Managing Information of CAD Projects in Virtual Environments

Luciano Soares, Felipe Carvalho, Alberto Raposo, Ismael Santos

Tecgraf – Computer Graphics Technology Group
Department of Computer Science
PUC-Rio – Pontifical Catholic University of Rio de Janeiro, Brazil

{lpsoares,kamel,abraposo,ismael}@tecgraf.puc-rio.br

Abstract

CAD models are an important step in several engineering projects. In order to maintain consistence, this information has to be attached to the 3D model, which is usually a difficult task. An even more important task nowadays is to set notes in the parts inside a model. Some techniques for setting annotations were proposed and developed, by integrating into a collaboration tool in order to make sure the procedure can be done in collaborative environments. Two representation for notes are explained here, one just for textual information which display the complete information in the virtual environment and other that uses icons based in GUI windows desktops to attach any kind of data file to an virtual object.

Keywords: Virtual Reality, CAD models, Collaborative Virtual Prototyping and Design review.

1. Introduction

Nowadays the engineering projects are using CAD (Computer-Aided Design) tools not just to plan the construction, but for operation queries and sharing information about the specifications to any other user working in the project. In order to do this, additional information is attached to the scene. Most of the time this information is linked to a particular object in the scene, although in some rare situations we need an annotation in the scene not connected to a particular object, but to a particular coordinate position in the space. This management of information gets more complicated when using a virtual reality environment as an advanced form of viewing CAD models in real time [1].

This information can be managed from the beginning of the project, when material and conditions of a specific system of a project is defined or during the execution when changes need to be updated. This information is managed by a database that could be synchronized to other system [2]. For instance, Figure 1 presents information that was attached to a gas tank, for operation of production process. This information is necessary when engineers are discussing about some point in the project and very often they are moving around the virtual environment and need a fast update of the information.

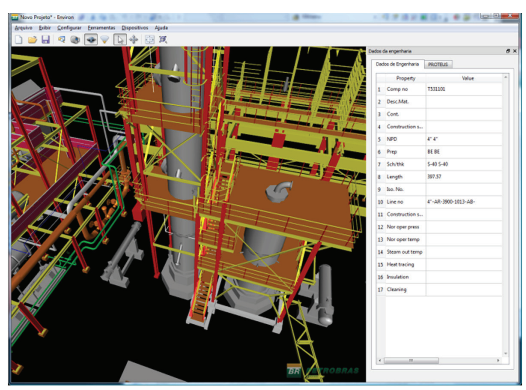

Figure 1. Example of Engineering Information of an Industrial Plant

Other possible information integrated in the CAD/VR data is an annotation. Different from the previous information, the annotation should be easily accessible and visible. Most of the time user wants to see the information together with the geometry of the model, since it is probably due a modification or status in the project that need to be monitored, then a kind of billboard is produced always facing the user. Proposals for annotations were developed and evaluated and will be further discussed in the body of this document.

The last kind of coupled information we are going to discuss is the use of 3D icons to represent information which is in a completely different file format from the CAD source, and can be accessed like a hyperlink, allowing the user to go back and forth in this data information from the 3D environment to the 2D desktop application.

The following section presents some related work in the area of managing CAD information. Section 3 discusses the virtual annotations that were developed in our research. Section 4 presents the proposed ways on how to manipulate annotations in virtual environments explaining the ways it can be used to control and access the information we need. Section 5 presents the issues related to a collaborative system when the information discusses in the previous section need to be worked with other people at the same time. Conclusions and future work follow in section 6.

2. Related work

There are some coupling ways in order to integrate CAD and VR [2], both systems needs geometry and integrated information. These systems could be connected by means of gateways from CAD to the VR model. In this process, CAD models are converted to a format suitable for VR. Other possibility is a common file format for both CAD and VR models like the XMpLant [3]. CAD and VR can also be connected by an API like the OMG CAD services interface [4], which is a CORBA-based interface standard, but the current limitation of CAD services interfaces is that it has not been widely adopted by the industry. Finally it is possible the integration in one process, where the VR system is integrated into the core of the CAD system [1].

