

Chapter 10

A Space Model for 3D User Interface Development

José P. Molina, Pascual González, Jean Vanderdonckt,
Arturo S. García and Diego Martínez

Abstract Space deserves a special attention when designing 3D user interfaces. However, many proposed methods rely on simple sketches and/or maps. Besides, those methods usually leave aside the 2D interfaces that may be found in the 3D environment or that surround it. This chapter presents an attempt to formalize 2D and 3D spaces by the definition of a meta-model. This is one of three meta-models that have been defined in *three-dimensional user interface development* methodology, which is also described here along with some case studies, focusing the description on space.

10.1 Introduction

As a definition, a 3D user interface (3D UI) is one 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. That is, either the user employs 3D devices to interact with the system, or the system renders 3D images to the user, or both. Over the years, there has been a remarkable effort to understand these 3D UIs through the tasks, the interaction techniques that were proposed to support them and the systematic evaluation of such techniques. However, if there is anything that characterizes these interfaces, this is precisely the 3D space, so it is important to understand and model it.

In the case of 2D graphical user interfaces (2D GUIs), the methods proposed for their development frequently included some sort of 2D space modelling, for example with diagrams that specify the layout of the controls on a window or diagrams that specify navigation paths through windows. These specifications are then used by programmers to create the final interface, for example LUCID [1] and OVID [2],

J.P. Molina (✉), P. González, J. Vanderdonckt, A.S. García and D. Martínez
Laboratory of User Interaction and Software Engineering (LoUISE), Instituto
de Investigación en Informática de Albacete (I3A), Universidad de Castilla-La Mancha,
Campus universitario s/n, S-02071 Albacete, Spain
e-mail: jpmolina@dsi.uclm.es

or compiled to generate it automatically, for example IDEAS [3]. In the case of 3D UIs, proposed methods usually rely on sketches and maps to specify the position of objects in 3D space, for example Fencott [4], SENDA [5] and IDEAS-3D [6], a sort of similar specification that, this time, is delivered to content creators that shape this space and its objects using 3D modelling tools.

However, previous methods usually forget that 3D environments do not exclude 2D GUIs [7], so it must be considered how to specify not only the two spaces but also their relationships. Granollers et al. [8], for instance, introduced an augmented dialogue model to match positions in the real world with the dialogues of a conventional interface, in the context of an Augmented Reality (AR) system. Furthermore, real space is not only of interest in systems that augment real places with virtual objects, but it is the space that devices and users belong to, so it should also be considered in the development of any kind of interface.

In this context, this chapter examines the problem of understanding and modelling space, or rather the spaces, and the relationships between them. To do so, this chapter will first present an extension to the Reality–Virtuality continuum by Milgram and Kishino [9], in order to cover three spaces, the real, the virtual and the so-called digital one, which is also useful to understand the evolution of UIs from one to three dimensions. Then, based on this new continuum, it will be presented the space meta-model proposed in this work, again distinguishing between those three spaces, but also describing their relationships.

This meta-model is actually one of the three that have been proposed for the development of 3D UIs. With respect to the other two, there is one that tackles objects and their classification, and the other one defines interaction elements and their relationships. The latter is described in [10]. Together they provide the language that supports a new methodology called *ThRee-Dimensional uSer interface Development* (TRES-D), a structured approach that is oriented to task and interaction as well as to objects and content, being space the third apex of this triangle. The process defined in this methodology is also described in this chapter, as well as some case studies, focusing on space and the proposed meta-model.

10.2 Extending the UI Continuum

On the basis of continuum by Milgram and Kishino, a new one is proposed with the aim of offering a wider view of UIs, considering not only 3D UIs, but also 2D GUIs and command-based interfaces, that is, 1D UIs. Figure 10.1 shows this new continuum.

