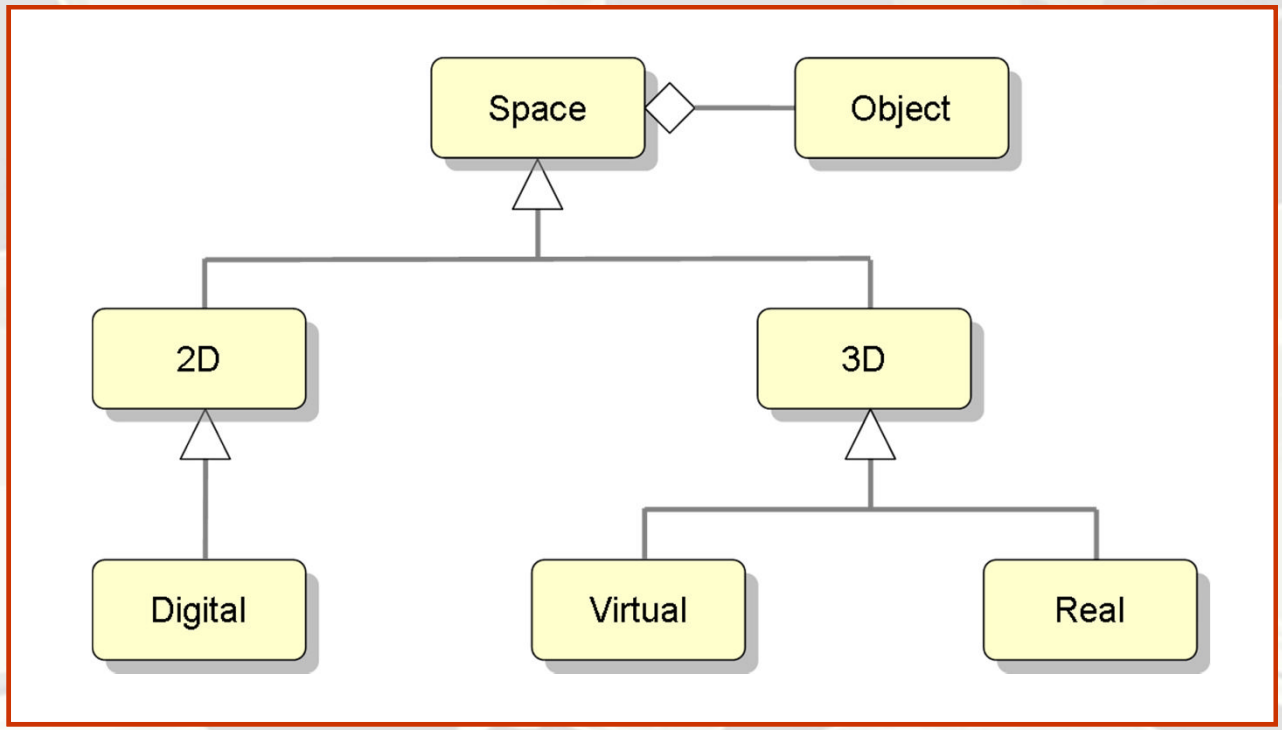
3D widgets



Obgets

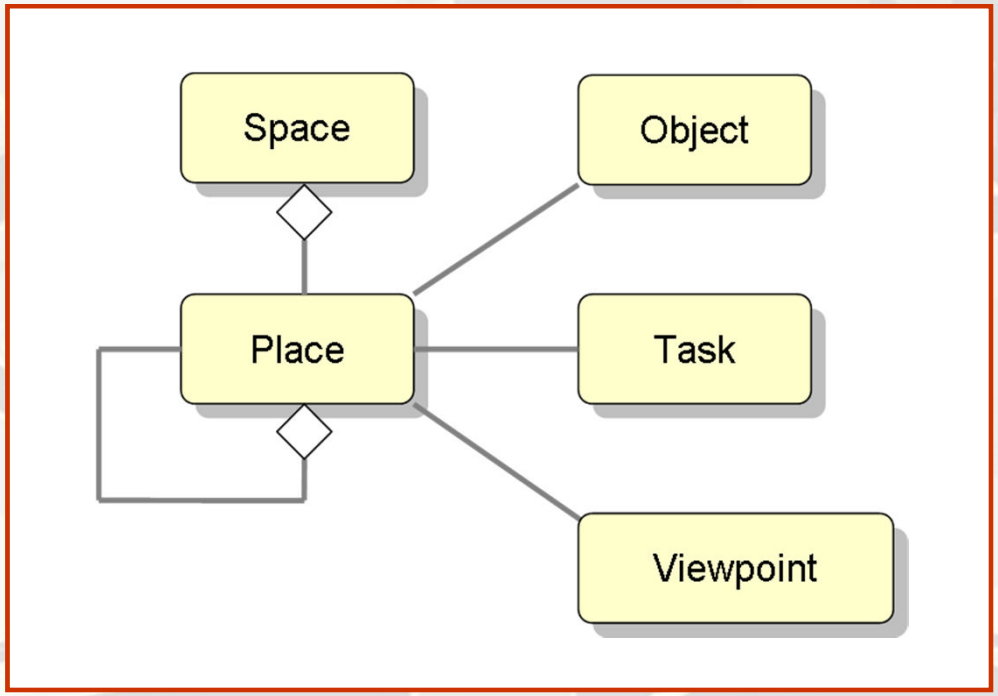
Meta-models for a new methodology

- Space meta-model
  - Based on the digital-virtual-real continuum, this meta-model distinguishes between the **2D digital space**, the **3D virtual space** and the **3D real space**.



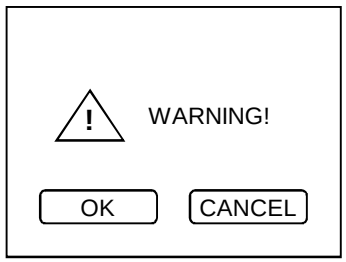
Meta-models for a new methodology

- Space meta-model (cont'd)
  - In the last two spaces, it is highlighted the concepts of **place**, zone or locus, as well as the **point of view** or point of observation, as important elements in the ordering of such spaces.

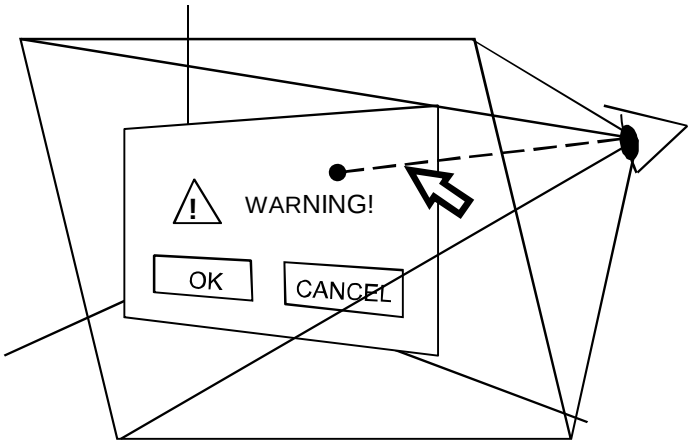


Meta-models for a new methodology

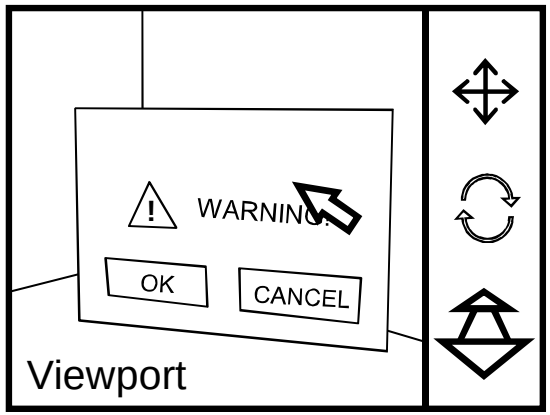
- Space meta-model (cont'd)
  - This model also remarks the relationships that link these spaces, as embedding a virtual world in a 2D interface through a viewport, or immersing a 2D interface in a 3D environment.



2D interface



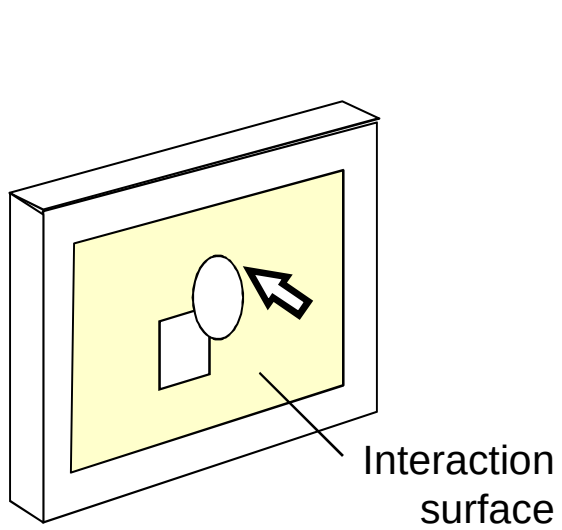
2D interface in a 3D world



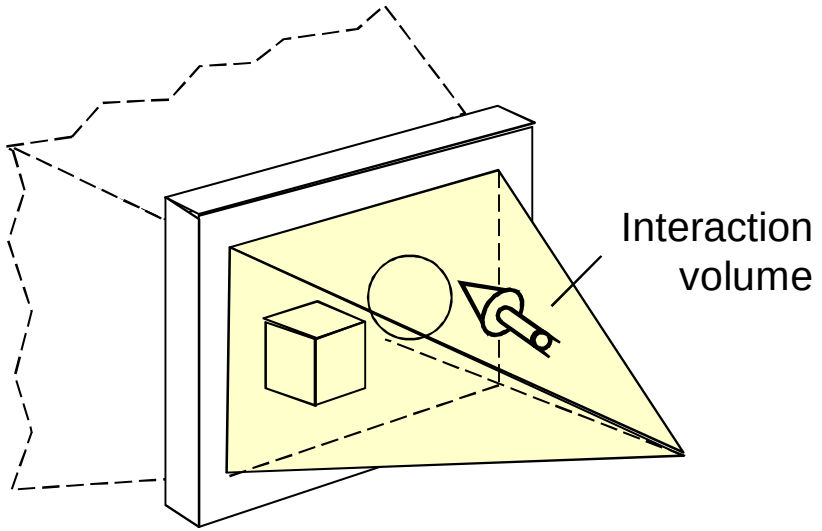
3D world in a 2D interface

Meta-models for a new methodology

- Space meta-model (cont'd)
  - The digital and virtual spaces are linked to the real world by means of physical devices that act as **interaction surfaces** or **interaction volumes**.



2D interface



3D world rendered with stereo 3D graphics

Meta-models for a new methodology

- The three meta-models are also related with each other:
  - The tasks are decomposed in particular actions that are performed on some objects
  - And both tasks and objects are tightly related to space
- **This triangle relationship strongly define the process model of the TRES-D methodology.**

Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - **The process model of the TRES-D methodology**
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

The process model of the TRES-D methodology

- This process is supported on the concepts and the terms provided by the three meta-models presented before, and shapes a framework where different practices and tools are gathered.

The process model of the TRES-D methodology

- **Principles:**
  - 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 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 UI designers and programmers as well as artists and digital content creators
  - Last, to involve not only those professionals, but also the client, domain experts and, of course, the user

The process model of the TRES-D methodology

- **Principles:**
  - **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 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 UI 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 of the TRES-D methodology

- **Principles:**
  - 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 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 UI 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 of the TRES-D methodology

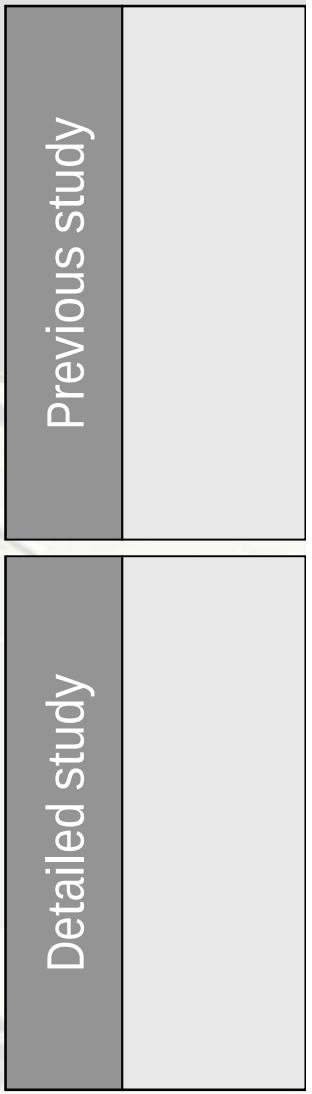
- **Principles:**
  - First, to reduce as much as possible the risks that the developer may face in a field still far from being mature
  - Second, to distinguish between design and implementation, and within the former to differentiate between implementation-independent design and a dependent one
  - **Third, to orient the development to both tasks and interaction as well as objects and content, that is, to both UI 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 of the TRES-D methodology

- **Principles:**
  - 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 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 UI 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**

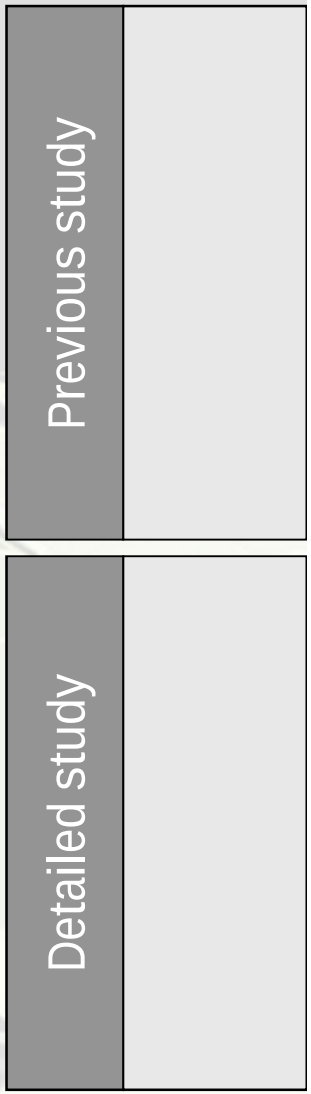
The process model of the TRES-D methodology

- The process is divided in two main phases:
  - The first phase, or **previous study**, is devoted to understand the problem and then propose a solution.
  - The second phase, or **detailed study**, is aimed at developing such solution up to its final deployment and further maintenance.
  - Between both phases mediates and agreement with the client, who has to approve the proposed solution.