Annotations are resources available in some CAD tools like the AutoCAD that has new features in the current release for annotations [5]. But these tools are not aimed to virtual reality, and then the annotations will not be useful in a VR environment, and even worse in a immersive environment.

Until now, the research around annotation inside VR has been based on the creation and placement of text, audio and drawing information. The creation of audio annotations was implemented in [6] to allow posterior playback inside the virtual environment. The use of text is found in some systems [7] [8] [9] and were implemented based on variations of the billboard visualization technique. The main difference between them and other approaches are the placement control (static/dynamic) of the annotation. Drawing information created by the use of free spatial movements (e.g. 2D movements on a surface or 3D movements in a space with or without force-feedback) is also founded in some works [10] [11].

3. Virtual Annotation

The interaction with VEs can be enriched if the user is able to insert, remove or modify some features of the environment and its 3D objects (dimensions and coordinates). These transformations can modify individual features of some objects on the environment, however, in some cases, they can not modify the scenario. The main idea of the approach presented in Virtual annotations are annotations created for virtual environments. It should be able to be created for any object in the 3D scene. These annotations in a VR system are usually created based on the user view, not limited in a list of the scene objects. These annotations are used in projects that need constant review and the user is analyzing the scene from different positions.

Figure 2 presents a user that inserted an annotation in a region of pipe. Note here that

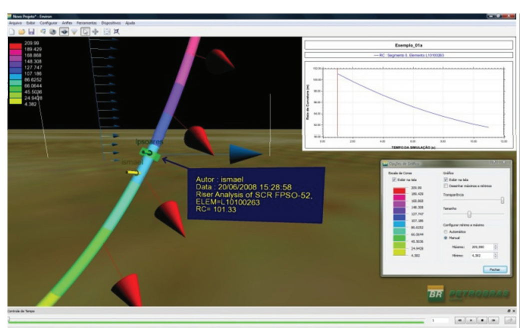

Figure 2. Annotation inserted in the environment

since the software allows to split the pipe in sub-regions annotations could be done in an exact position.

3.1. Annotation Fields

The amount of fields in an annotation should be enough to display the textual information necessary to explain what we desired. After some tests the following fields were defined:

Author: defines who created the note. This is automatically defined with the user name acquired from the operating system, but can be changed to something more human readable like the user full name;

Date: The date when the note was created or updated. This field can be changed, since in some cases the annotation was defined in a prior meeting and the user wants to use that date;

Description: A free text space, where the user can write anything. No special resources, such as bullets or numbering are allowed here, since it was making the notes more confuse.

Beside these fields some resources were incorporated in order to create some flexibility for the users that are working with the annotation:

Display Author and Date: Most of the time people is not concerned who wrote the note or when it was written. Then the information is by default turned off, but if the user wants to display in the 3D environment who created it and when an annotation was created it can be switched on;

Text and Background Color: Depending on the environment the black and white is not a good option to display the note, in these cases, a specific color can be chosen in order to produce a better contrast;

Size: The 3D annotation has a specific size that is proportional to the objects, but the user can customize the annotation to any size.

The center of the annotation is created in a position pre-configured at one unit distant from all the axes. But it is not always a good position, and then the annotation can be moved as a normal object. Rotation is not allowed since the text is always facing the viewer. Figure 3 presents an example of an annotation test in an oil platform. It is also depicted that the annotation is part of the scene graph but can be accessed individually.

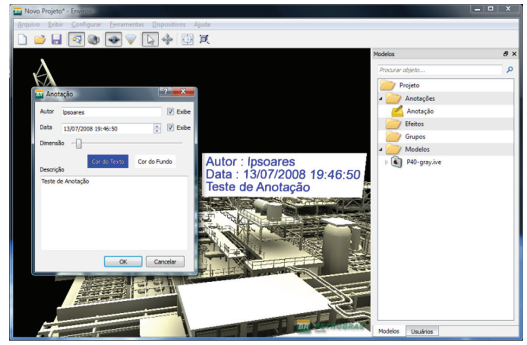

Figure 3. 3D Annotation Dialog Window

Other representation that is being under development is the use of icons to represent any kind of annotation, or link. In this case an annotation can be more complex information made from desktop applications such as Microsoft Word or PowerPoint file for instance. Representation of annotations with icons helps the user know what is linked. For instance Figure 4 presents a link to a text produced in the Microsoft Windows Notepad. The selection of such kind of annotation launches the appropriate application for the indicated icon.