In this proposal, the UIs are distributed along one axis according to their number of dimensions, from one to three. This way, the continuum represents a wider one, named *digital–virtual–real*, using “digital” to refer to the world of binary digits used in computers as opposed to the world of real objects where humans live. Besides, in the same way, the intermediate space between the real and the virtual is called *Mixed Reality*; there is an intermediate space between the digital and the virtual too, named *Mixed Virtuality*. 3D desktops fit in this new intermediate

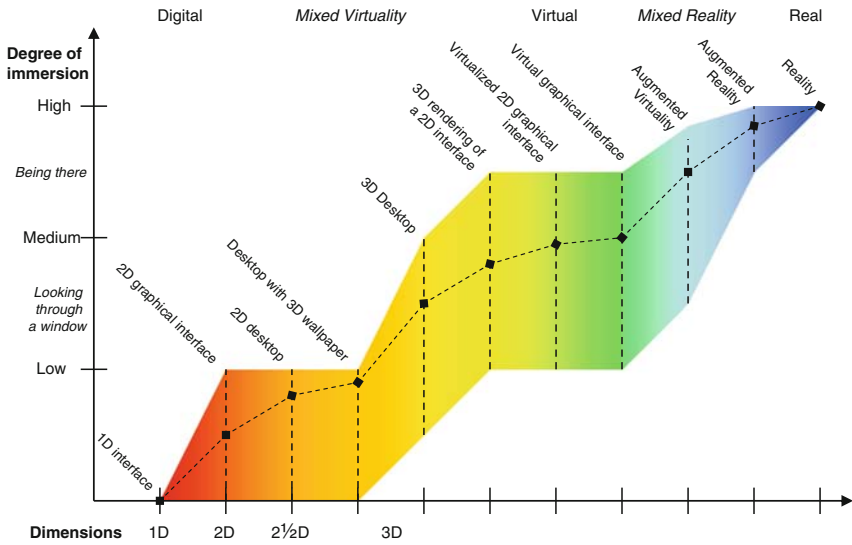


Fig. 10.1 Digital–virtual continuum for user interfaces

space, such as Win3D [11] or 3DNA [12], where the wallpaper is replaced by a 3D world, and so do Cube [13], SphereXP [14], Looking Glass [15] and Compiz Fusion [16], where windows are rendered as textures on flat surfaces that can be moved further or closer, and even be rotated. Transforming widgets into true 3D objects takes the UI one step further, resulting in a virtualized user interface (VUI) [7].

Furthermore, this new digital–virtual–real continuum adds a second axis that is related to the degree of immersion that, compared to reality, is experienced by the user when interacting with the space provided by the interface. This vertical axis follows this scale: *zero, low, medium, high*. Thus, when an interface is classified between the first two values, it means that users do not feel immersed in the space, or that feeling is very poor. In these interfaces, some depth cues may be given, such as shadows or overlays in 2D desktop interfaces, but that does not make the user feel in the space. Between low and medium, the user looks at the space through a window, possibly interacting with objects within using the “TV model,” using the same term as by van Dam [17]. Between medium and high, the user does feel immersed in the 3D space. Current limits of Virtual Reality (VR) technology prevent it from offering the highest degree of immersion, which belongs to reality.

10.3 A Space Meta-Model

This meta-model is inspired in the digital–virtual–real continuum introduced in Sect. 10.2, making a distinction between these three spaces, as the diagram shown in Fig. 10.2 illustrates. That diagram also shows an aggregation relationship between objects and

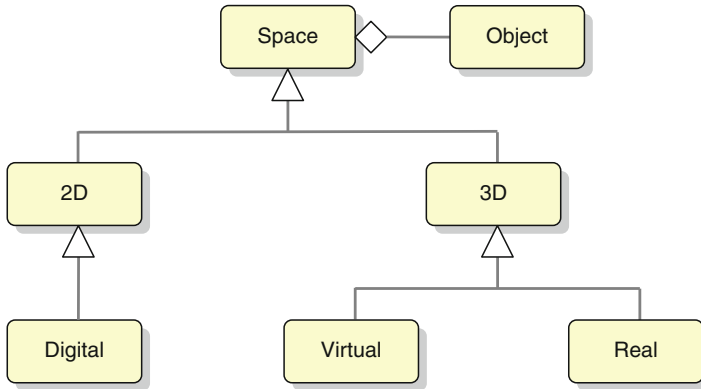


Fig. 10.2 Digital, virtual and real spaces in the meta-model

space, as the objects shape the space. The following paragraphs detail this meta-model through each space, paying special attention to the way they relate to each other.

10.3.1 *The Digital 2D Space*

As stated in Sect. 10.1, the idea of modelling space is not associated to the design of 3D UIs alone, and these interfaces do not exclude 2D interaction. Examples of modelling 2D space in conventional interfaces are the dialogue interaction and the component specification diagrams introduced by the IDEAS methodology in its dialogue model, graphically defining the navigation paths and the layout of window elements.