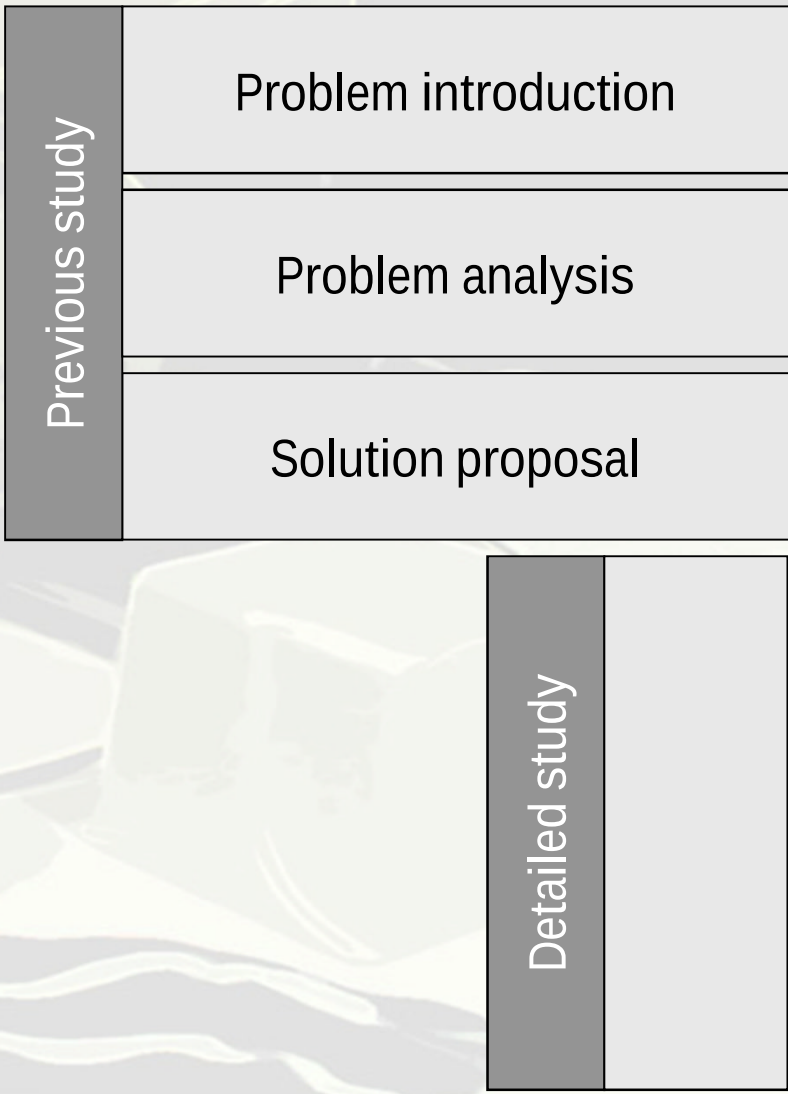
The process model of the TRES-D methodology

- Each phase is also composed of three stages:



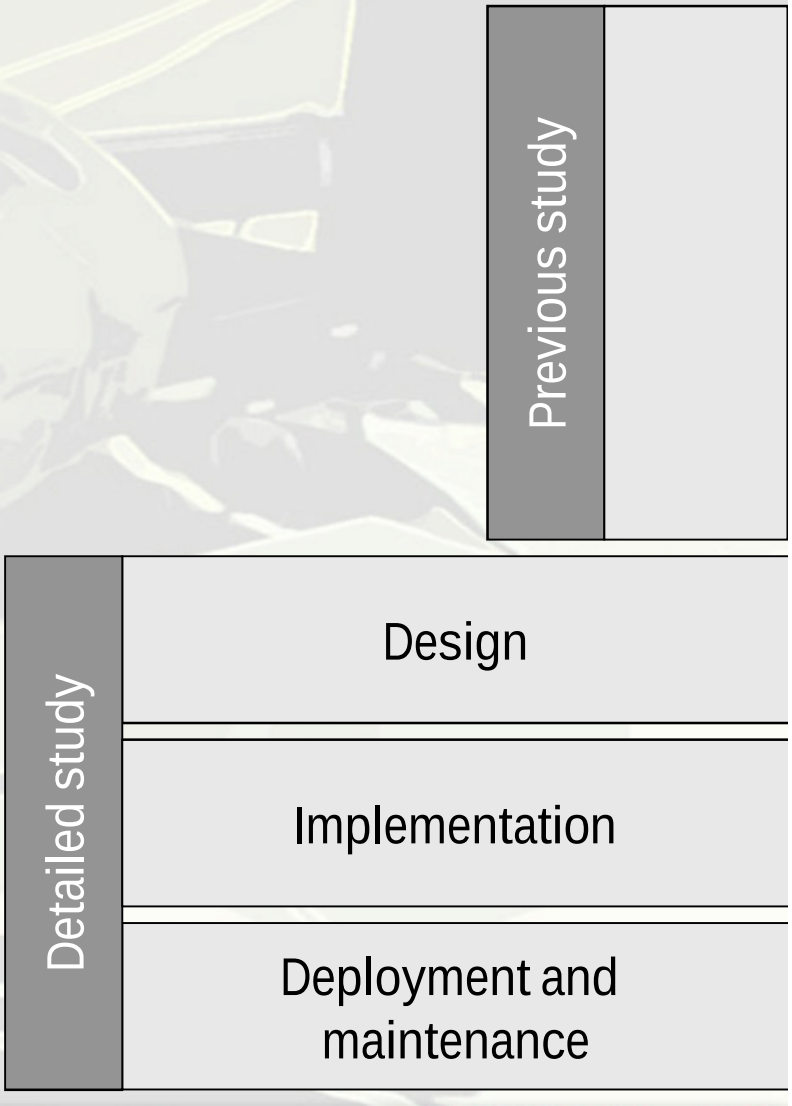
The process model of the TRES-D methodology

- Each phase is also composed …
  - The **previous study** consists of *problem introduction*, *problem analysis* and *solution proposal*.



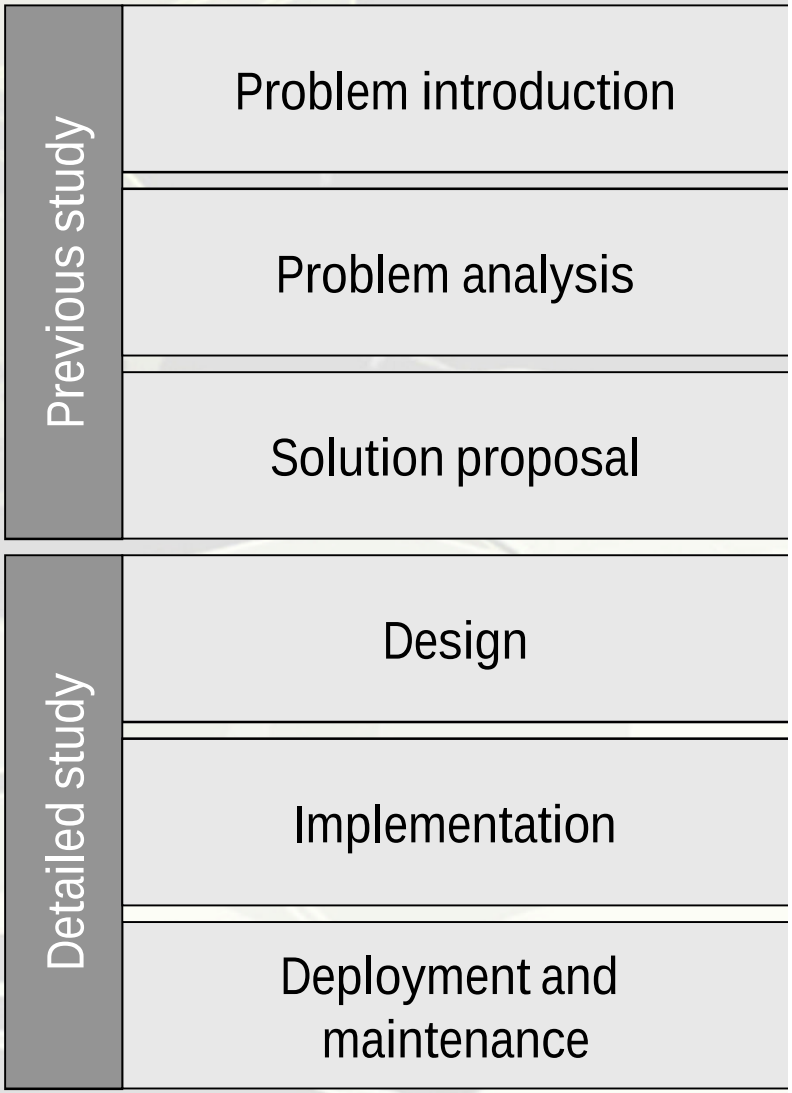
The process model of the TRES-D methodology

- Each phase is also composed …
  - The **detailed study** is composed of *design*, *implementation*, and *deployment and maintenance*.



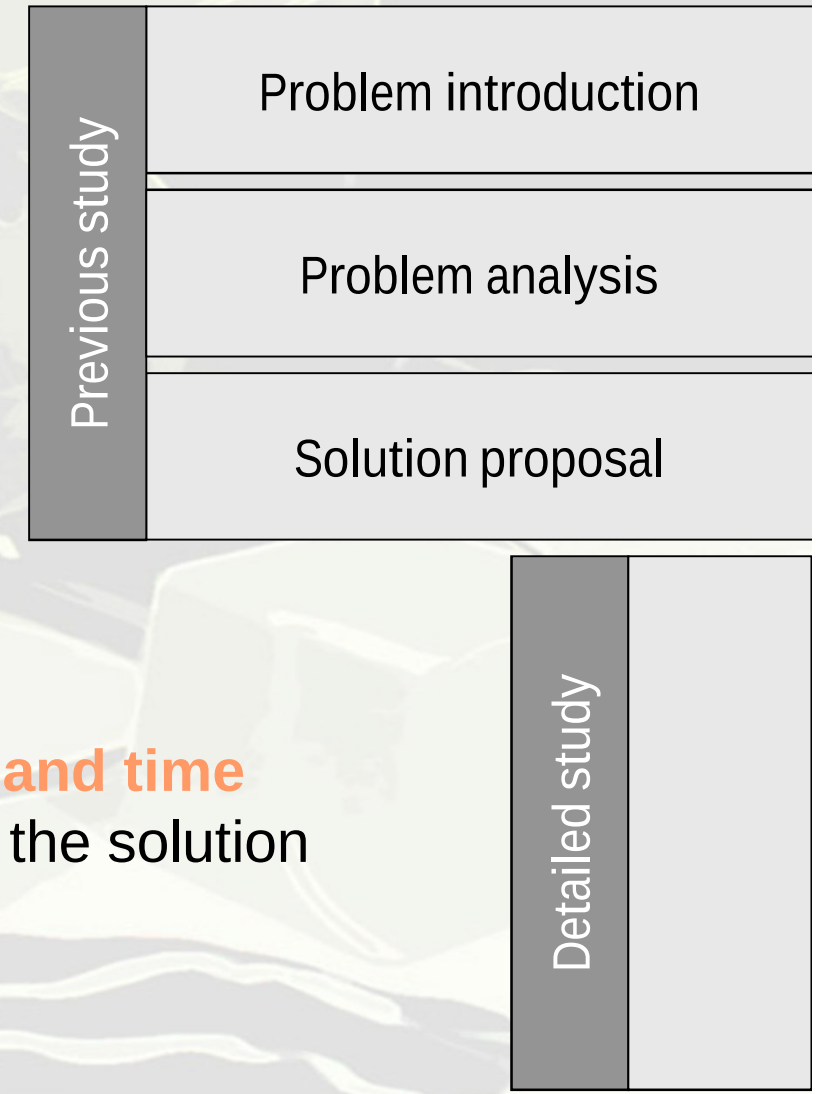
The process model of the TRES-D methodology

- Each phase is also composed …
  - 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 face.