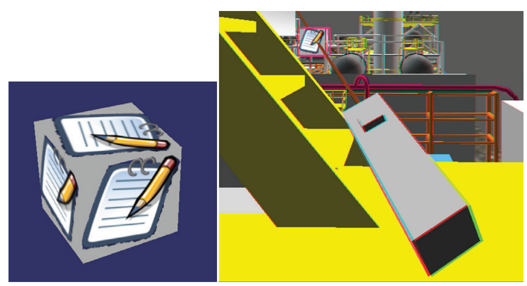

Figure 4. Annotation with a Cubic Icon

In the case of the use of an annotation with an external file, it creates a change of context that is something that should be kept as simple as possible.

4. Manipulating Annotations in Virtual Environments

In order to select objects an arrow representation is used, like a 2D mouse pointer. This arrow is positioned based on a simple ray casting algorithm that is generated depending on the degree of immersion desired. In the case of a desktop computer, the viewing direction and a click with the mouse on the screen determines the arrow position and direction. The source is defined in the virtual camera and the direction based on the coordinates on the screen. Other possibility is the use of a tracking device, such as electro-magnetic or optical tracking systems. In this case the tracking will work as a wand, allowing the creation of the arrow in any direction, and it is quite important for immersive environments. Tracking user position and orientation is fundamental in the virtual reality system proposed. Although it is possible to track and interact with the viewing direction and mouse clicks, it is limited (it is not a 6 Degree of Freedom device).

The ray casting algorithm is used to detect the first object intersected by a ray. The tip of the arrow will intersect the object. If no object is hit by the ray, then no arrow is drawn. The text is always facing the observer, but it does not rotate in the arrow axis orientation. It helps identify the other user's orientation.

There are some devices in the market that can be used as a pointing device, but the Nintendo Wii controller, known as Wiimote [12], has presented several advantages. First it is wireless, using Bluetooth that is based in a protocol broadly supported by several computers. It has also an accelerometer inside that can be used to detect the speed and orientation. Besides, the integrated speaker and rumble functionality give a good feedback for the user about the interaction procedures. One drawback is the limited capacity to track position and orientation. In order to overcome this problem an optical tracking was used together, then retro-reflexive spheres were installed in the Wiimote, as presented in the Figure 5, allowing the tracking solution to calculate the position and orientation of the device.

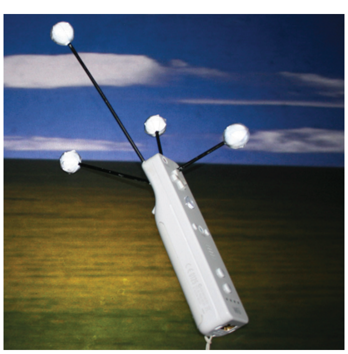

Figure 5. Wiimote with Retroreflexive Spheres

In Figure 6 we can see a CADmodel of a ship and a 3D model of a Wiimote device interacting in the scene. The pointing device has a line with source in the model and long enough to reach any object. The line is drawn in red to give an idea of a laser beam although we would not see a laser beam in a dust free environment, the line and ray casting algorithm is started just when is turned on with the Wiimote button "A" is pressed. When the ray hits an object the bounding box of this object is drawn in white just to inform you are selecting the right object, if you want to do some kind of interaction you have to press a button. In our case the Wiimote button "B" is defined to manipulate the object (rotate and translate) and the button "+" is used to open or create an annotation.