In general, the layout of controls in a window can be formalized defining a set of spatial relationships. Trevisan et al. [18] explain how to generalize Allen's 13 temporal relationships to 2D space, building a matrix (*Allen's 2D matrix*) that formalizes the possible space relationships between two controls in a 2D GUI. This formalization can also be useful in this part of the meta-model that is being proposed, as 2D controls can also be found in 3D UIs. Indeed, there is one 2D control that deserves special attention, the viewport, which represents the link between the digital and the virtual space, as it will be explained next.

10.3.2 *The Virtual 3D Space*

When working on IDEAS-3D, a previous attempt to 3D UIs development by extending IDEAS methods, first thoughts led to relate *viewpoints* and *places* in virtual worlds to dialogues in 2D GUIs. In fact, a window represents a *view* of the application dialogue. And, in the same way, the user of a 2D interface navigates through windows, the user of a virtual world can also jump from one viewpoint to another, from place to place.

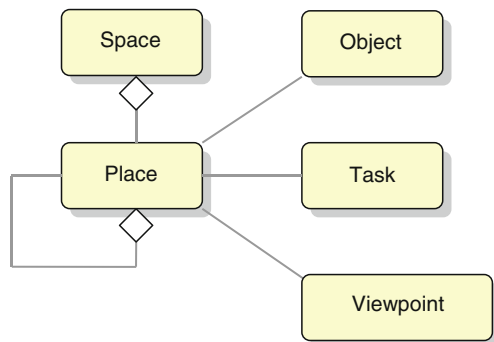


Fig. 10.3 The concept of place is common to all spaces

This follows the idea proposed in the RealPlaces design guide [19], so that the virtual space is structured in places in a hierarchical way, then associating an appropriate viewpoint to each place. In the IDEAS-3D methodology, that hierarchy of places mirrors the one of tasks, and so it is also a mechanism to relate space with the rest of models.

However, in both RealPlaces and IDEAS-3D, the world in mind was a horizontal, flat one, where the user navigates following the terrain, only taking off of it to jump from one place to another. However, such a conception of 3D space is limited for the general development of 3D UIs, such as the virtual interfaces found in many virtual worlds. Thus, places of interest are not always open spaces related to particular tasks; in some virtual worlds, the avatar can fly or orbit and, in general, there is not a single world but a set of interconnected worlds. At the same time, the identification of distinct regions in space can also be useful for other purposes, such as to specify the boundaries of the virtual world that limits the avatar movement, the maximum distance illuminated by the rays of a light source or reached by the waves of a sound source, or the view frustum associated to a viewpoint.

Then, even though the concept of place is included in this model, it gives way to a free structuring that can serve to different purposes needed in 3D UIs. As shown in Fig. 10.3, this concept of place is also common to the three spaces, although it is in the virtual one, here described, and in the real one, described later, where it has more relevance. That figure also shows the relationship that a place can have with objects, tasks and viewpoints. A place can also relate with other places, for instance building a hierarchy as in RealPlaces, and so making that hierarchy extensible to objects, tasks and viewpoints that relate to those places.

That freedom when structuring 3D space does not mean that this space cannot be formalized. In fact, the previous proposal from Trevisan et al. could be brought to this space too, generalizing Allen's relationships to 3D space by building a new matrix, which should now formalize the spatial relationships between two objects in a virtual world (and then named *Allen's 3D matrix*).

Furthermore, it could be said that this part of the model does not only cover the space in virtual worlds, featuring their three dimensions, but also the space of windows and widgets, as the latter could be described as a subspace of the former (Fig. 10.4).

