

Designing a interface for virtual reality and haptic application prototyping: a case study

Joan De Boeck^{*} Emanuele Ruffaldi[‡] Marcello Carrozzino[‡] Chris Raymaekers^{*}
Karin Coninx^{*} Massimo Bergamasco[‡]

(^{*})Hasselt University, Expertise centre for Digital Media and transnationale Universiteit Limburg
Wetenschapspark 2, B-3590 Diepenbeek, Belgium

([‡])PERCRO, Scuola Superiore Sant'Anna, Via Rinaldo Piaggio, 34,
56025 Pontedera, Italy

{joan.deboeck,chris.raymaekers,karin.coninx}@uhasselt.be
{e.ruffaldi,m.carrozzino,bergamasco}@sssup.it

Abstract

It may sound obvious that computer interfaces strive to adopt the same patterns of learning as we know from our daily world. In particular we refer to the enactive knowledge, acquired by 'doing', such as "driving a car", or "playing an instrument". However, even in highly interactive 3D Virtual Environment applications, the enactive approach is still not common. This paper shows how the adoption of a hardware standard, and the use of high level interaction diagrams may support the easier creation of such an interface.

1. Introduction

Technical problems, as well as user interface design problems, may be the main reasons why effective enactive interfaces are difficult to develop. First of all, a rich and multimodal communication channel between human and machine is required for this kind of interfaces, very often resulting in the use of specialized (and hence less standardized) hardware, such as haptic devices and trackers. Moreover, when a specific hardware setup may be operational, the designer has the possibility to choose between dozens of possible interaction techniques to perform a certain task. Some techniques may carry similarities with common techniques in the real world, but they also can act as some kind of "magic" within the virtual world. The most reliable method to know which technique may be suitable in a given situation still consists in testing the proposed setup. This implies that a working prototype must be implemented, which may result in long development times and several iterations of the system. Designing an enactive interface clearly is not a

simple process. In this paper we therefore describe the design of an application as a case study, in order to prove how hardware standardization and the use of high level interaction diagrams may facilitate the creation of this kind of interfaces.

2. Existing Application

Virtual Prototyping exploits VR technologies to let the users perceive a non physically existing object as real as possible. The aim is to allow the user to perform operations directly on the virtual prototypes with operative modalities almost identical to those used in the real world. The automotive industry is the sector where these technologies have been introduced with the best success, being used to verify complex assemblies of mechanical groups designed by different workgroups[1], to verify the ergonomics of the vehicle interiors, to perform analysis in the recycling stage and to verify the disassembly procedures of the product[2].

PERCRO, in collaboration with Piaggio S.p.A. developed the Interactive 3D Visualization System (IVS) [3], a Virtual Prototyping system

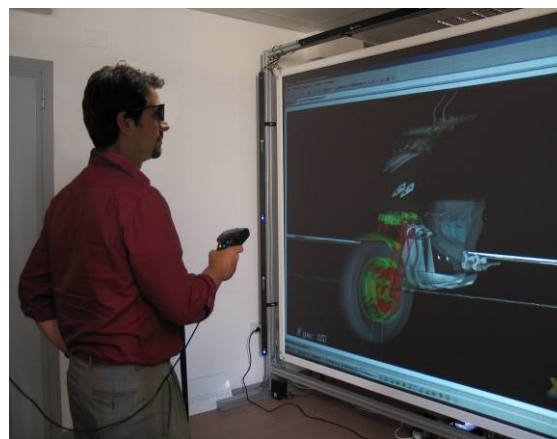


Figure 1: The IVS System

dedicated to the stereoscopic visualization of a 3D model of a motorvehicle in order to verify the final mechanical assembly on a PowerWall-like device. All the software functionalities have been realized with XVR [4], a developing tool for VR applications realized by PERCRO in collaboration with VRMedia.

The system allows to interactively explore the assembly and to manipulate its components. It is possible to select the elements of the assembly and to move them with operations of roto-translation, to temporarily hide them in order to assist the vision of the assembly interior, and to perform measurements of distance between points of the assembly. Two kinds of interaction controllers have been used: a sensorized wand (provided with a joystick and an additional set of buttons associated to various functionalities related to movement, selection etc.) or a glove coupled to a 6-DOF wrist sensor. In the latter case the functionalities are associated to the postures of both hands. In both cases the ray-casting interaction metaphor was chosen for the selection of subcomponents. The two hands are used in different ways: the dominant hand is associated to action tasks (like selecting, moving. etc.) while the non-dominant hand is associated to option tasks (like switching interaction modality, active status etc.). The user's head is also tracked in order to produce a correct dynamic perspective.