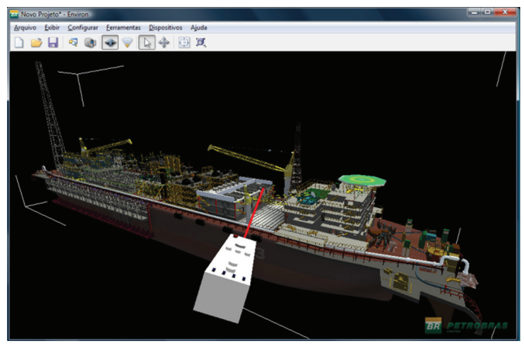

Figure 6. Wand manipulation

This solution can be used from a fish-tank VR environment to a semi-immersive solution. For instance the user at Figure 7 is navigating and interacting in a desktop display. The stereo used in anaglyph and the head a wand is tracked by an optical solution.

Figure 7. Person interacting in a virtual environment with head-tracking and wand

Figure 8 presents a user interacting with the same devices in a large display wall. The use of the same devices in both environment works helps the user to control the system. The change from one architecture to another should have the minimum impact for the user.

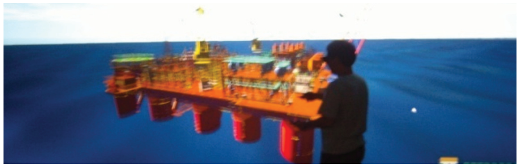

Figure 8. Interacting in a Display Wall

In a semi-immersive environment the use of tracking is mandatory, since a standard mouse will not be able to track all possible positions. Figure 9 show a user interacting in the same model and with the same device.

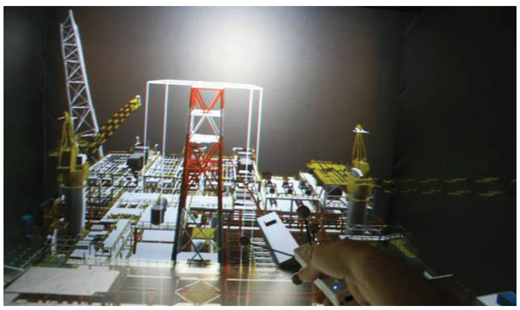

Figure 9. Selecting Object and adding notes in a Semi-immersive System

The precision of the tracking system appeared not be so important for the user, the main problem with the tracking is when it gets lost. This is reported as very uncomfortable.

5. Sharing a Virtual Environment

The use of avatars is common in shared virtual environments [13], but in simulation that users are interacting like their CAD application, avatars could confuse some users. In this case a 3D pointer enables users from different points of view to understand what each one is looking at. In such collaborative environment, the annotation feature is allowed for all the participants too.

Figure 10 presents a simple scene where two users are interacting in the same area, because of the 3D pointer it is possible to know where each user is gazing or pointing their interacting devices. Colors are defined to each user for each system. In our case the green arrow is always used for the local user, this avoids confusions since you know the color you are controlling is the green. In a simple version a text with the user name is presented with the arrow.

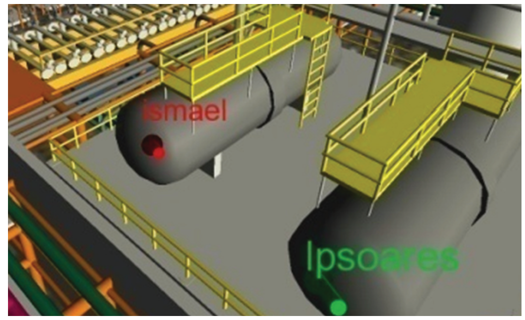

Figure 10. 3D Cursors in Collaboration Tasks

In a collaborative environment it is necessary to have a better communication between the participants, then the system was coupled with a video conference software in order to enable talking and visualizing the other users. Figure 11 presents two users sharing a virtual environment with the collaboration software in the side of the application. The system was tested in Microsoft Windows XP and Vista. In this example a small picture of the users is also used together with the arrow and text.