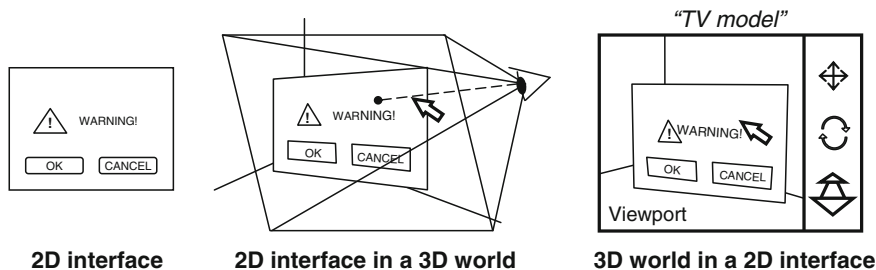


Fig. 10.4 A 2D interface in a 3D world and vice versa

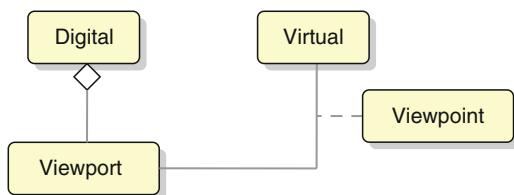


Fig. 10.5 A viewport serve as a link between digital and virtual spaces

However, integrating both spaces is not usually a straightforward task. On the one hand, in contrast to the components of a window, the objects found in virtual worlds are not measured in screen units – pixels, but in physical units such as metres. One solution to this is to render 2D GUIs on the screen of a virtual computer in the 3D space, a solution proposed in [20] and realized in [7].

On the other hand, 3D environments are also integrated in 2D interfaces, in this case by means of an element named *viewport* (see Fig. 10.5), cited in Sect. 10.3.1, a 2D component that is used to project the virtual world onto a plane using a viewpoint as the centre of projection. The viewport is, this way, the retina of the eye of the user’s avatar and, as such, is part of that avatar, as the rest of its virtual representation. As a 2D component, the viewport is integrated in a 2D GUI, being usually surrounded by other 2D controls in desktop applications, which represents the “TV model” cited before (see Fig. 10.4). In other cases, particularly in immersive applications, this component fully covers the screen, and those 2D controls are then rendered within the viewport as a *Head Up Display*. Besides, a 2D interface can have several viewports showing different views on the same virtual world, even in different screen displays, as it will be seen in the next section.

10.3.3 The Real 3D Space

In an AR application, space may be a building, for example a museum, or a place outdoors, for example an archaeological site. In the case of VR or Augmented Virtuality,

it is easier to find smaller places, such as the one that fills a CAVE-like installation or even a simple desktop. As in virtual worlds, the concept of *place* is also necessary, describing the spaces that physical structures shape, for example the halls of a museum, or the spaces defined by the application, for example the space that surrounds a showcase in one of those halls.

Similarly, the concept of *viewpoint* is also important here. However, unlike the virtual worlds, where the user's avatar can lie exactly on the point of view just by clicking on a button, in the AR paradigm, it is the users who move, by their own means, to such position, or turn their own head to look in that direction, so the result is rather an approximation to the fixed positions and/or orientations. Despite differences such as this, the model of the real space is, so far, very similar to the model of the virtual world.

Indeed, the physical devices are the link between the real world and the digital or virtual one, and at this point, the model relies on previous ideas already presented in [21], particularly on the environment model that was developed for the UsiXML language [22]. That previous model describes the hardware platforms that the users interact with in their physical environment, with special attention to the surfaces associated with such platforms, such as the screen of a monitor or a projection system, considered as *interaction surfaces*. The latter term was introduced in [23], and defined as any physical surface that can be observed (e.g. by looking at the screen) or that can be acted on (e.g. by using the mouse pointer), which allows interaction with the system, either visible or embedded (Figs. 10.6 and 10.7).

However, there is no doubt that the concept of interaction surface was introduced with the 2D GUIs in mind, which are presented in flat panel displays. Instead, there is a need to extend it beyond 2D and introduce the concept of *interaction volume*, more suitable for 3D UIs where interaction is not constrained to the plane but takes place in the 3D of real space. Thus, an interaction volume can be defined by the scope of a tracking system, but also a computer monitor can create a volume of interaction in front of it when presenting 3D stereo images, overlapping the virtual with the real dimensions (Figs. 10.6 and 10.7). In fact, the representation of synthetic 3D images in a viewport on the screen of a monitor also creates a volume of interaction