Although the overall evaluation was evaluated positively, several issues arose about the interaction metaphors implemented in the system. Although Ray Casting is one of the easiest and most common selection metaphors, it suffers from being less accurate with distant objects because of the relative sensitivity of the rotation of the ray. This behavior discourages users to directly manipulate distant objects, and induce them in performing a preliminary manipulation of the model to bring closer the desired component, in order to allow an easier and more precise manipulation. Another drawback is related to occluded objects which can be accessed only rotating the main model or removing occluding subcomponents.

As the design and evaluation of the initial application was a lengthy process in which each change must be written in code, we have chosen to use this particular application as a case study to evaluate the proposed platforms that may help to facilitate the development and evaluation of an enactive interface.

3. Used Platforms

3.1 VRPN, Virtual Reality Peripheral Network

The haptic rendering framework used for this experimentation is based a new module for the Virtual Reality Peripheral Network protocol. Most of Haptic Rendering libraries are based on a local

communication with the rendering library or with simple communication interface with a remote haptic server. Given the required flexibility we have chosen to use the VRPN protocol for providing remote haptic rendering. The contribution of this paper is related to the generalization of the VRPN for multiple contact points and for a flexible approach in the geometry sharing between elements. The use of a remote protocol is extremely useful for also implementing a visual debugging tool of the haptic scene.

3.2 NiMMiT, Notation for Multimodal Interaction Techniques

The creation of (multimodal) user interaction in a 3D environment is a time-consuming and expensive process. Solutions have to be implemented using programming code, and as the acceptance of a multimodal interaction paradigm can not be fully

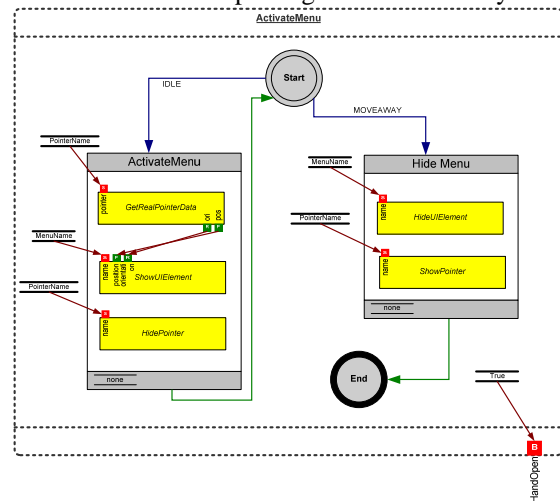


Figure 2: NiMMiT Diagram of the Aperture Selection

predicted in advance, the result often must be evaluated and adapted afterwards. To simplify this process, EDM has developed NiMMiT [6], a graphical notation intended to *describe* multimodal user interaction, rather than *implement* it. The notation can also be used to capture performance data during a user experiment, providing formal data for statistical analysis during the evaluation phase [7]. An example of a NiMMiT diagram is given in Figure 2. NiMMiT inherits the formalism of a state chart, and defines 'events' as a result of the user's actions within the virtual world. In each state, the interaction responds to a given set of events. Dependent on the events occurred, a given set of 'tasks' (grouped in a 'task chain') is executed, executing what is necessary in this particular phase of this interaction technique. For a more comprehensive explanation and more examples of NiMMiT, we refer the interested reader to [6][7][10]. In the remainder of this paper we will explain some simple diagrams, as well.

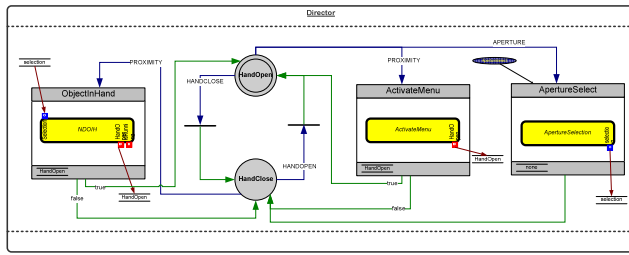


Figure 3: Overall NiMMiT diagram with hierarchical use of the components

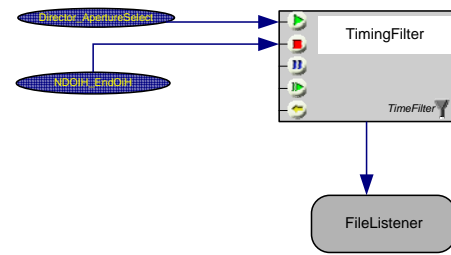


Figure 4: NiMMiT Filter Diagram

4. Case Study: Virtual Prototyping

4.1 Input devices

The proposed experimentation has been prepared with the use of the GRAB haptic interface, a device with two point of contacts that can be used for the experimentation of two hand interaction.

