Towards Open Virtual Environments Interoperability

F. Bréant, H. Lê Văn Gông & M. Soto
Laboratoire MASI
Université P. & M. Curie
Paris, France
Tel : +33 (1) 44.27.34.23
Fax : +33 (1) 44.27.62.86
E-mail:{frbr,lvg,soto}@masi.ibp.fr

Key-words : Networked Virtual Environments, Interoperability, Object Modeling.

Abstract

Future trends in virtual environments will involve distributed computing concepts and techniques to allow simultaneous interactions between distant users and virtual objects. We present in this paper some of the requirements needed to build virtual worlds involving cooperating virtual environments. We propose a model that aims to ensure consistent interactions in the distributed virtual environment and independence of object specifications from this environment.

1 Introduction

Research on hardware devices for human immersion in Virtual Reality is still very active. Nevertheless many new virtual environments are developed that take advantage of existing devices.

As hardware gets increasingly computational power, virtual environments can manage an increasing number of more complex objects [LBSC93]. Moreover, communication improvements allow networked implementations [BCML92],[SLGS92]. So, current trends in virtual environments involve distributed computing concepts and techniques to allow simultaneous interactions between many users and virtual objects (Rubber Rocks [CJK+92], dVS [Gri91], VEOS [Bri90]). However, as existing networked environments will spread, the problem of interconnection of heterogeneous virtual environments becomes crucial. One of the major need that have to fulfill virtual environments is cooperative work. Currently, cooperative work is only possible in one virtual environment.

The inescapable evolution will be the extension of cooperative work within *cooperative heterogeneous virtual environments* [FB92].

We propose in this paper a model of architecture for open cooperative virtual environments. Our model is mainly based on the key concepts of *metaphor* and *object exchanging* and *object sharing*. Metaphor allows to specify the laws that rule a virtual environment. We derive from this concept the notion of compatibility of virtual environments. Metaphor concept has also a positive consequence by simplifying the object modeling process and increasing the consistency of the modeling. Our model defines services of migration and sharing of objects to enable cooperation between heterogeneous environment. Migration service is useful to navigate among virtual environments. Sharing service constitutes the most advanced feature of our model because it leads to the creation of common virtual spaces.

Section 2 exhibits basic concepts encountered in existing virtual environments. Section 3 introduces the architecture and underlying concepts of our model. Section 4 proposes the services for achieving interoperability.

2 Current virtual environments

A study of existing systems shows the necessity of a common model to achieve interoperability between VR systems. To illustrate our purpose, we have selected two existing virtual systems NPSNET [ZPMW92]. and DIVE [ACHS94]). We choose those systems for several reasons. The main one is that they both are successful implementations of their own original concepts. Moreover those systems have different aims. NPSNET has been designed to manage an increasing number of participants with a high level of performance while DIVE designers intended to validate a specific methodology and an implementation of notions such as *focus, nimbus or aura.*

To experiment a virtual world, ones need virtual objects to interact with [1]. Thus, we compare DIVE and NPSNET using key concepts such as virtual objects modeling, objects interactions management and distribution mechanisms.

2.1 Object Modeling

Whereas modeling the geometric description of objects is a quite normalized task [CB93], modeling the behavior of objects is a more complex job. By behavior, we mean rules to evaluate the response of an object to an interaction.

Dealing with detailed object behavior may cause performance problems and many VR systems have no choice than to limit the complexity of object behavior.Therefore, objects are modeled using a set of predefined behaviors [Pra93]. In such systems the mechanisms used to compute the response of objects are optimized. Such an approach has been chosen in the design of NPSNET. As mentioned before, the goals aimed by NPSNET were scalability of the number of participants and performance.

[1]The participant himself may be represented as an object

Another solution consists in modelling object behaviors outside the environment. Such an approach allows an extensive modelling of objects. From there, the behavior of the object interacting in a virtual world is rather more complex to manage. Such an approach has been chosen in the design of DIVE.

2.2 Objects interactions Management

One of the main function that a virtual system has to offer is the ability to manage interactions between virtual objects.

To deal with interactions, many current systems [LKL91, ACHS94] use **event notification** to communicate information about those interactions to modules of the system. Briefly, events consist in modification of the virtual world in response to actions executed by an object of this world.