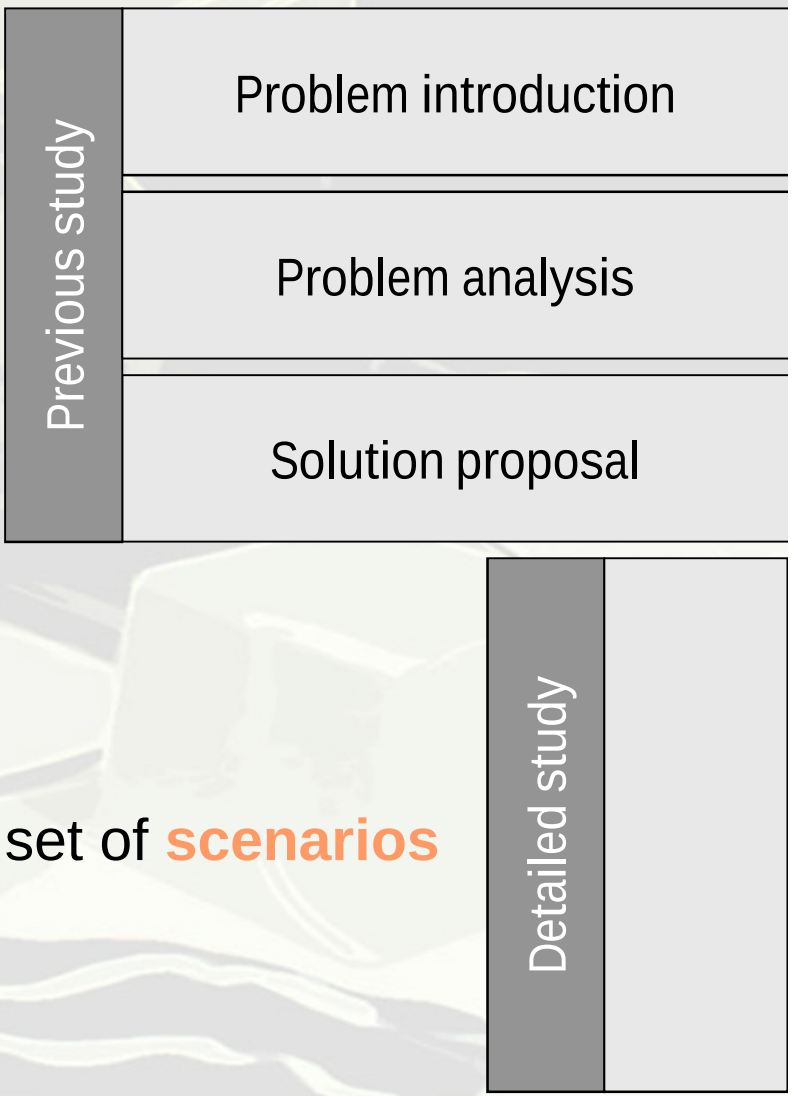
The process model of the TRES-D methodology

- Initial study: Problem introduction
  - In this stage, the **client** meets the architect to set up the **initial requirements** of the application.
  - The **architect**, as an expert in UI development, makes use of their experience to discuss and clarify at this first moment some of the objectives, while the rest of them will need to be studied in depth.
  - That role also prepares a **first budget and time planning** for the problem analysis and the solution proposal.



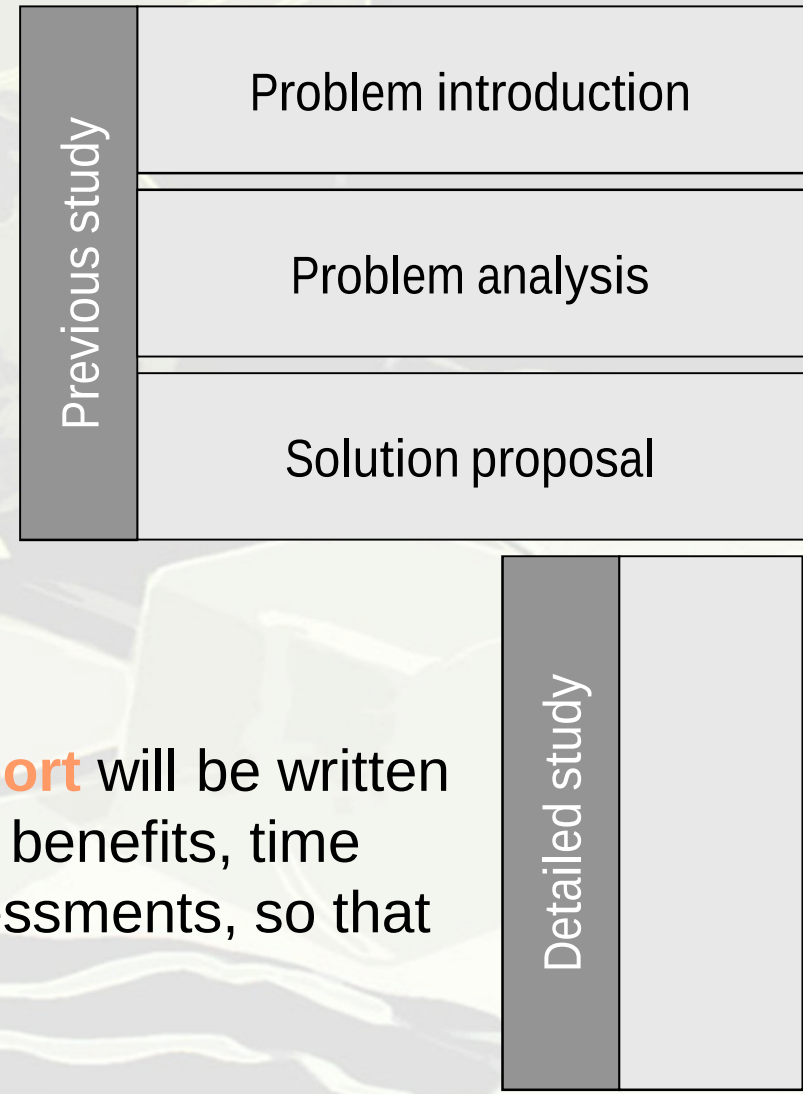
The process model of the TRES-D methodology

- Initial study: Problem analysis
  - Before carrying out the design it is necessary to understand what is meant to be accomplished, task that is performed by **analysts**.
  - At the end of this stage we want to know what **kind of application** is to be developed, which are the characteristics of the people that will use it (**user profiles**), and what tasks will be carried out by them.
  - A **task and domain analysis** shapes a set of **scenarios** and highlights spatial relationships.



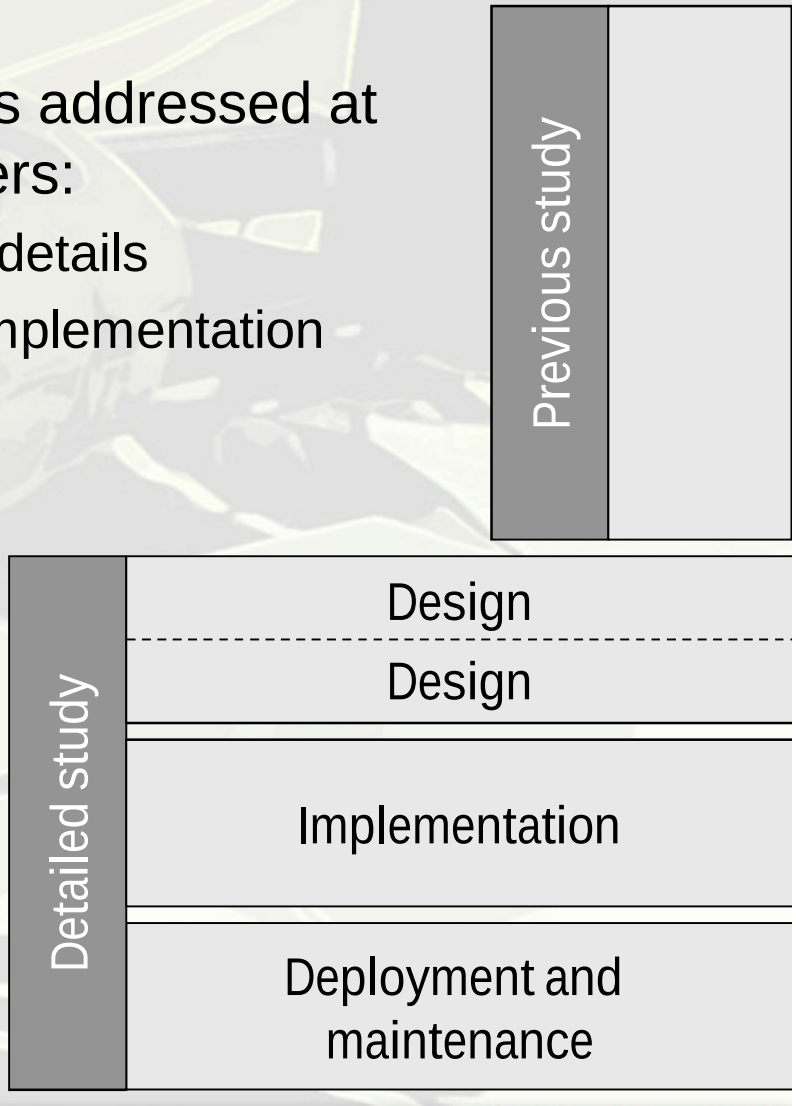
The process model of the TRES-D methodology

- Initial study: Solution proposal
  - A group of **designers** works on different solutions for the application. Artists, digital content creators, programmers, domain experts and users participate too.
  - The main aim of this stage is to **envision a solution** that solves the problem, satisfies the user and, at the same time, is feasible.
  - At the end of the stage, a complete **report** will be written describing the solution and detailing its benefits, time and cost of development, and risk assessments, so that a proper decision can be made.



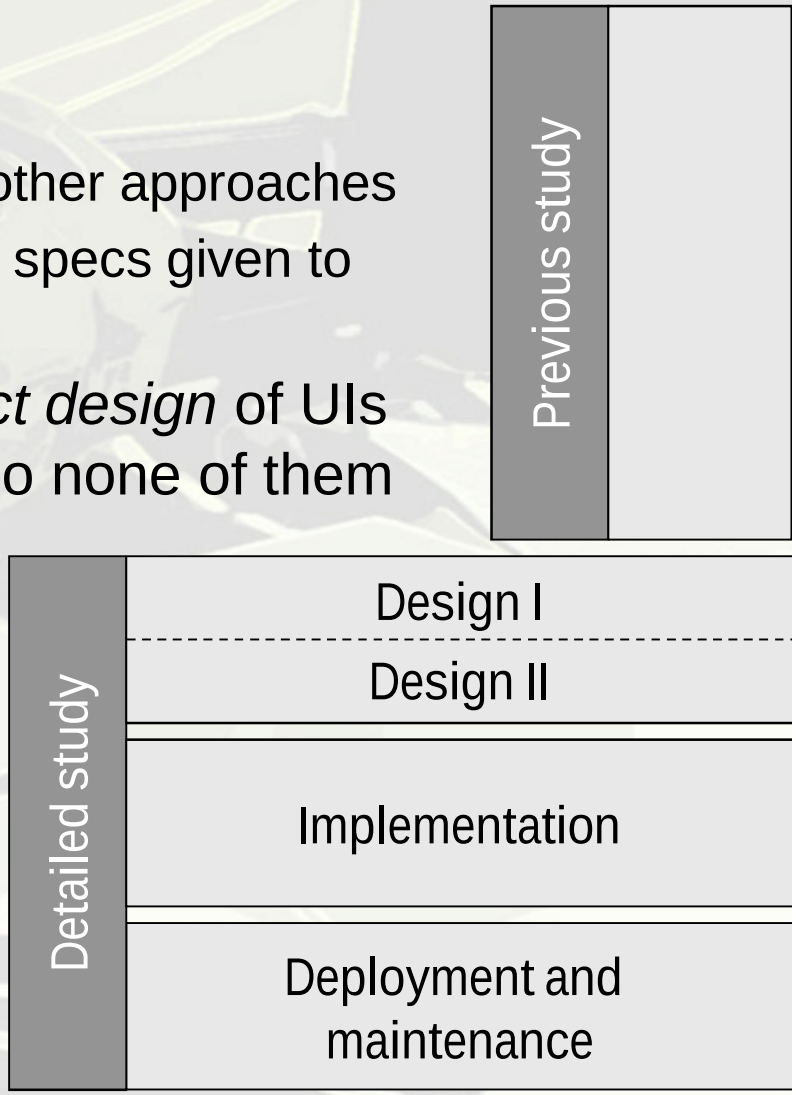
The process model of the TRES-D methodology

- Detailed study: Design
  - Once at the second phase, the design is addressed at two different levels, two abstraction layers:
    - One detached from the implementation details
    - And another one tightly related to that implementation



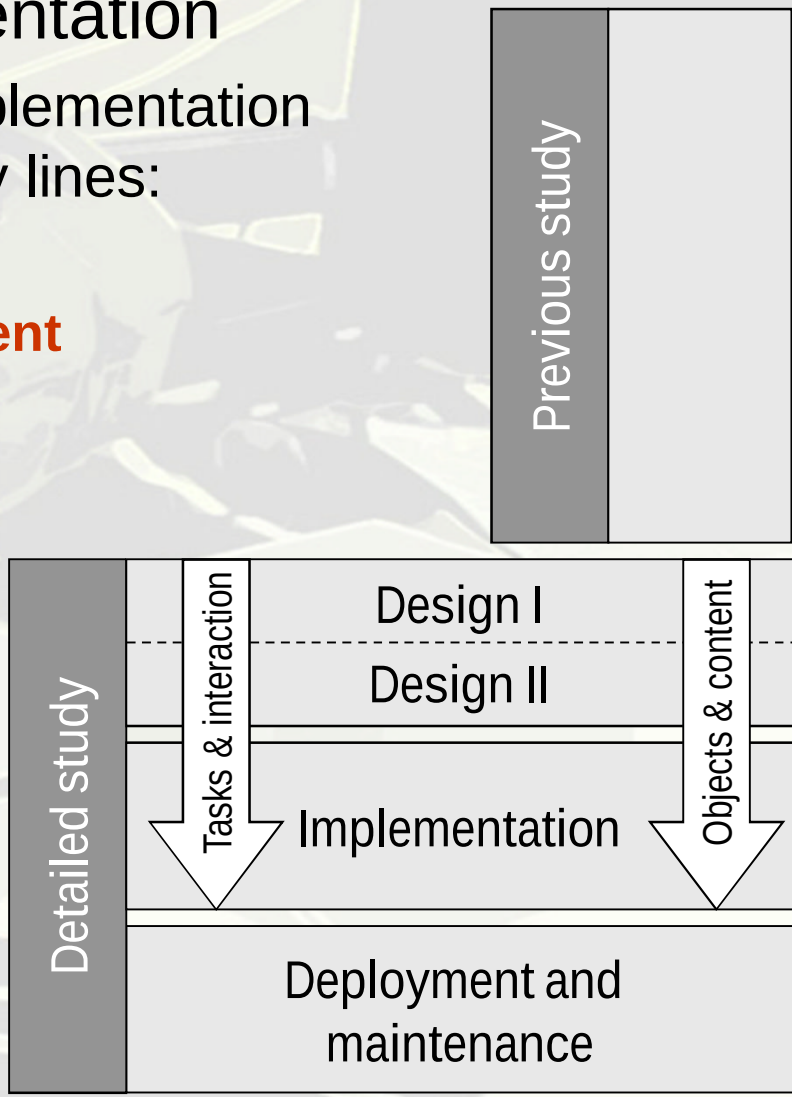
The process model of the TRES-D methodology

- Detailed study: Design (cont'd)
  - This horizontal division separates:
    - Interaction tasks from techniques as in other approaches
    - But also artists' impressions from object specs given to content creators
  - Thus, the higher level mixes the *abstract design* of UIs with the *concept design* of artists, and so none of them is used as a name.
  - Instead, **Design I** is used.
  - **Design II** just follows the previous.
  - A decision on which software and hardware will support the solution divides both design levels.