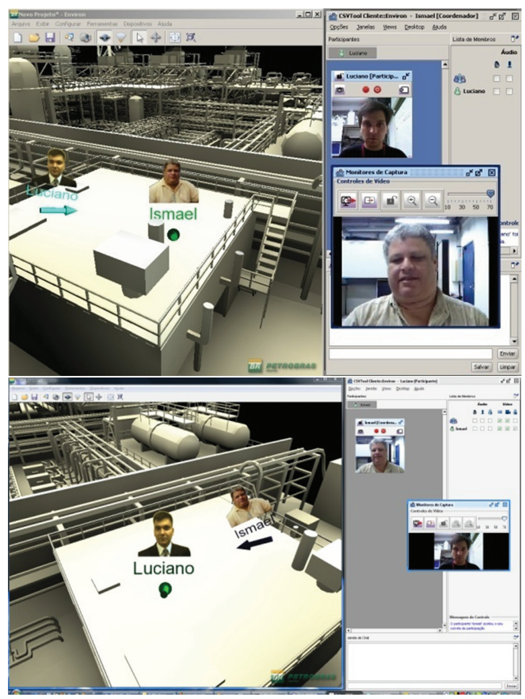

Figure 11. Collaboration in a 3D Environment with Video Support

5.1. Synchronizing Environments

Before proposing a customizable solution for Synchronization, it should be flexible enough to provide the users with the widest range of controls, supporting upgradability and being easy to understand, besides the fact that it should be fast enough to present an environment completely consistent. Due to these points a scripting language was chosen to synchronize the environment. Figure 12 presents an example of synchronization between two processes in the same computer.

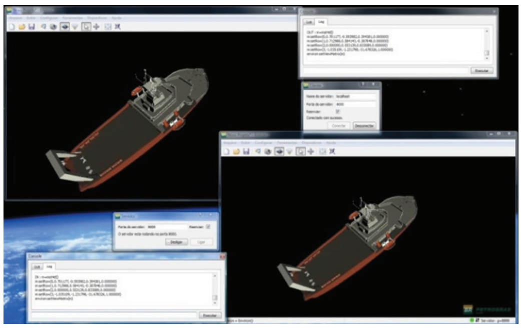

Figure 12. Script Synchronization

The language chosen to synchronize the environment in the collaborative session was LUA [14] since it is quite simple and light, it uses very small amount of CPU resources to run. We decided to use TCP/IP to assure a consistence in the environment. This then can be used in a system with the users distributed geographically or in a cluster system [15]. Code 1 shows some simple commands used to configure the pointers and can be used in a network communication.

```
instance = Prog();
wand = instance:getWand()
p1 = wand:getPointer()
p1:show()
p1:setPosition(-89.6,-96.9,-261.2)
p1:setOrientation(0.15,0.11,0.2,0.85)
p1:setText("User1")
p1:setPhoto("user1.jpg")
p2 = wand:createPointer()
p2:show()
p2:setPosition(-97.3,-100.8,-265.1)
p2:setOrientation(0.5,-0.3,-0.4,0.8)
p2:setColor(1,0,0)
p2:setText("User2")
p2:setPhoto("user2.jpg")
```

Code 1. Example of Commands Used

Of course the interpretation of the script language has some overhead, but in our usage we were not able to detect any big latency in the system.

6. Conclusions and future work

Work with CAD models in VR environments is a task that currently requires significant effort. The retrieve of additional information of the project is already a problem since the conversion from CAD to VR is not well defined in any format.

The use of annotations created the opportunity for engineers to include information inside the virtual world. That mechanism attempts to help the understanding of the virtual content by making those additional information visible by the use of objects with particular visual feedbacks (billboards and 3D icons). An initial empirical evaluation under development with few users is helping us to detect parameters for the annotations, until now we were able to detect that the parameters presented are enough for most of the annotations, the users that we tested are giving good feedbacks since no one requested net fields up to now, but it is still being evaluated with new possibilities. We tested to put the text in the annotation always in front of any object independent if there is something on front of it, but it created some problems since two annotations could be at the same point in the display making it unreadable.