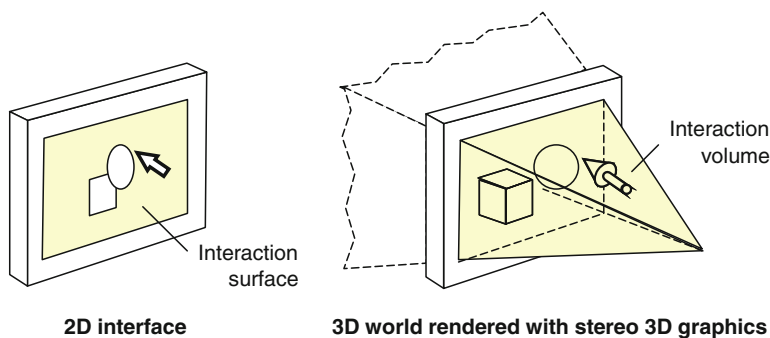


Fig. 10.6 Interaction surface and volume associated to a presentation device

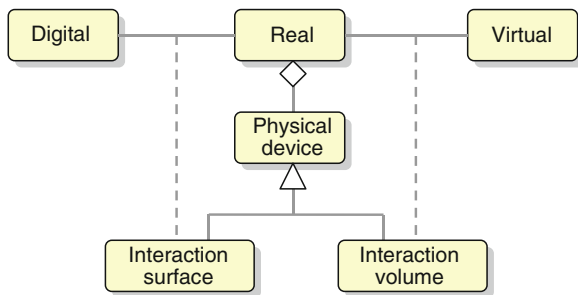


Fig. 10.7 Interaction surfaces and volume link real space to digital and virtual ones

behind that screen, although it cannot be reached by extending the hand. Moreover, these images are calculated on the basis of a point of view that has its correspondence with a point in the real world, so that the users will see the 3D effect if their position matches that point.

10.3.4 *Intra-, Inter- and Trans-Spatial Transitions*

The digital and virtual spaces are potentially infinite, only limited by technology, and the limits of the real space are beyond our world. However, digital and virtual content are usually bounded, as well as objects in the real world involved in the interaction. Nevertheless, the user's view of the previous spaces and their contents is often partial, incomplete. Some of the reasons for this are physical, related to human perception or presentation devices. Other reasons come from interface design, related to the tasks to be supported and the users' abilities to perform them. In any case, users must navigate to get a complete view, changing their viewpoint.

In the case of digital 2D space, proposed methods include specific diagrams to model these transitions, such as the Dialogue Interaction diagram in the IDEAS methodology, cited before. This is the *inter-window navigation*, but the concept of window should not be understood in the context of a window-based environment, but in a broader sense, for example screens instead of windows.

In the case of the virtual 3D space, even though the user can freely navigate through the virtual world, and thus explore it from very different locations, it is common to set a pre-defined set of viewpoints, and also an order for them, thus creating a guided tour through the world. This is the *intra-world navigation*, but is not the only one in such spaces. Beyond simple intra-world navigation, the immaterial nature of virtual environments allows playing with space in ways impossible in the real world. Thus, it is possible to travel from one world to another through a single door that connects the spatial dimensions of both worlds, which in InterSpace are called super- or subspace doors [24], but in Croquet are called portals or 3D hyperlinks [25]. Pre-defined viewpoints may well serve as entry points to the target world.

This is the *inter-world navigation*. The last type of navigation that is included in this meta-model is named *trans-spatial navigation*. It models transitions that occur in a space that is tied to another. This is what happens when the current viewpoint in a virtual world changes according to the movement of the user in the real world, thanks to the use of certain tracking devices. Another example, this time from an AR application, is the augmented dialogue that is described in [8], which binds the presentation of certain 2D GUIs to the arrival of a visitor at different points in the archaeological site.

10.4 The TRES-D Methodology

The process model of the TRES-D methodology is supported on the concepts and the terms provided by the space meta-model presented in Sect. 10.3, and the other two cited in Sect. 10.1, that is, the object and the interaction meta-models. This methodology derives from a careful study of previous approaches and one first attempt, named IDEAS-3D. It gives shape to a framework where different practices and tools for the general development of 3D UIs are gathered. The whole process is characterized by two main phases. The aim of the first one – the *previous study* – is to propose a solution to a given problem and, only if it is accepted, then the second phase follows – the *detailed study* – to complete the proposed design, and to implement and deploy it (Fig. 10.8).