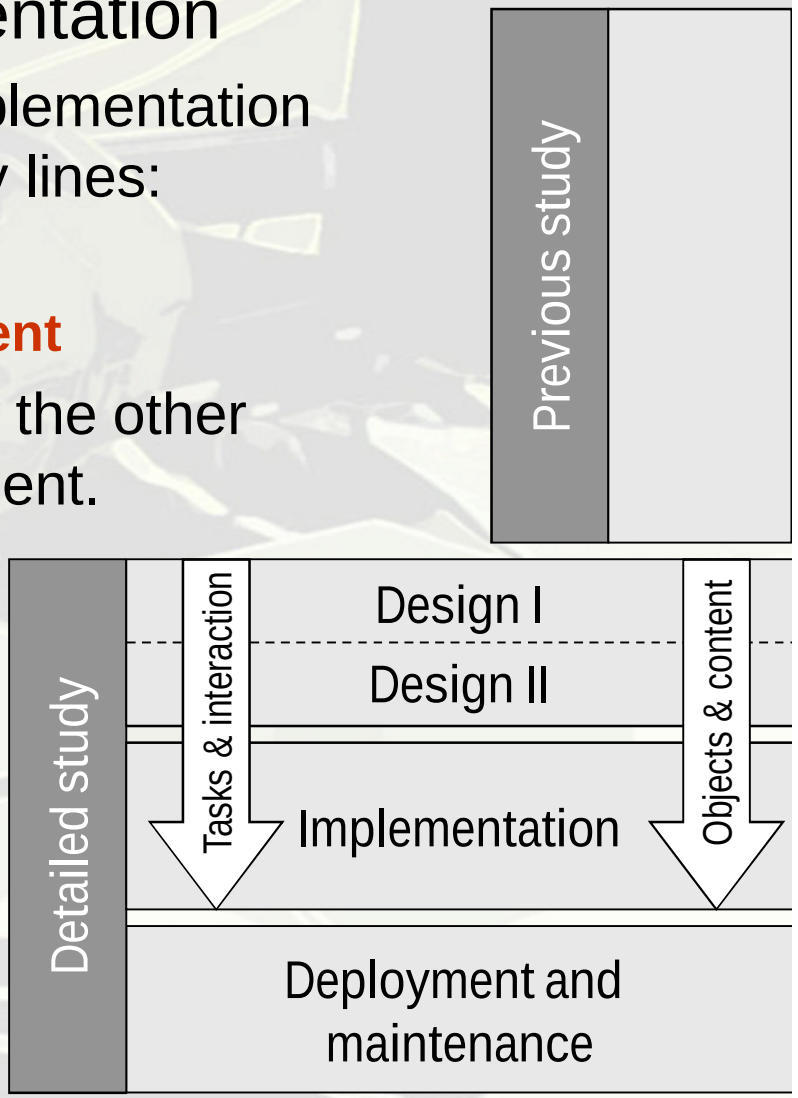
The process model of the TRES-D methodology

- Detailed study: Design and implementation
  - Both design I and design II, and the implementation stage, are divided in two parallel activity lines:
    - One oriented to **tasks and interaction**
    - And the other one to **objects and content**



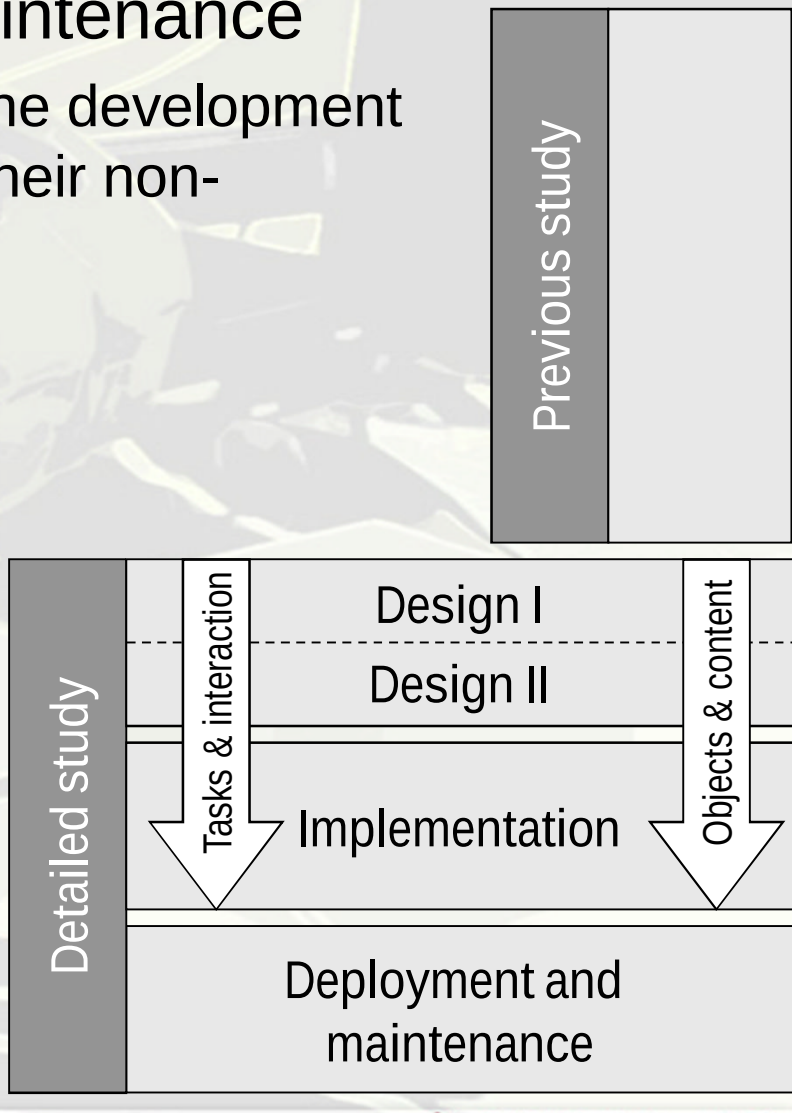
The process model of the TRES-D methodology

- Detailed study: Design and implementation
  - Both design I and design II, and the implementation stage, are divided in two parallel activity lines:
    - One oriented to **tasks and interaction**
    - And the other one to **objects and content**
  - The weight is translated towards one or the other depending on each particular development.
  - In any case, both lines cross at several points during the process development, due to their relationship of interaction and objects with **space**, thus following the three meta-models.



The process model of the TRES-D methodology

- Detailed study: Deployment and maintenance
  - This last stage can not be forgotten in the development of 3DUIs, quite often characterized by their non-conventional hardware.



Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - **Exemplar advise tools**
  - The VUIToolkit library
  - Case studies
- Conclusions and further work

Exemplar advise tools

- 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:
  - Some of these tools advise and guide the developers in their design decisions
  - Others help them to document their ideas
  - Finally, there are others that make possible to turn those designs into real products

Exemplar advise tools

- 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:
  - **Some of these tools advise and guide the developers in their design decisions**
  - Others help them to document their ideas
  - Finally, there are others that make possible to turn those designs into real products

Exemplar advise tools

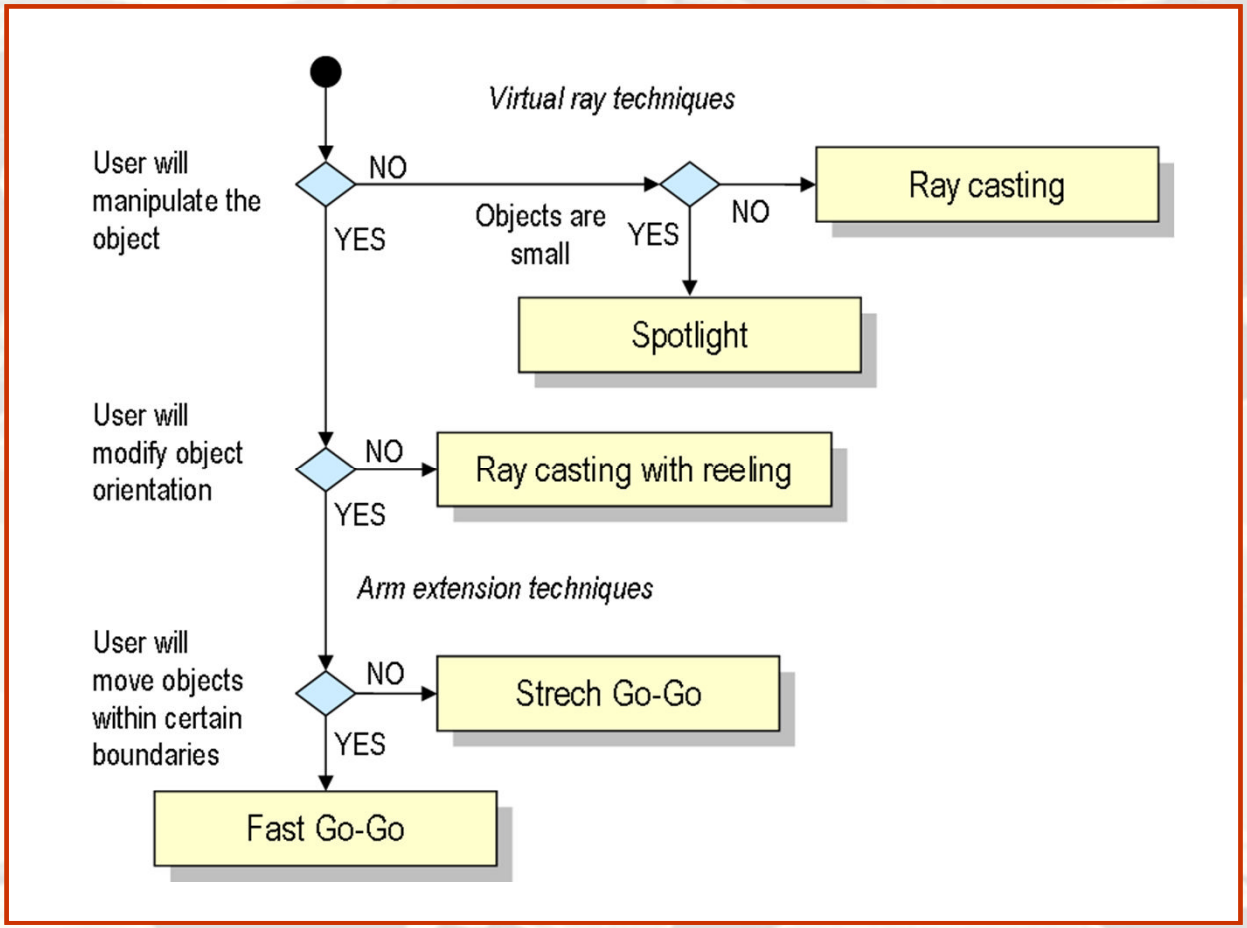
- Regarding the first ones, the main **motivation** is to **record the lessons learned** from pasts experiences in a form that allows an easy application of them in future projects.
- The TRES-D framework incorporates proposals from other authors, but also introduces two new **advise tools** meant for selecting interaction techniques, in particular in the context of immersive VEs:
  - Selection and manipulation techniques
  - Symbolic (text) input techniques