4.2 Interaction Metaphors

4.2.1 Designing Interaction

In the proposed virtual prototyping application, users may have two approaches in order to explore the model. Either they may to explore the entire model, and navigate through and around the model, or they want to ‘grasp’ certain parts of the model and inspect them separately. The choice between the different inspection techniques can be done using a 3D floating menu.

The menu is activated as described in [8]. By bringing the user’s non-dominant hand close to the dominant hand, the menu appears at the position of the user’s pointer, so that they can quickly interact with that menu. This approach has proven to be very intuitive, as the user’s proprioceptive knowledge is exploited to activate and operate the menu. For the interested reader, we have shown the diagram of this interaction in Figure 3.

For the ‘picking’ and inspection of individual parts of an assembly, we adopted the ‘Object in Hand’ metaphor [9], combined with the aperture selection [12], as proposed and evaluated in [11]. Using the non-dominant hand, an ‘aperture’ is moved onto the screen, highlighting the object that is in the aperture. When the user decides that the desired part is highlighted, this can be ‘grasped’ by bringing the fist of the non-dominant hand close to the dominant hand. The object then pops out of its context and comes to a central position closer to the user. By moving and rotating the non-dominant hand, the object is affected similarly, allowing the user to inspect it.

As NiMMiT allows hierarchical reuse of diagrams, the combined action of the different parts of the interaction (menu activation, object selection, object grasp and release) can be described by a diagram as well. This is depicted in figure Figure 3. Here we distinguish two separate states, reflecting

whether the user’s non-dominant hand is open or closed. Dependent on the active state and the event occurring, one of the three task chains is executed. Each task chain contains only one (hierarchical) task, executing the particular NiMMiT diagram.

For the navigation, which can be used for the exploration of the entire model, we have chosen to use the ‘scene in hand’-metaphor [13], in which force feedback with the dominant hand, simulating the surface of a sphere using force feedback features, may improve the interaction.

4.2.2 Evaluating Interaction

For the evaluation of the interaction metaphors, we can make use of the Probe and Filter primitives of NiMMiT [7]. For instance, we can easily capture the time that a user needs to select an object and subsequently grasp it. This may be useful, for instance to evaluate whether the combination of the selection technique and the grasping technique is efficient enough. The capturing of the data can be implemented by adding a probe to the ‘ApertureSelect’ taskchain and another probe at the ‘ObjectInHand’ taskchain. With the probes, we can receive raw info about the execution of the diagram at these points. Next, in ‘filter diagram’, the probes can be connected to a ‘Filter’, processing this raw data. In Figure 4, we can see how the data of the two probes is routed to a timer-filter, using the data of the probes to simply start or stop a timer. The result of the time measurement is then routed to a FileListener, storing the results in a file where it can be used for any statistical analysis afterwards.

5. Discussion

After the time we needed to write an initial interface to VRPN to operate the particular force-feedback device (which only needed to be done once), we could start using VRPN for our project. It turns out that the development of the interface at the labs of EDM, without the availability of the specialized force-feedback devices that are to be used at PERCRO, caused no significant problems. As VRPN acts as a transparent layer between the application and the hardware, only a configuration file at the server -or- at the client side has to be adapted in order to change the input devices.

Designing NiMMiT diagrams is a significant improvement over manually implementing each

interaction technique using programming code. First of all, NiMMiT gives the designers an easy to read, easy to understand diagram, allowing them to discuss the proposed solution with other colleagues, when necessary. Next, the key actions in an interaction diagram are the ‘tasks’. Dependent on the domain on which NiMMiT is used, several frequently used tasks may be predefined, such as in our case ‘moving an object’, or ‘calculating collision detection’. This speeds up the design a lot, but because designers still have the possibilities to write their own specific tasks (‘custom tasks’), NiMMiT’s high-level approach does not limit the power we have in low-level solution. In this case study, only one custom task (for the sphere navigation) had to be coded and one task could be reused but had to be adapted from a previous project.

