Collaboration between Networked Heterogeneous 3D Viewers through a PAC-C3D Modeling of the Shared Virtual Environment

Thierry Duval [*]
Université de Rennes 1, IRISA UMR CNRS 6074, Rennes, France

Cédric Fleury[†]
INSA de Rennes, IRISA UMR CNRS 6074, Rennes, France

Abstract

We propose to illustrate how the PAC-C3D software model makes it possible to share networked 3D Virtual Environments (VE) between heterogeneous 3D viewers written in Java3D and jReality.

Keywords: Software Architectural Models for CVE

Index Terms: H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: User Interfaces—Theory and methods; I.3.7 [Computer Graphics]: 3-Dimensional Graphics and Realism—Virtual reality; D.2.11 [Software Engineering]: Software Architectures—Patterns

1 The PAC-C3D Model

We propose to design each object of a Collaborative Virtual Environment (CVE) according to the PAC-C3D model [1] illustrated figure 1. It is an explicit evolution of the PAC model dedicated to 3D CVE. On each user's computer, shared virtual objects of a CVE must be decomposed into three main kinds of components described by three interfaces. The *Abstraction* is in charge of the core data and behavior of the object, the *Presentations* are in charge of the virtual representation of the object to the user, and the *Control* is in charge of the consistency maintenance between *Abstraction* and *Presentations*, and between all the distributed *Controls* of the shared object.

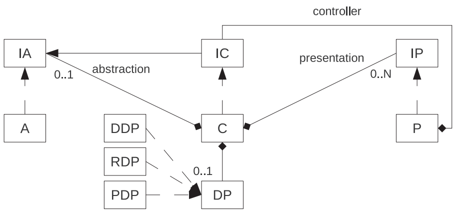

Figure 1: Adaptation of the PAC model for 3D CVE

2 Viewing the same VE with different viewers

PAC makes it possible to design a VE with very small dependency to the 3D graphics API used for the 3D rendering: it proposes to confine all the graphics features of the virtual object in its *Presentation*. PAC-C3D proposes to use explicit interfaces components to strengthen this separation: it makes the *Control* components totally independent of the implementation of the *Presentation* components. The same 2D GUI and external interaction devices allow also to drive in a similar way these 3D viewers, as all the interaction and navigation orders are sent to *Control* and then to *Abstraction* components. We have used this model to design our IIVC [2] and three viewers based on Java3D, jReality and jMonkey.

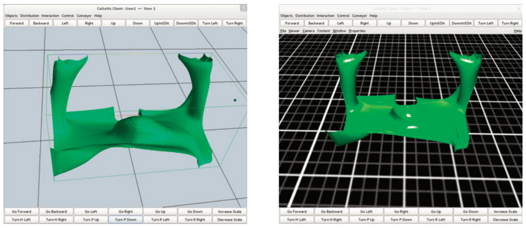
CollaViz Client : User1 -- View 1
Objects Distribution Interaction Control Conveyor Help
Forward Backward Left Right Up Down UpInSDit DownInSDit Turn Left Turn Right
Go Forward Go Backward Go Left Go Right Go Up Go Down Increase Scale
Turn H Left Turn H Right Turn P Up Turn P Down Turn R Left Turn R Right Decrease Scale
Objects Distribution Interaction Control Conveyor Help
Forward Backward Left Right Up Down UpInSDit DownInSDit Turn Left Turn Right
File Viewer Camera Content Window Properties
Help
Go Forward Go Backward Go Left Go Right Go Up Go Down Increase Scale
Turn H Left Turn H Right Turn P Up Turn P Down Turn R Left Turn R Right Decrease Scale

Figure 2: Two different viewers sharing the same virtual environment

3 Sharing the same VE between different viewers

PAC-C3D allows these different 3D viewers to share the same 3D VE at run-time (see figure 2) over a network. PAC-C3D proposes to put all the collaborative features (distribution, synchronization, etc.) in the *Control* components, ensuring that each evolution of a virtual object is distributed to the other *Control* components of this virtual object, according to their distribution policy [3].

4 Enriching 3D viewers inter-operability

It is possible to enrich the interaction possibilities of a viewer *X* with extra functions provided by an other viewer *Y*. It consists in allowing the viewer *X* to control a virtual object provided by the viewer *Y*. The viewer *X* owns a local *Control* of this object, so each order given to this local *Control* by the viewer *X* will be forwarded to the *Control* of the object on the viewer *Y* that will forward it to its *Abstraction*. To be more efficient, the distribution policy of this virtual object can be changed dynamically at run-time. For example, the *Referent Control* can be migrated to the same process than the viewer *Y*. These exchanges between different viewers strengthen their inter-operability.

5 Conclusion

PAC-C3D makes it possible to design a CVE with very small dependency on a 3D graphics API, and it makes it easy to use different 3D graphics API on different remote computers sharing the same collaborative session, providing easy inter-operability between 3D graphics API such as Java3D, jReality and jMonkey.

Acknowledgements

This work was partly funded by the French Research National Agency project named Collaviz (ANR-08-COSI-003-01).

References

[1] T. Duval and C. Fleury. PAC-C3D: A New Software Architectural Model for Designing 3D Collaborative Virtual Environments. In *Proc. of ICAT*, page to Appear, 2011.

[2] C. Fleury, A. Chauffaut, T. Duval, V. Gouranton, and B. Arnaldi. A Generic Model for Embedding Users' Physical Workspaces into Multi-Scale Collaborative Virtual Environments. In *Proc. of ICAT*, pages 1–8, 2010.

[3] C. Fleury, T. Duval, V. Gouranton, and B. Arnaldi. A New Adaptive Data Distribution Model for Consistency Maintenance in Collaborative Virtual Environments. In *Proc. of JVRC*, pages 29–36, 2010.

[*] e-mail: thierry.duval@irisa.fr
[†] e-mail: cedric.fleury@irisa.fr

The 21st International Conference on Artificial Reality and Telexistence
November 28-30, 2011, Osaka, Japan
ISSN: 1345-1278 © 2011 The Virtual Reality Society of Japan