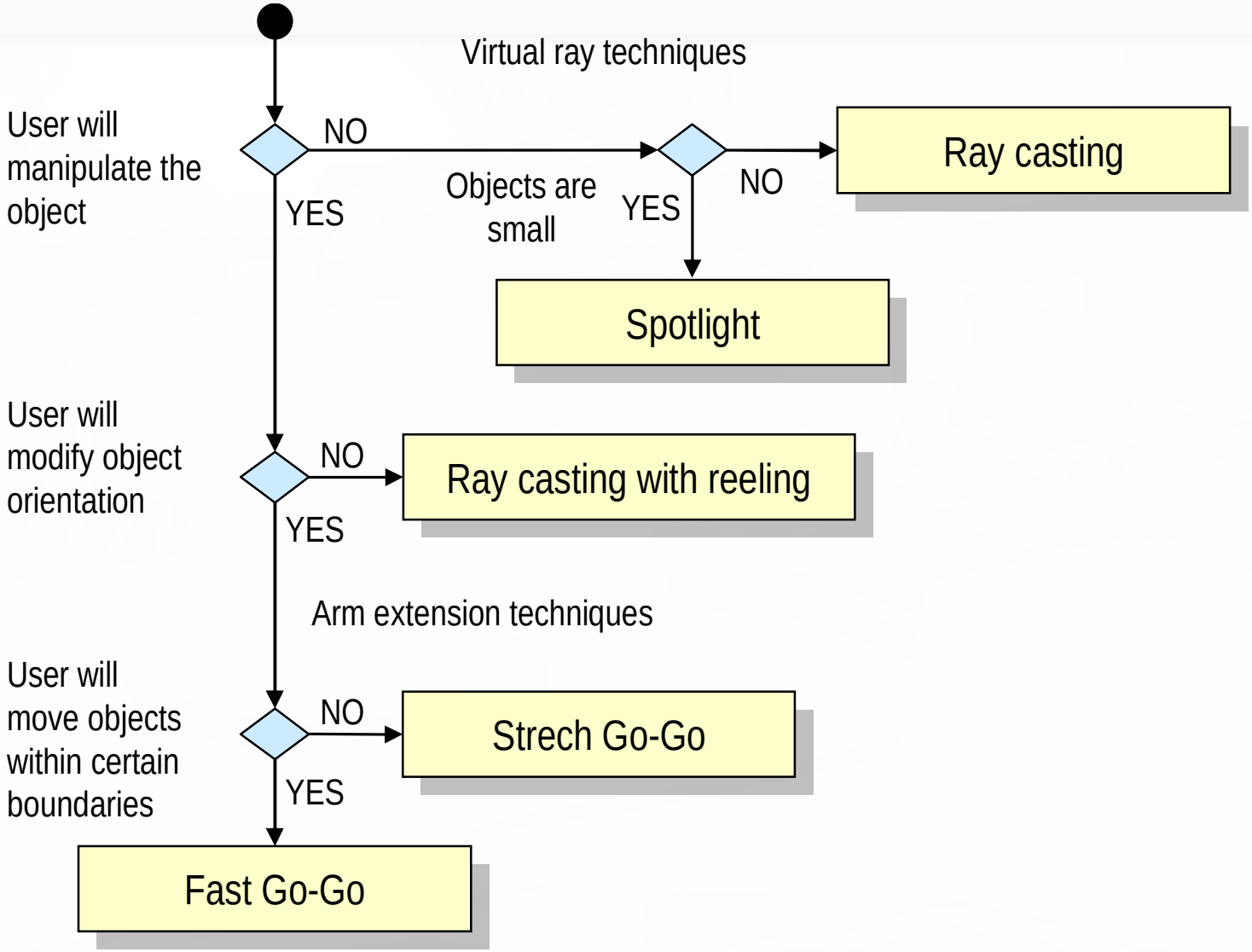
Exemplar advise tools

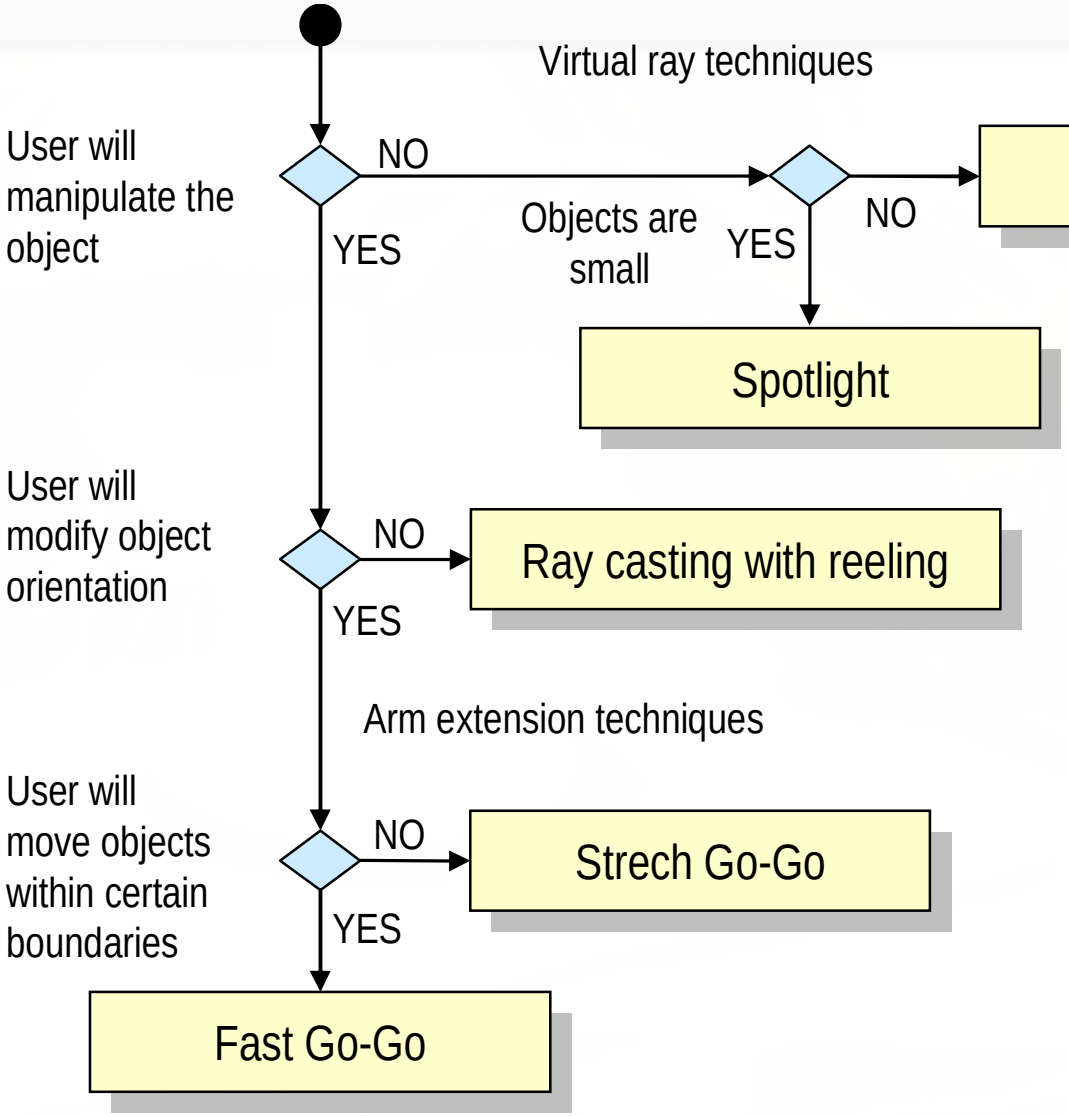
- These two tools are based on the results obtained in different experiments carried out at LoUISE VR lab, where several participants performed different tasks under different conditions.
- Similarly to previous works, such as [Eastgate, 01] or [Marsh, 98], these tools were given the form of a decision tree that guides the UI designer when deciding, depending on the specific conditions, which is the most appropriate interaction technique.

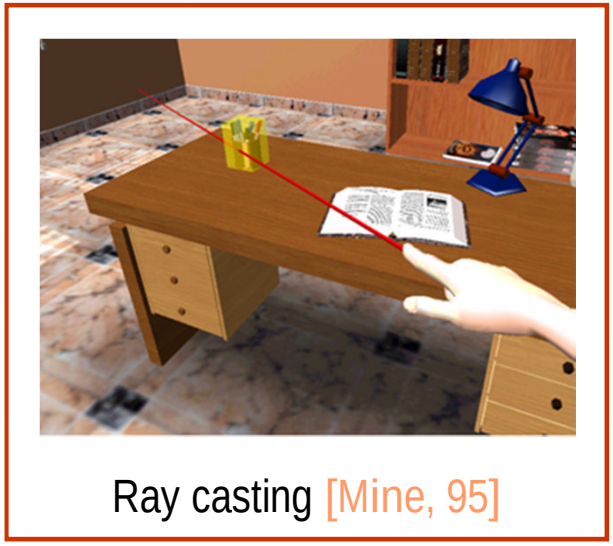
Exemplar advise tools

- For selecting and manipulating objects




Virtual ray techniques
User will manipulate the object
NO
YES
Objects are small
YES
NO
Ray casting
Spotlight
User will modify object orientation
NO
YES
Ray casting with reeling
Arm extension techniques
User will move objects within certain boundaries
NO
YES
Strech Go-Go
Fast Go-Go

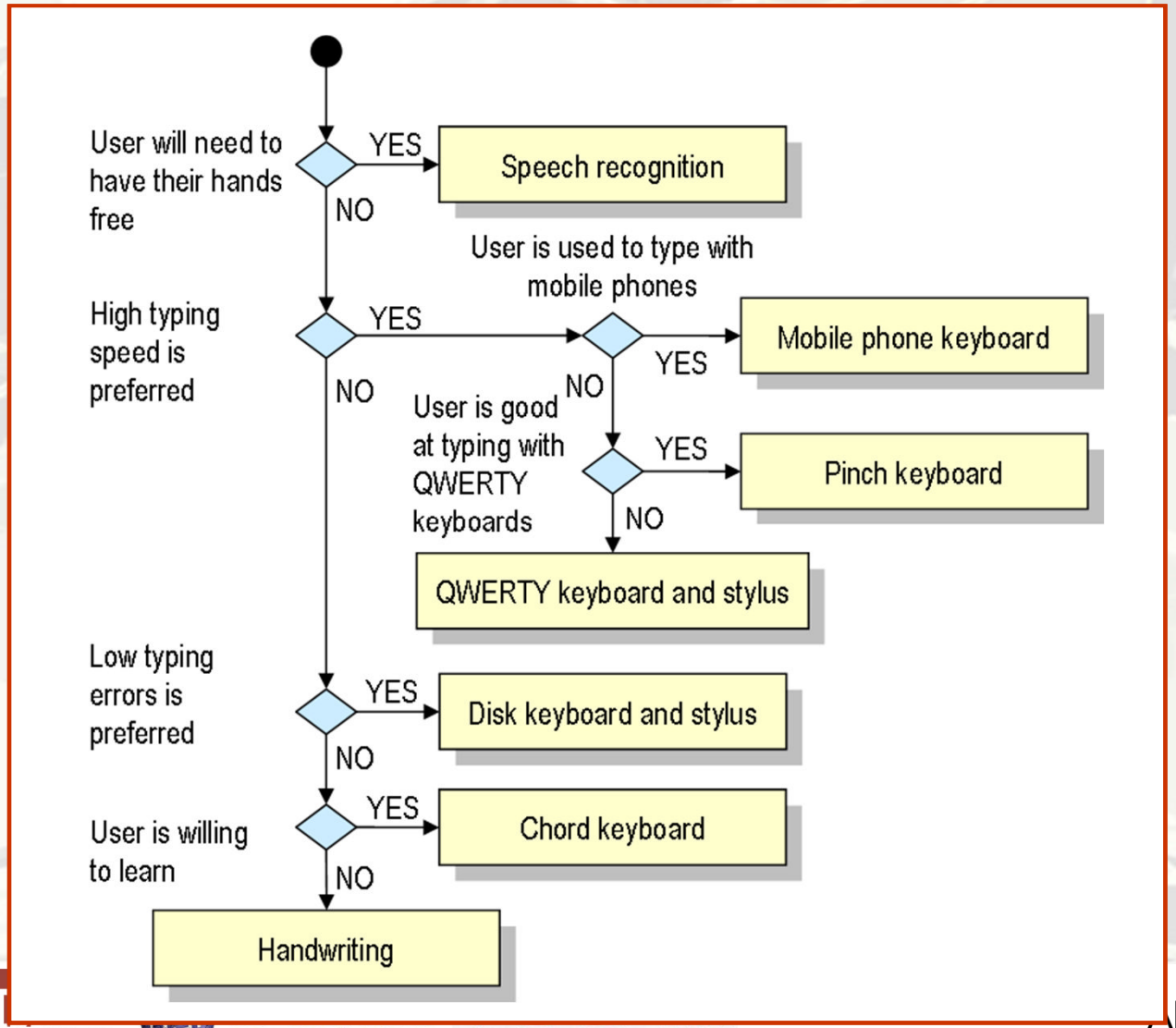




Ray casting [Mine, 95]

Exemplar advise tools

- For symbolic (text) input tasks



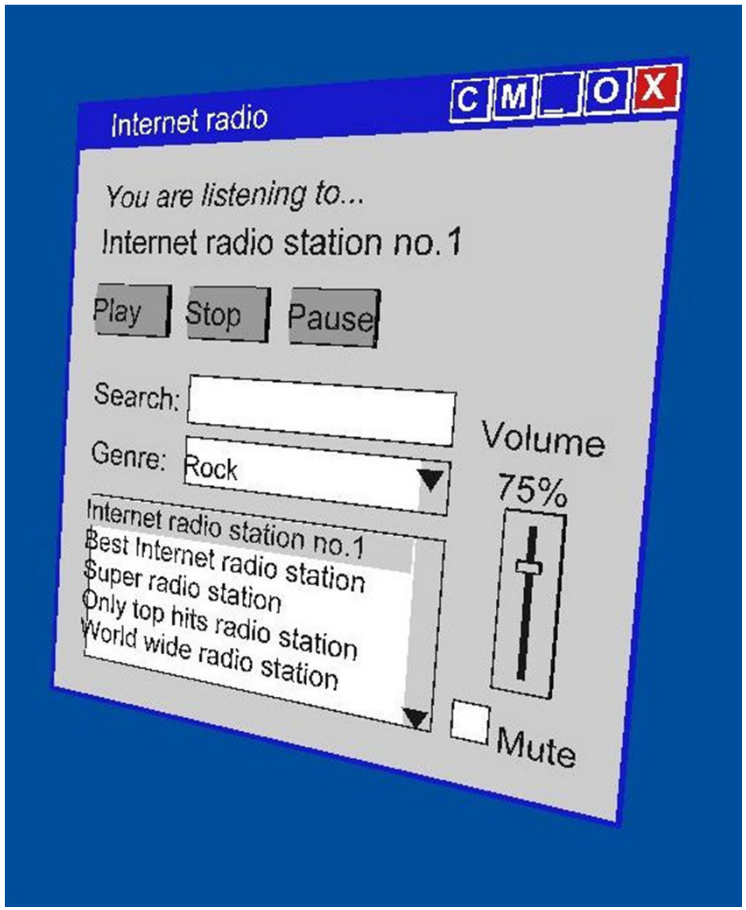
Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - **The VUIToolkit library**
  - Case studies
- Conclusions and further work

The VUIToolkit library

- Motivated by the lack of standard sets of controls for 3DUIs, and the need to support desktop UIs in 3D environments, this library is then introduced as **a set of pre-defined 3D widgets**, meant for **Virtualized User Interfaces** (**VUI**s).
- VUIToolkit offers a set of VRML97/X3D PROTOs that transform the standard plain widgets into a truly 3D representation.