Finally, the addition of features to capture ‘real user data’ using ‘probes’ and ‘filters’ gives the opportunity to quickly add code for interim user experiments, facilitating the iterative aspect of user interface design.

5. Conclusion and Future Work

In this project paper, we described the redesign of an existing Virtual Prototyping application, aiming to improve the enactive aspects of the interface. We focused on two platforms, VRPN and NiMMiT, both facilitating the development process. VRPN makes abstraction of the concrete hardware device, avoiding the need to have specific and expensive hardware to be available at all times. NiMMiT on its turn allows a designer to easily describe the user interaction and adds additional features to capture real user performance data for an evaluation of the designed interface.

We can conclude that our approach improves the flexibility of Interface Design, and especially addresses the experimental approach of creating enactive interfaces by facilitating the iterative process of choosing devices, designing interaction techniques, evaluating the result and iterating over again. Taking into account the possible reuse of previous investments, we can state that this strategy allows an interface designer to deliver an interface prototype in much less time than by coding it manually.

Concerning the case study described in this paper, after extending the features of the application we want to conduct a formal usability experiment, evaluating the different interaction techniques. As NiMMiT is currently still running on top of an experimental research framework (VRment), it may be valuable to integrate it in a more powerful environment such as XVR, as well.

6. Acknowledgements

Part of the research at EDM is funded by EFRO (European Fund for Regional Development), the Flemish Government and the Flemish Interdisciplinary institute for Broadband Technology (IBBT).

Part of the research at PERCRO and EDM has been funded by European Community in the Enactive NoE

Reference

- [1] Niesen M. R. and Luecke G. R. “Virtual dynamic prototyping for operator interface design”. Proc. Of IEEE ro-Man, September 27-28 1999.
- [2] Roy U, Bharadwaj B, Kodkani SS, Cargain M. “Product development in a collaborative design environment.” Concurrent Engineering Research And Applications 1997
- [3] Marcello Carrozzino, Andrea Baroni , Franco Tecchia, Fabio Rossi, Massimo Bergamasco :”Interacting with complex virtual models: a suitable metaphor for Virtual Prototyping”, proceedings of Enactive Conference 2005
- [4] <http://www.vrmedia.eu>
- [5] Taylor II, R. M. (August 2006). VRPN virtual reality peripheral network. <http://www.cs.unc.edu/Research/vrpn/index.html>.
- [6] Vanacken, D., De Boeck, J., Raymaekers, C., and Coninx, K. (2006). NiMMiT: A notation for modeling multimodal interaction techniques. In Proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP06), Setbal, Portugal.
- [7] Coninx, K., Cuppens, E., De Boeck, J., and Raymaekers, C. (2006). Integrating support for usability evaluation into high level interaction descriptions with NiMMiT. In Proceedings of 13th International Workshop on Design, Specification and Verification of Interactive Systems (DSVIS’06), Dublin, Ireland.
- [8] De Boeck, J., Raymaekers, C., and Coninx, K. (2006d). Exploiting proprioception to improve haptic interaction in a virtual environment. In Presence: Teleoperators and Virtual Environments, volume 16. The MIT Press.
- [9] De Boeck, J., Cuppens, E., De Weyer, T., Raymaekers, C., and Coninx, K. (2004). Multisensory interaction metaphors with haptics and proprioception in virtual environments. In Proceedings of the third ACM Nordic Conference on Human-Computer Interaction (NordiCHI 2004), Tampere, FI.
- [10] De Boeck, J., Raymaekers, C., and Coninx, K. (2006c). Comparing NiMMiT and data-driven notations for describing multimodal interaction. In Tamodia 2006, Diepenbeek, Belgium.
- [11] De Boeck, J. A User and Designer Perspective on Multimodal Interaction in 3D Environments, PhD Thesis, Diepenbeek, 2007
- [12] Forsberg, A., Herndon, K., and Zeleznik, R. (1996). Aperture based selection for immersive virtual environments. In Proceedings of UIST96, pages 95–96.
- [13] Ware, C. and Osborne, S. (1990). Exploration and virtual camera control in virtual three dimensional environments. In Computer Graphics, volume 24, 1990.