In each stage, it is possible to prepare in advance some work of subsequent stages, so that the developer can foresee the problems that may need to be faced. In fact, the stage of solution proposal is meant to tackle in advance part of the work that would belong to the phase of detailed study. An agreement with the client mediates between both phases, as it is the client who has to approve the proposed solution. Once at the second phase, the design is addressed at two different levels, one detached from the implementation details – design I, and another one tightly related to that implementation – design II. A decision on which software and hardware will support the solution divides both design levels. In any case, both design I and II, and later the implementation stage, are divided in two parallel

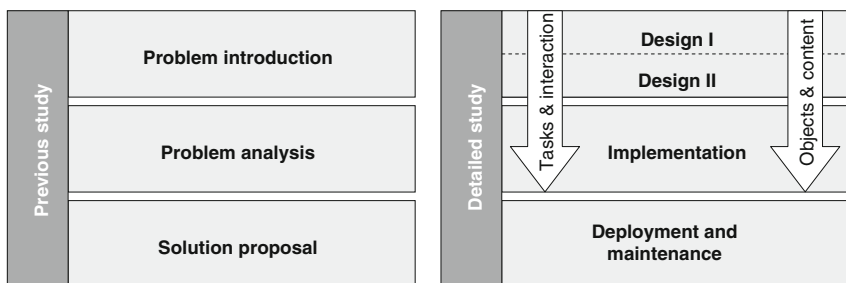


Fig. 10.8 TRES-D process model

activity chains. One of them is oriented to the interaction and the other one to the objects, following the proposed meta-models, translating the weight towards one or the other depending on each particular development, but in any case crossing both lines at several points of the development, specially due to their relationship with the third model, the space model. The last stage, deployment and maintenance, cannot be forgotten in the development of this kind of interfaces, which is quite often characterized by their non-conventional hardware.

10.5 Case Studies

Molina et al. [26] used three different projects to illustrate the structured development of 3D UIs that the TRES-D methodology proposes. This time, these projects are used to illustrate the meta-model presented in this chapter. One of these projects had the objective of creating a videogame for PC desktop that allowed the user to get into the game using 3D stereographics and a dataglove, and so the interest was to find a low-cost solution. The second of these projects pursued the virtualization of a block-based construction game in an immersive VR system, looking this time for the highest degree of naturalness. Finally, the third one was aimed at reproducing the UI shown in “Minority Report,” the movie, using all available means for this purpose. Figure 10.9 shows all of them.

Starting with the videogame, it added stereo 3D graphics to the popular Tetris game, and so it was named TTristereo. It involved three spaces, the digital, the virtual and the real one. On the one hand, game menus were designed in 2D space, using a diagram to specify inter-window navigation. On the other hand, the urn and the blocks of the game were modelled in the virtual 3D space, and later shown from a fixed point of view on a viewport in the application window. Next to the viewport, a 2D panel displayed information related to the game in that window. A widescreen TV served as an interaction surface when displaying menus of the game, and as an interaction volume when showing the urn and the blocks with stereo 3D graphics. Another volume of interaction was associated with the optical system that tracked the P5 glove. Both volumes must correctly overlap in real space to allow the player to grasp, rotate and drop the blocks in the virtual space.



Fig. 10.9 TTristereo (*left*), VRPrismaker (*centre*) and “Minority Report” interface (*right*)

In the case of the block-based construction game, its VR counterpart was named VRPrismaker, exhibiting a complete 3D interface. The playground, table, shelves, game box, blocks and the user's avatar were modelled in the virtual 3D space. In order to immerse users in the virtual world, they put on a stereo-capable head-mounted display and a couple of datagloves (the PinchGlove system). Attached to these devices, a set of sensors tracked user's head and hands, within the interaction volume associated to an electromagnetic system. Such movements were mapped to the user's avatar, resulting in a trans-spatial navigation. Because of that mapping, the range of the tracking system in real space had to be taken into account when modelling objects in the virtual space, so to allow the user to play with the blocks without leaving this volume.

Finally, in the case of the "Minority Report"-like interface, this represents a 3D UI that does not render 3D images, they are 2D ones; it is the actions that are performed in 3D space. Then, elements in the 2D image were designed with a component specification diagram, and the presentation device played the role of an interaction surface. To track and recognize user's hand gestures, an electromagnetic tracking system was used too, with its associated interaction volume. In this case, it was necessary to adjust the position of this volume in front of the interaction surface, so that the user could correctly point to the images displayed.

10.6 Conclusions

This work represents an attempt to formalize 2D and 3D spaces in the context of 3D UI development by introducing a meta-model. A key feature is the distinction between digital, virtual and real space, based on a new continuum introduced here as an extension to Milgram and Kishino's Reality-Virtuality. This meta-model also contributes to define a language for a proposed methodology, named TRES-D, which has been described too, along with some case studies. However, TRES-D does not achieve the automation that was aimed when the first proposal, IDEAS-3D, was presented. Further work on the proposed meta-models must be done in order to address that goal, for instance merging them with UsiXML models so to benefit from its established model-based process.