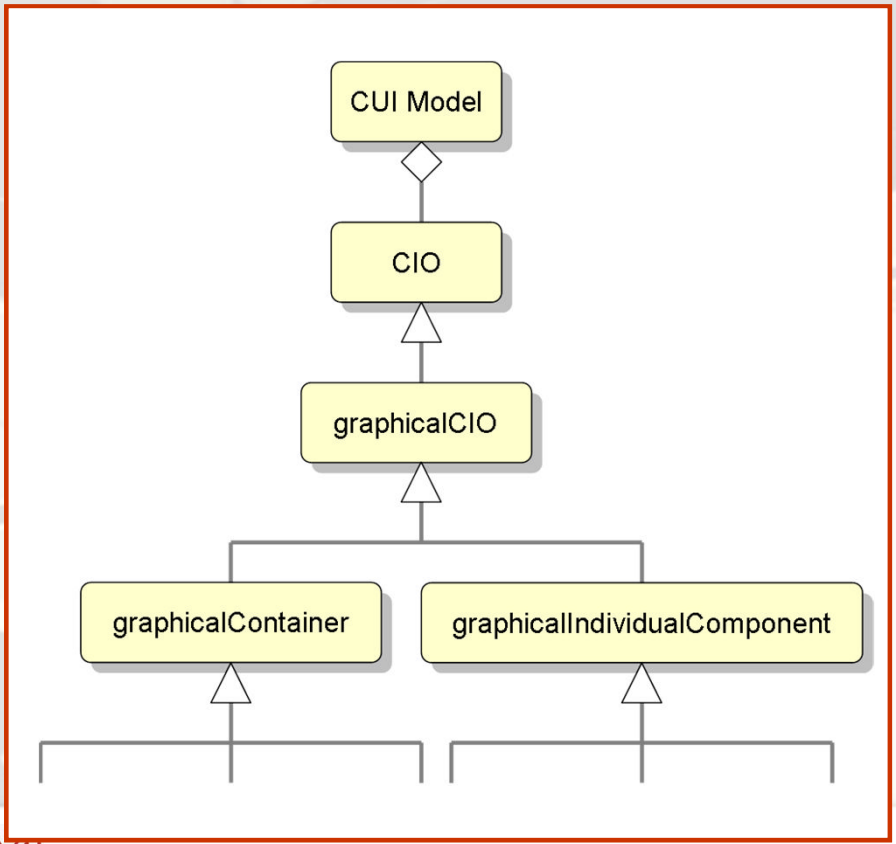
www.web3d.org



The VUIToolkit library

- VUIToolkit was developed starting from the object classes described in the **Concrete User Interface** (**CUI**) model of **UsiXML language**.
- UsiXML, a XML-compliant User Interface Description Language, provides a CUI description that remains independent from any toolkit, whether graphical or virtual.
- Thus, from a same specification of CUI, a GUI can be obtained in parallel to a virtualization of this GUI.

www.usixml.org



The VUIToolkit library

- The **architecture** for both toolkits is based on:
  - The prototyping mechanism available in both Web3D languages
  - Some ideas borrowed from the VRML Widget WG [VRML WWG, URL]
- Then, for each **Concrete Interaction Object** (**CIO**) of the CUI model, there is a final widget implemented as a PROTO that consists of:
  - An *interface* based on a set of fields
  - *Shapes* for the geometry and appearance of the widget
  - *Sensors* that allow user-widget interaction
  - *Internal logic* that manages the state of the widget
  - *Routes* that links shapes, sensors and logic
- Differences between both toolkits are meant to take advantage of new features of X3D specification.

The VUIToolkit library

- VUIToggleButton: VRML97 PROTO interface

```
PROTO VUIToggleButton [
# USIXML: cio
# ...
# USIXML: graphicalCio
# ...
# USIXML: graphicalIndividualComponent
# ...
# USIXML: toggleButton
field         SFBool   defaultState FALSE
# VRML97 VUI Toolkit fields
field         SFInt32  top    0
field         SFInt32  left   0
field         SFInt32  width  75
field         SFInt32  height 25
exposedField MFNode    label  [ ]

eventOut      SFTime   touchTime
eventIn       SFBool   set_state
eventOut      SFBool   state_changed

eventIn       SFInt32  set_top
eventIn       SFInt32  set_left
eventIn       SFInt32  set_width
eventIn       SFInt32  set_height
eventOut      SFInt32  top_changed
eventOut      SFInt32  left_changed
eventOut      SFInt32  width_changed
eventOut      SFInt32  height_changed
```

- From attributes specified for *toggleButton* in the CUI model.
- Some UsiXML attributes have not a meaning in the toolkit.
- Other attributes are added, plus *set_* and *_changed* fields.
- Type of attributes must be adapted, e.g.:
  - *boolean = SFBool*
  - *string*? If represents a colour, *SFColor* seems more appropriate.

The VUIToolkit library

- VUIToggleButton: X3D PROTO interface
  - *inputOutput* fields were not possible with VRML97 as this language does not allow us to match them with Script fields.

```
<ProtoDeclare name="VUIToggleButton">
<ProtoInterface>
<!-- USIXML: cio -->
<!-- . . . -->
<!-- USIXML: graphicalCio -->
<!-- . . . -->
<!-- USIXML: graphicalIndividualComponent -->
<!-- . . . -->
<!-- USIXML: toggleButton -->
  <field accessType="initializeOnly"
  name="defaultState"
  type="SFBool" value="false"/>
<!-- X3D VUI Toolkit fields -->
<field accessType="inputOutput" name="top"
   type="SFInt32" value="0"/>
<field accessType="inputOutput" name="left"
   type="SFInt32" value="0"/>
<field accessType="inputOutput" name="width"
   type="SFInt32" value="75"/>
<field accessType="inputOutput" name="height"
   type="SFInt32" value="25"/>
<field accessType="inputOutput" name="state"
   type="SFBool"/>
<field accessType="inputOutput" name="label"
   type="MFNode"/>
<field accessType="outputOnly"
   name="touchTime" type="SFTime"/>
```

The VUIToolkit library

- VUIToggleButton: VRML97 vs. X3D PROTO interface

```
PROTO VUIToggleButton [
# USIXML: cio
# ...
# USIXML: graphicalCio
# ...
# USIXML: graphicalIndividualComponent
# ...
# USIXML: toggleButton
field        SFBool   defaultState FALSE
# VRML97 VUI Toolkit fields
field        SFInt32  top    0
field        SFInt32  left   0
field        SFInt32  width  75
field        SFInt32  height 25
exposedField MFNode   label  [ ]

eventOut     SFTime   touchTime
eventIn      SFBool   set_state
eventOut     SFBool   state_changed

eventIn      SFInt32  set_top
eventIn      SFInt32  set_left
eventIn      SFInt32  set_width
eventIn      SFInt32  set_height
eventOut     SFInt32  top_changed
eventOut     SFInt32  left_changed
eventOut     SFInt32  width_changed
eventOut     SFInt32  height_changed
```

```
<ProtoDeclare name="VUIToggleButton">
<ProtoInterface>
<!-- USIXML: cio -->
<!-- . . . -->
<!-- USIXML: graphicalCio -->
<!-- . . . -->
<!-- USIXML: graphicalIndividualComponent -->
<!-- . . . -->
<!-- USIXML: toggleButton -->
  <field accessType="initializeOnly"
  name="defaultState"
  type="SFBool" value="false"/>
<!-- X3D VUI Toolkit fields -->
<field accessType="inputOutput" name="top"
   type="SFInt32" value="0"/>
<field accessType="inputOutput" name="left"
   type="SFInt32" value="0"/>
<field accessType="inputOutput" name="width"
   type="SFInt32" value="75"/>
<field accessType="inputOutput" name="height"
   type="SFInt32" value="25"/>
<field accessType="inputOutput" name="state"
   type="SFBool"/>
<field accessType="inputOutput" name="label"
   type="MFNode"/>
<field accessType="outputOnly"
   name="touchTime" type="SFTime"/>
```

The VUIToolkit library

- Input/output mapping from desktop-oriented 2D GUI to VRML/X3D virtual world:
  - **Display** = “Screen” PROTO.
    - Rectangular area where screen units (pixels) are converted into world units (meters).
    - Serves as a container of UI elements (*children* field).
  - **Mouse** = User’s pointing device.
    - VRML/X3D sensors.
  - **Keyboard?**
    - VRML: requires non-standard extensions (PG’s *KbdSensor*).
    - X3D: *StringSensor*.
    - Virtual keyboard.

The VUIToolkit library

- Input/output mapping from desktop-oriented 2D GUI to VRML/X3D virtual world:
  - **Display** = “Screen” PROTO.
    - Rectangular area where screen units (pixels) are converted into world units (meters).
    - Serves as a container of UI elements (*children* field).
  - **Mouse** = User’s pointing device.
    - VRML/X3D sensors.
  - **Keyboard?**
    - VRML: requires non-standard extension
    - X3D: *StringSensor*.
    - Virtual keyboard.

Virtual laptop

Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - **Case studies**
- Conclusions and further work

Case studies

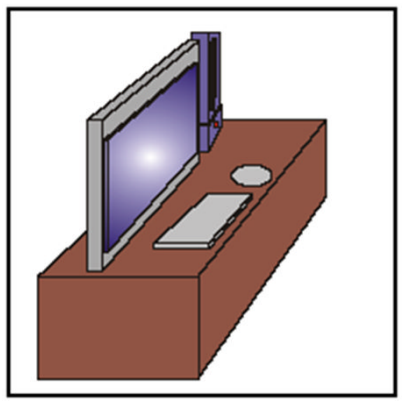
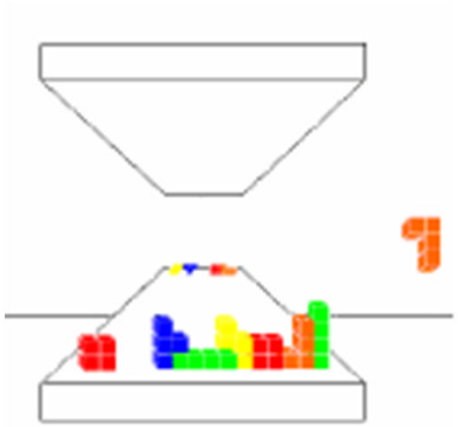
- Four case studies have been detailed, three of them showing the structured development of 3DUIs with the TRES-D methodology, and the forth showing the use of the VUIToolkit library:
  - **TTristéreo**: a desktop-based VR game
  - **VRPrismaker**: an immersive construction system
  - **Minority report**: as seen in the movie
  - **MigriXML**: prototyping distributed UIs

Case studies

- TTristéreo: a desktop-based VR game
  - As big panoramic displays are becoming a commodity these days, this project proposed their transformation into VR systems using 3D stereo graphics
  - Following the TRES-D methodology, different proposals where analyzed at the concept design stage, then deciding to design and develop a videogame based on the well-known Tetris, but introducing new features

Case studies

- TTristéreo: a desktop-based VR game (cont'd)
  - Pieces appear from the left side, the user have to grasp them, rotate and leave them fall down to the desired place by hand gestures in 3D space
  - The Essential Reality P5 data glove was considered as the right choice, as it allowed bend and position tracking at an affordable price

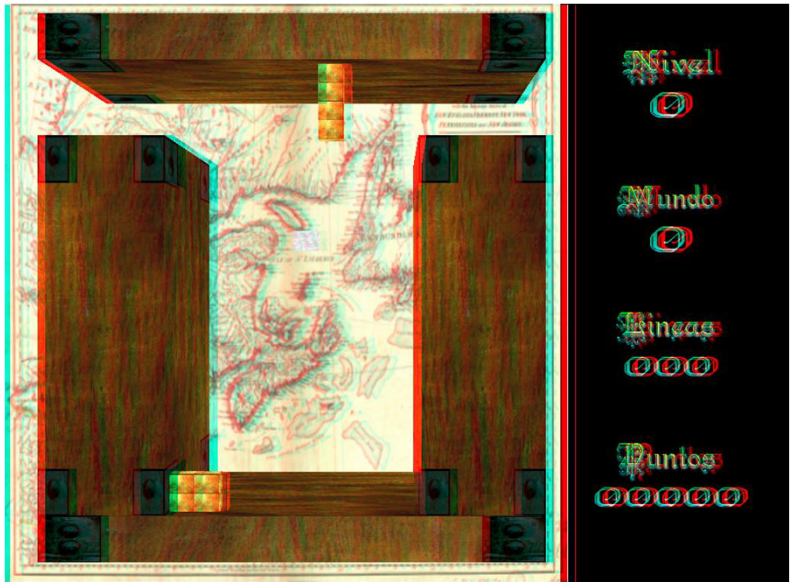




Case studies

- TTristéreo: a desktop-based VR game (cont'd)
  - In the design stage, the interaction tasks are described in terms of operations, their parameters and conditions.
  - At presentation level the designer detailed which P5 actions were associated to each operation. In other words, interaction techniques were selected.