The use of 3D icons for annotation creates other possibility to link information in the 3D model creating a "bridge" between the virtual environment and the desktop's application features for the creation of more complex data. As a future work we are going to develop the 3D icons in order to use the same icon used in the Microsoft Windows interface.

One current problem is when a user wants to write an annotation in an immersive environment. In our semi-immersive cave-like system [16] we included an area to put a keyboard as presented in the Figure 13, but in a system where it is not possible to put a keyboard, we are working in a virtual keyboard that can be used with the tracking device.

Figure 13. Keyboard and Mouse on the Semi-Immersive System

7. Acknowledgments

The authors would like to thank Petrobras and Tecgraf/PUC-RJ for their expressive support. Tecgraf is a laboratory mainly funded by PETROBRAS. Most of the evaluations and development of this research weres made over the Environ software [16].

8. References

[1] J. Berta, "Integrating VR and CAD," *IEEE Computer Graphics and Applications*, vol. 19, no. 5, p. 14–19, 1999.

[2] M. Vahl and U. von Lukas, "Integration of CAD and virtual reality bases on OMG's CAD services interface," in *European Concurrent Engineering Conference 2003*, Plymouth, 2003, pp. 54-61.

[3] Aveva Group. (2007) VNet - XMpLant. [Online]. http://www.aveva.com/products_services_aveva_plant_vnet.php

[4] Object Management Group. (2007, January) OMG Computer Aided Design Service, V 1.2.. [Online]. http://www.omg.org/technology/documents/formal/cad.htm

[5] AUTODESK INC., *AutoCAD 2008 New Features Manual.*: Autodesk, 2007.

[6] R. Harmon, W. Patterson, W. Ribarsky, and J. Bolter, "The Virtual Annotation System," in *Proceedings of IEEE VRAIS'96*, 1996, pp. 239-245.

[7] F. Ritter, H. Sonnet, K. Hartmann, and T. Strothotte, "Illustrative Shadows: Integrating 3D and 2D Information Displays," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, Miami, Florida, 2003, p. 166–173.

[8] H. Sonnet, S. Carpendale, and T. Strothotte, "Integrating Expanding Annotations with a 3D Explosion Probe," in *Proceedings of the working conference on Advanced visual interfaces*, 2004, pp. 63-70.

[9] T. Jung, M. D. Gross, and E. Y.-L. Do, "Annotating and Sketching on 3D Web Models," in *Proceedings of IUI'02*, 2002, pp. 95-102.

[10] K. Osman, F. Malric, and S. Shirmohammadi, "A 3D Annotation Interface Using the DIVINE Visual Display," in *HAVE\'2006 - IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, Ottawa, Canada, 2006.

[11] T. Stefanut and D. Gorgan, "Graphical Annotation – a New Interaction Method with 3D Learning Objects in eLearning Environments," Scientific Bulletin of "Politehnica" University of Timisoara, ROMANIA, Transaction on Automatic Control and Computer Science 2008.

[12] Nintendo. (2009) Wii. [Online]. http://www.nintendo.com/wii

[13] VRContext. (2009, Feb.) Walkinside. [Online]. http://www.walkinside.com

[14] Roberto Ierusalimschy, *Programming in Lua*, 2nd ed. Rio de Janeiro, Brazil: Lua.org, 2006.

[15] L. P. Soares, M. C. Cabral, P. A. Bressan, and R. D. and Zuffo, M. K. Lopes, "Powering Multiprojection Immersive Environments with Cluster of Commodity Computers," in *Proceeding of The 1st Ibero-American Symposium in Computer Graphics*, Guimarães , Portugal, 2002.

[16] Carolina Cruz-Neira, D.J. Sandin, and T. DeFanti, *Surround-screen projection-based virtual reality.*: SIGGRAPH 92, 1992.

[17] I. H. F. Santos et al., "EnViron: An Integrated VR Tool for Engineering Projects," in *Proceedings of the 12th International Conference on CSCW in Design*, Xi'an , China, 2008. Avaliable on URL: http://www.w3.org/MarkUp/VRML/. 2009.