Acknowledgements This work has been partially supported by the Ministerio de Educación y Ciencia of Spain (CICYT TIN2004-08000-C03-01) and by the Junta de Comunidades de Castilla-La Mancha (PAI06-0093).

References

1. Cognetics Corp. The LUCID Framework. URL: <http://www.cognetics.com/lucid>
2. Bardon D, Berry D, Bjerke C, et al. (2001) Crafting the Compelling User Experience Using a Methodical Software-Engineering Approach to Model Users and Design. Tutor. URL: http://www-3.ibm.com/ibm/easy/eou_ext.nsf/publish/1650

3. Lozano M (2001) A Methodological Approach for the Specification and Development of Object Oriented User Interfaces. Doctoral Thesis. Univ Politéc de Valencia, Valencia, Spain
4. Fencott C (1999) Towards a Design Methodology for Virtual Environments. UC DIVE Workshop, York, UK
5. Sánchez MI, de Antonio A, de Amescua A (2005) SENDA: A Whole Process to Develop Virtual Environments. In: Sánchez MI (ed) Developing Future Interactive Systems, Idea Group
6. Molina JP, González P, Lozano MD, et al. (2003) Bridging the gap: developing 2D and 3D user interfaces with the IDEAS methodology. In: Jorge J, et al. (eds) DSV-IS: Issues in Designing New-generation Interactive Systems, LNCS, Springer, Berlin, pp. 303–315
7. Molina JP, Vanderdonckt J, Montero F, et al. (2005) Towards Virtualization of User Interfaces based on UsiXML. 10th ACM Web3D Symposium, Bangor, UK, 169–179
8. Granollers T, Lorés J, Raimat G, et al. (2002) Vilars, un Nuevo Modelo de Diálogo Aplicando Realidad Aumentada. III Cong. Interacción, Leganés, Spain, 276–279
9. Milgram P, Kishino F (1994) A taxonomy of mixed-reality visual displays. IEICE Trans. Inf. Syst., Special Issue on Networked Reality, E77-D:12:1321–1329
10. Molina JP, García AS, Martínez D, et al. (2006) An Interaction Model for the TRES-D Framework. 13th IEEE MELECON, Benalmádena, Spain
11. ClockWise Technologies Ltd. Win3D. URL: <http://www.clockwise3d.com>
12. 3DNA Desktop. URL: <http://www.3dna.net>
13. Y. Fei. Cube Features. URL: <http://www.infinite-3d.com/cube.html>
14. The SphereXP. URL: <http://www.spheresite.com>
15. Sun Microsystems. Project Looking Glass. URL: http://www.sun.com/software/looking_glass
16. Compiz Fusion. URL: <http://www.compiz-fusion.org>
17. van Dam A (1997) Post-WIMP User Interfaces. Commun. ACM, 40:2:63–67
18. Trevisan DG, Vanderdonckt J, Macq B (2004) Conceptualising mixed spaces of interaction for designing continuous interaction. Virtual Real J., 8:2:83–95
19. IBM. RealPlaces Design Guide. URL: <http://www-03.ibm.com/easy/page/580>
20. Roberts D (2000) RealPlaces, 3D Interfaces for Office Applications. TFWWG Workshop, Biarritz, France
21. Molina JP, Vanderdonckt J, Gonzalez P, et al. (2006) Rapid Prototyping of Distributed User Interfaces. In: Computer-Aided Design of User Interfaces V, ISS, Springer, Berlin, pp. 151–166
22. UsiXML.org, User Interface eXtensible Markup Language. URL: <http://www.usixml.org>
23. Coutaz J, Lachenal Ch, Dupuy-Chessa S (2003) Ontology for Multi-surface Interaction. 9th IFIP INTERACT, Zurich, Switzerland, 447–454
24. Leftwich J (1993) InfoSpace: A Conceptual Method for Interacting with Information in a Three-Dimensional Virtual Environment. 3rd Int. Conf. on Cyberspace, Austin, USA
25. The Croquet Project. URL: <http://croquetproject.org>
26. Molina JP, García AS, Martínez D, et al. (2006) The Development of Glove-Based Interfaces with the TRES-D Methodology. 13th ACM VRST, Limassol, Cyprus, 216–219