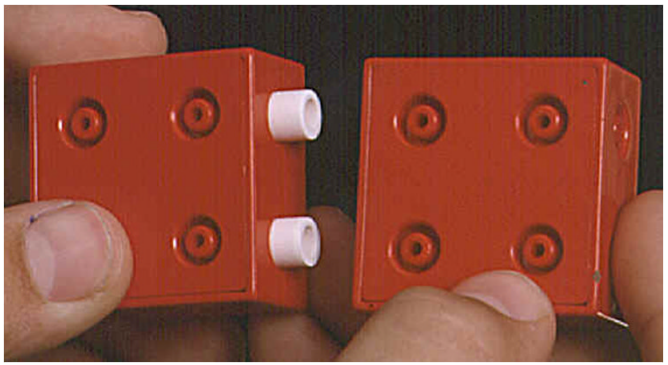
Case studies

- TTristéreo: a desktop-based VR game (cont'd)
  - This time, the P5 data glove allowed us to set a direct match between real world actions and these actions in the virtual environment.



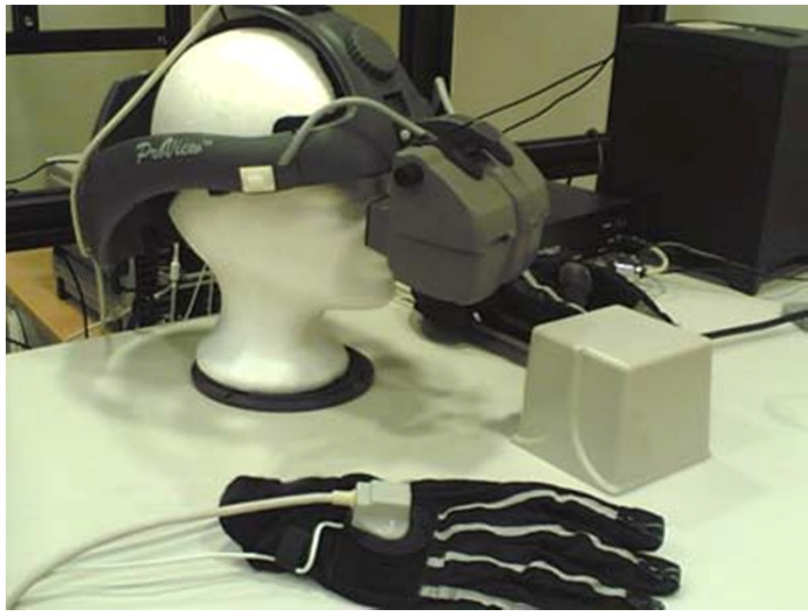
Case studies

- VRPrismaker: an immersive construction system
  - This second project pursued the virtualization of a block-based construction game, known as Prismaker, in a fully immersive Virtual Reality application, looking this time for the highest degree of naturalness.



Case studies

- VRPrismaker: an immersive construction system (cont'd)
  - Once the requirements were analyzed, the solution proposal stage followed by proposing an immersive Virtual Reality system based on HMD and data gloves.

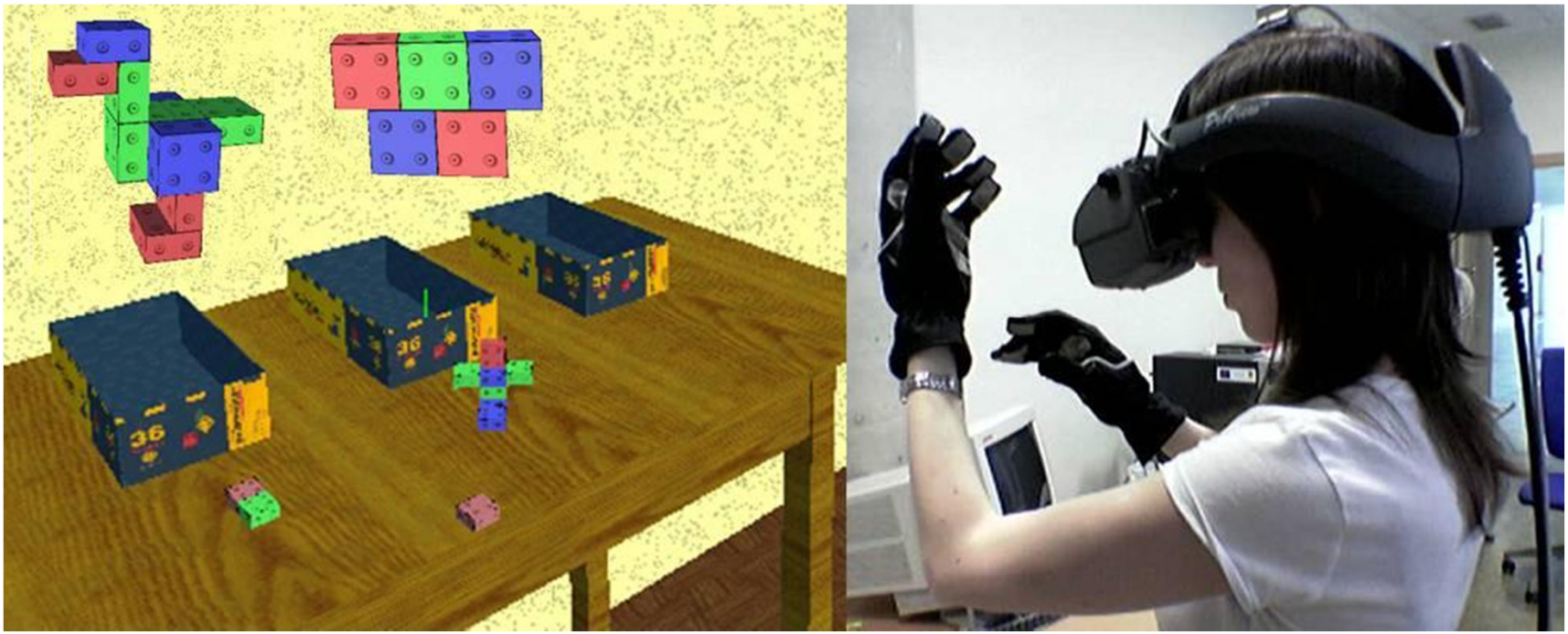


Case studies

- VRPrismaker: an immersive construction system (cont'd)
  - The detailed design started from the abstract level by analyzing the use cases, which were transformed this time into a list of operations more than a hierarchical task tree.
  - Every operation was based on two actions: *pick* and *drop*. At presentation level, those operations were match with a "pinch" gesture.

Case studies

- VRPrismaker: an immersive construction system (cont'd)



Case studies

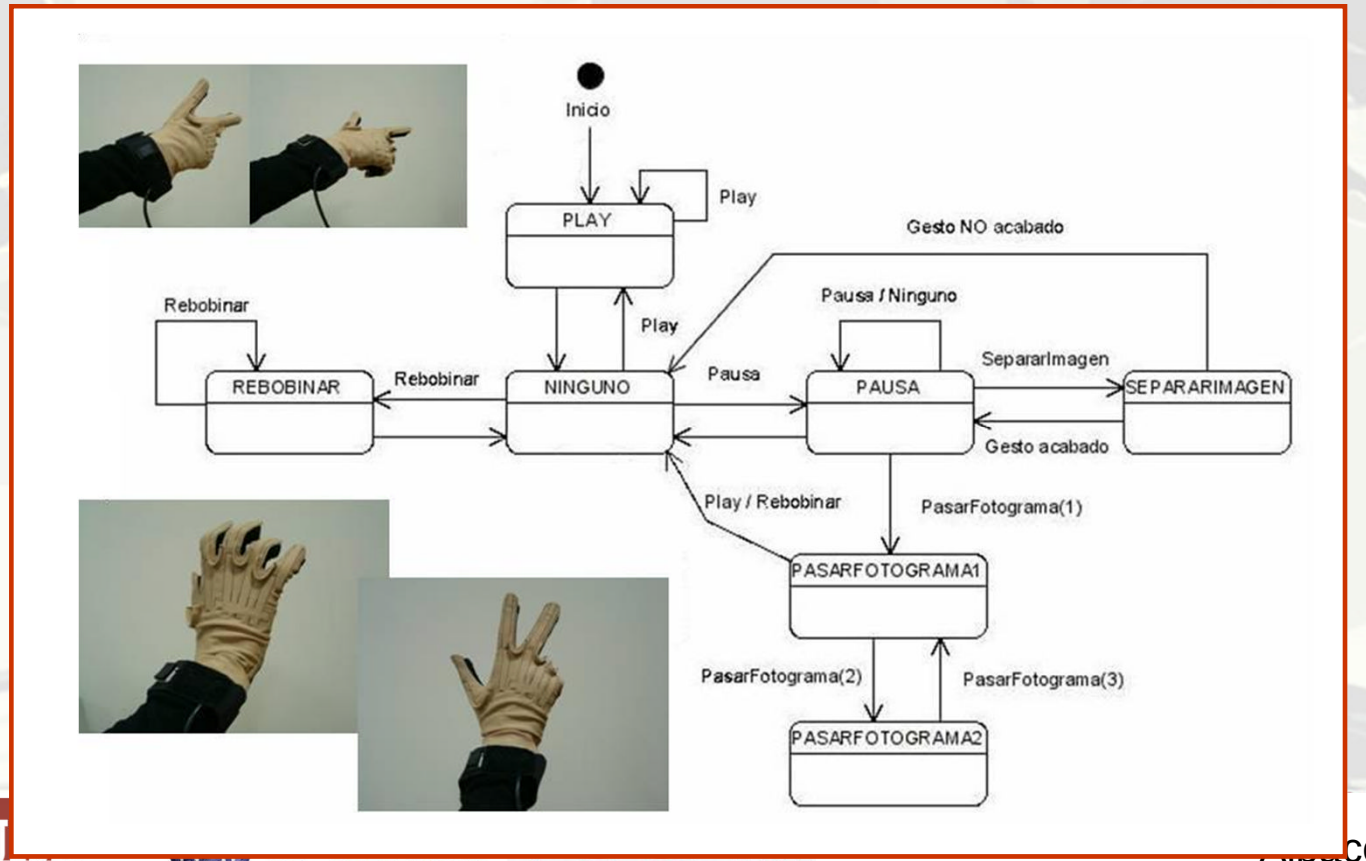
- Minority report: as seen in the movie
  - That movie shows an example of 3DUI where only input is three-dimensional, as the main character manipulates, by means of hand gestures in 3D space, images projected onto a flat crystal panel.
  - Those gestures were analyzed in the second stage of the TRES-D methodology, resulting in 11 possible actions, although the number of gestures was notably higher, 17 in total, as they differed from one actor to the other.

Case studies

- Minority report: as seen in the movie (cont'd)
  - In the concept design, it was proposed the Immersion Cyberglove as the input device, together with a Ascension Flock of Birds motion tracking system.

Case studies

- Minority report: as seen in the movie (cont'd)
  - At the beginning of the design stage, the UI was defined in a platform-independent way, using a state diagram to detail when the operations can be executed.



Case studies

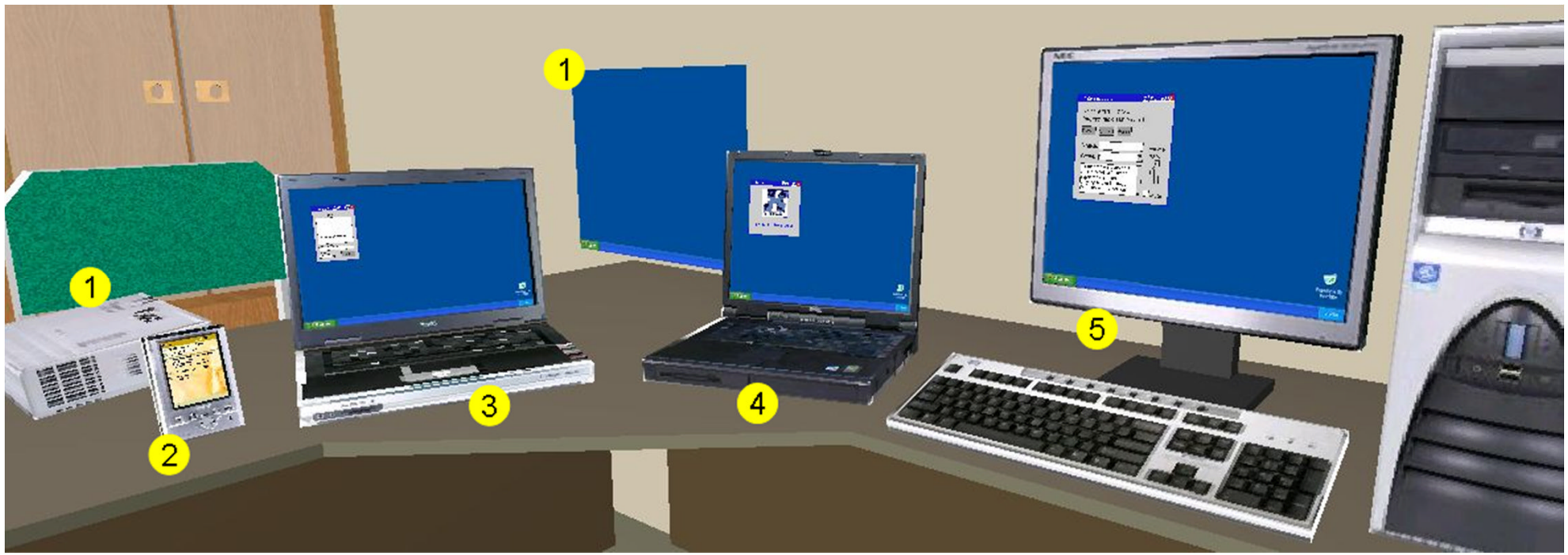
- Minority report: as seen in the movie (cont'd)
  - At presentation level, the interaction techniques were detailed, as for instance the ray-casting technique associated to the point and select gestures.

Case studies

- MigriXML: prototyping distributed UIs
  - The forth case study shows the VUIToolkit library as a tool for designing and prototyping **Distributed User Interfaces** (**DUI**s), which apply the notion of distributing parts or whole of a UI across several places or locations like Distributed Systems.
  - Starting from a UsiXML specification that describes the UI, 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 scene is generated and can be viewed with a Web3D browser.

Case studies

- MigriXML: prototyping distributed UIs (cont'd)
  - In this case study, the user's environment is a small office, with five different platforms.



Case studies

- MigriXML: prototyping distributed UIs (cont'd)
  - The simulation also renders the platforms and the UIs of the applications that are executed in these platforms, using the VUIToolkit to represent such interfaces and allow their manipulation and distribution across different platforms, all with prototyping purposes.

Contents

- Introduction
- The 3D user interface and its design
- Methods for the development of 3DUIs
- The TRES-D methodology
  - Meta-models for a new methodology
  - The process model of the TRES-D methodology
  - Exemplar advise tools
  - The VUIToolkit library
  - Case studies
- **Conclusions and further work**

Conclusions and further work

- **3DUIs have not reached the same state of maturity that can be observed in desktop interfaces**, where well funded methods and practices allow a rapid development, highly automated.
- This doctoral dissertation then focused on existing approaches to the design and creation of 3DUIs. Many proposals were found, and many more may have been ignored.
- Following a critical review of all of them, it was concluded that a new proposal was needed. That proposal was named **TRES-D**.

Conclusions and further work

- The TRES-D methodology is characterized by:
  - First, dividing the development into two 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-
  - Second, design stage is 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
  - And third, at design and implementation stages, the UI designers and programmers tackle tasks and interaction and, in parallel, artists and digital content creators focus on objects and content

Conclusions and further work

- The TRES-D methodology is characterized by:
  - **First, dividing the development into two 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-**
  - Second, design stage is 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
  - And third, at design and implementation stages, the UI designers and programmers tackle tasks and interaction and, in parallel, artists and digital content creators focus on objects and content

Conclusions and further work

- The TRES-D methodology is characterized by:
  - First, dividing the development into two 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-
  - **Second, design stage is 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**
  - And third, at design and implementation stages, the UI designers and programmers tackle tasks and interaction and, in parallel, artists and digital content creators focus on objects and content

Conclusions and further work

- The TRES-D methodology is characterized by:
  - First, dividing the development into two 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-
  - Second, design stage is 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
  - **And third, at design and implementation stages, the UI designers and programmers tackle tasks and interaction and, in parallel, artists and digital content creators focus on objects and content**

Conclusions and further work

- The TRES-D methodology is characterized by:
  - First, dividing the development into two 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-
  - Second, design stage is 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
  - And third, at design and implementation stages, the UI designers and programmers tackle tasks and interaction and, in parallel, artists and digital content creators focus on objects and content
- However, **this work is not simply a sequence of phases, stages and activities**. The following slides sum up the contributions of this doctoral dissertation. Many of them have been presented in national and international scientific events.

Conclusions and further work

- Contributions
  - ✓ Many terms and definitions are related to 3D, so the first step 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 in [Molina, 05a].
  - ✓ The different elements that compose a 3DUI have been identified and described: 3D space, objects, behaviour and interaction. This work served as a base for the definition of the three meta-models included in the TRES-D proposal.

Conclusions and further work

- Contributions
  - ✓ Many terms and definitions are related to 3D, so the first step 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 in [Molina, 05a].
  - ✓ The different elements that compose a 3DUI have been identified and described: 3D space, objects, behaviour and interaction. This work served as a base for the definition of the three meta-models included in the TRES-D proposal.

Conclusions and further work

- Contributions
  - ✓ Many terms and definitions are related to 3D, so the first step 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 in [Molina, 05a].
  - ✓ The different elements that compose a 3DUI have been identified and described: 3D space, objects, behaviour and interaction. This work served as a base for the definition of the three meta-models included in the TRES-D proposal.

Conclusions and further work

- Contributions
  - ✓ Many terms and definitions are related to 3D, so the first step 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 in [Molina, 05a].

J.P. Molina, J. Vanderdonckt, F. Montero & P. González. Towards Virtualization of User Interfaces based on UsiXML. Proc. of **ACM Web3D** 2005 Symposium, March – April 2005, University of Wales, Bangor, UK, pp. 169-179.

Conclusions and further work

- Contributions
  - ✓ Many terms and definitions are related to 3D, so the first step 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 in [Molina, 05a].
  - ✓ The different **elements that compose a 3DUI** have been identified and described: *3D space*, *objects*, *behaviour* and *interaction*. This work served as a basis for the definition of the three meta-models included in the TRES-D proposal.

Conclusions and further work

- Contributions (cont'd)
  - ✓ The **design problems related to 3DUIs** have being studied. On the one hand, interaction does not follow the conventional turn-taking dialogue. On the other hand, there is no standard set of ITes nor controls. These findings encouraged the proposal of advise tools and toolkits in the TRES-D proposal.
  - ✓ 26 methodologies have been described and analyzed, grouping them in 7 different categories according to some similarities found in these approaches. This large review is summarized in [Molina, 05b].

Conclusions and further work

- Contributions (cont'd)
  - ✓ The design problems related to 3DUIs have being studied. On the one hand, interaction does not follow the conventional turn-taking dialogue. On the other hand, there is no standard set of ITes nor controls. These findings encouraged the proposal of advise tools and toolkits in the TRES-D proposal.
  - ✓ **26 methodologies have been described and analyzed**, grouping them in 7 different categories according to some similarities found in these approaches. This large review is summarized in [Molina, 05b].

J.P. Molina, A.S. García, V. López & P. González . Developing VR applications: the TRES-D methodology. 1st International Workshop on Methods and Tools for Designing VR applications (**MeTo-VR**) at VSMM'05, Ghent, Belgium, 3 October 2005.

Conclusions and further work

- Contributions (cont'd)
  - ✓ Based on the experience gained through the application of the IDEAS methodology to the development of desktop and Web interfaces, a first proposal is presented, named **IDEAS-3D**. This is presented in [Molina, 03a], [Molina, 03b] and [Molina, 03c].

J.P. Molina, P. González, M.D. Lozano, F. Montero & V. López. Bridging the gap: developing 2D and 3D user interfaces with the IDEAS methodology. In DSV-IS: Issues in Designing New-generation Interactive Systems. **Springer-Verlag LNCS**, 2003, pp. 303-315.

J.P. Molina, P. González y M.D. Lozano. Desarrollo de interfaces de usuario tridimensionales: un enfoque metodológico basado en IDEAS. Actas del IV Congreso Internacional Interacción Persona-Ordenador, **Interacción**'03,, Vigo, junio 2003. Reprogalicia Edicións. CD-ROM.

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.

Conclusions and further work

- Contributions (cont'd)
  - ✓ Based on the experience gained through the application of the IDEAS methodology to the development of desktop and Web interfaces, a first proposal is presented, named IDEAS-3D. This is presented in [Molina, 03a], [Molina, 03b] and [Molina, 03c].
  - ✓ Heading to the TRES-D framework, **three new meta-models** were then introduced to better understand 3DUIs and facilitate their design and implementation: *object*, *interaction* and *space*. The interaction meta-model is described in [Molina, 06b].

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 MELECON**, special session "New interaction paradigms in Virtual Environments", Málaga, May 16-19, 2006.

Conclusions and further work

- Contributions (cont'd)
  - ✓ As part of the framework, **two advise tools** have been presented, which help designers selecting interaction techniques. One is for selection/manipulation techniques, and it is described in [García, 05a] and [García, 05b]. The other one is for text input techniques, and it is described in [González, 07].

A.S. García, J.P. Molina & P. González. Exemplar VE design guidance tool for selection and manipulation interaction techniques. Proc. of 11th **HCI International** Conference, Las Vegas, Nevada, USA, 22-27 July 2005.

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.

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.

Conclusions and further work

- Contributions (cont'd)
  - ✓ As part of the framework, **two advise tools** have been presented, which help designers selecting interaction techniques. One is for selection/manipulation techniques, and it is described in [García, 05a] and [García, 05b]. The other one is for text input techniques, and it is described in [González, 07].
  - ✓ A toolkit for programmers has also been included, to help them translating flat interfaces into 3D environments. Its name is **VUIToolkit**, and was first introduced in [Molina, 05a].

Conclusions and further work

- Contributions (cont'd)
  - ✓ Finally, the proposed framework has been applied in **four case studies**. Three of them demonstrate the process of the TRES-D methodology, and have been detailed in [Molina, 06b] and [Molina, 06b]. …

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 VRST**, Symposium on Virtual Reality Software and Technology, Cyprus, Nov 2006.

Conclusions and further work

- Contributions (cont'd)
  - ✓ Finally, the proposed framework has been applied in **four case studies**. Three of them demonstrate the process of the TRES-D methodology, and have been detailed in [Molina, 06b] and [Molina, 06b]. The forth one showed the application of VUIToolkit in the design and prototyping of distributed UIs, as seen in [Molina, 06a], [Molina, 06c] and [Vanderdonckt, 07].

J.P. Molina, J. Vanderdonckt & P. González. Direct manipulation of User Interfaces for Migration. Proc. of 10th **ACM IUI**, Int. Conf. on Intelligent User Interfaces, Sydney, Australia, January - February 2006, pp. 140-147.

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

J. Vanderdonckt, H. Mendonça & 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 LNCS**, 2007, pp. 44-52.

Conclusions and further work

- Projects
  - This work has been supported by different regional and national projects, in which the candidate has been involved:
    - **"Entornos virtuales colaborativos aplicados a sistemas de aprendizaje"**, funded by Junta de Comunidades de Castilla-La Mancha (JCCM), ref. PAI-06-0093, 2006-2008.
    - **"Sistemas Adaptativos y Colaborativos con soporte WEB (ADACO)"**, funded by Ministerio de Ciencia y Tecnología (MCYT), ref. TIN2004-08000-C03-01, 2005-2007.
    - **"Metodologías de desarrollo de interfaces de usuario dinámicas"**, funded by Junta de Comunidades de Castilla-La Mancha (JCCM), ref. PBC-03-003, 2003-2005.

Conclusions and further work

- Scientific stay
  - Part of this work has been done during a stay of 7 months at the **Université catholique de Louvain** (**UCL**), Belgium, as a member of the **Belgian Computer-Human Interaction** (**BCHI**) group, being supervised by **Prof. Dr. Jean Vanderdonckt**.

Conclusions and further work

- Further work
  - The concept of **sub-models** is presented in the object meta-model so that each object can be described from different views, 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 also guide developers in their work, as a public repository.
  - TRES-D does not achieve the automation that was aimed when IDEAS-3D was presented. Further work on the proposed meta-models must be done.
  - New decision trees could be produced for other universal tasks, such as navigation and system control. A pattern format could also detail when each recommendation applies.

Conclusions and further work

- Further work
  - The concept of sub-models is presented in the object meta-model so that each object can be described from different views, 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 also guide developers in their work, as a public repository.
  - TRES-D does not achieve the automation that was aimed when IDEAS-3D was presented. Further work on the proposed meta-models must be done.
  - New decision trees could be produced for other universal tasks, such as navigation and system control. A pattern format could also detail when each recommendation applies.

Conclusions and further work

- Further work
  - The concept of sub-models is presented in the object meta-model so that each object can be described from different views, 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 also guide developers in their work, as a public repository.
  - TRES-D does not achieve the **automation** that was aimed when IDEAS-3D was presented. Further work on the proposed meta-models must be done.
  - New decision trees could be produced for other universal tasks, such as navigation and system control. A pattern format could also detail when each recommendation applies.

Conclusions and further work

- Further work
  - The concept of sub-models is presented in the object meta-model so that each object can be described from different views, 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 also guide developers in their work, as a public repository.
  - TRES-D does not achieve the automation that was aimed when IDEAS-3D was presented. Further work on the proposed meta-models must be done.
  - **New decision trees** could be produced for other universal tasks, such as navigation and system control. A **pattern format** could also detail when each recommendation applies.

Conclusions and further work

- Further work (cont'd)
  - As for the **VUIToolkit**, there are still CUI classes that can be added, such as the CUI Box. It could even be ported to other development environments, for example VR Juggler.
  - VR Juggler is currently used at the LoUISE VR lab to develop Collaborative Virtual Environments or CVEs. In this development, TRES-D is also applied. However collaboration puts forward important challenges that open a whole new line of research.

Conclusions and further work

- Further work (cont'd)
  - As for the VUIToolkit, there are still CUI classes that can be added, such as the CUI Box. It could even be ported to other development environments, such as VR Juggler.
  - VR Juggler is currently used at the LoUISE VR lab to develop **Collaborative Virtual Environments** or **CVE**s. In this development, TRES-D is also applied. However collaboration puts forward important challenges that open a whole new line of research.

Thesis submitted to the University of Castilla-La Mancha for the European degree of Doctor of Philosophy in Computer Science

Albacete, 29 February 2008

A Structured Approach to the Development of 3D User Interfaces

Thank you very much!

¡Muchas gracias!

I3A Instituto de Investigación en Informática de Albacete

LoUISE Laboratorio de Interacción con el Usuario e Ingeniería del Software

DSI Departamento de Sistemas Informáticos

UCLM Universidad de Castilla-La Mancha