Thanks to its specialization, NPSNET has been able to reduce the number of the possible interactions between objects. It can thus optimize checks for all the actions to perform and the interactions to deal with because responses are compiled.

In DIVE, notification of event is done using a kind of message named *signal.* The signal is broadcast by the system to all the objects to ensure that each active function knows the events related to it. Responses are executed using *callbacks.* Callbacks consist in user functions registered as to be invoked when the associated event occurs. The callbacks prototypes are pre-defined such as "behaviour_signal", "person_change_world" or "collision_signal". In fact, there is one callback prototype for each signal. There may be several user functions handling on the same callback. Thus it is possible to build complex response to an event due to the interpretation of responses, but signals are still statically defined. In DIVE model of interaction relies on original notions such as *focus, nimbus or aura* [ACHS94].

2.3 Distribution mechanisms

As virtual worlds will become larger and more complicated, splitting the virtual system with the management of events in a separate module will allow the designers to adapt and to maintain the management of events. Events are also fundamentals because they constitute the common language for different systems to communicate with objects.

It is quite interesting to notice that while they belong to two different approaches, DIVE and NPSNET share a similar scheme of distribution over the network. Indeed they both have replicated objects on each node. We do believe that, as in distributed systems, the notion of exchange of message (such as events) is natural, though it is well known that it may cause performance problems.

3 Model of architecture for interoperability

In the previous section, we have presented two examples of virtual environments. The question is now: "Is it possible to make them interoperate ?" Another way to formulate the question is for instance: "Can a plane from NPSNET fly in the virtual space of DIVE and, a user of DIVE, walk through a battle-field in NPSNET ?". We have seen that DIVE and NPSNET strongly differ on key points such as object modeling and objects interactions management. These differences would be serious obstacles in an attempt to make them interoperate. It is important to notice that these differences do not arise because interoperability does not belong to their initial priorities. These differences are necessary and arise because initial aims of DIVE and NPSNET are different: methodological aims in one side and performance aims on the other side. The point of this analysis is that heterogeneity is necessary and can not be avoided.

Interoperability of virtual environments can be significantly increased if heterogeneous environments support an external interface defined by a common *model of reference.* This allows to address interoperability problem in general and make virtual environments open. The model we propose describes what should be virtual environment interoperability in terms of concepts, architecture, services and underlying protocols of those services.

In this section, we first define the necessary concepts of a model of reference. Some of them are alredy "in the air". In this case, the validity of the proposed definitions is limited to our model of reference independently of their common use.

3.1 Metaphor

We define a metaphor as a set of simulated laws describing the possible features, behaviors and interactions of any possible entity including human participant. A metaphor is then assigned to a virtual environment to rule this one. Physics is the most common metaphor in use in virtual packages. Depending on the purpose of those packages (e.g., entertainment, business or research), the physics metaphor is proposed with different degrees of fidelity and at various scales (e. g., human and microscopic). Other metaphors derive, restrict or extent the physics one; for example, the human body metaphor for surgeons training or aid. More generaly, a metaphor may rely on any arbitrary set of laws. Discussion about classification of existing or future metaphors is beyond the scope of this paper. Nevertheless, the point is that metaphor is a key concept regarding compatibility and, thus, interoperability. Several benefits from the metaphor concept may be exhibited.

Consistency

When using the metaphor concept, people have first to answer the question: "What is possible to do in the virtual environment we are going to design ?". The needed discussion to answer this question increases the consistency of the final environment. Also, consistency may be significantly improved and even proved if formal techniques are used to describe a metaphor. Comparison of metaphors is possible and, as a metaphor rules a virtual environment, comparison of distinct environments is thus possible and easier. This is an important step toward interoperability: comparing distinct environments by way of their respective metaphors allows to decide if they can interoperate.

Compatibility

As metaphor may be a derivation, an extension or restriction of another one, the notion of compatible metaphors appears natural. From compatible metaphors, we also derive the notion of compatible environments. Compatibility means that enough similarities exist to enable environment to interoperate. For instance, let us define M1, a gravitationnal metaphor, and M2 a spatial one. Let us assume the main difference between those metaphors is that M2 does not take into account properties such as mass and weight. We may consider two virtual environments, E1 and E2, respectivly ruled by M1 and M2, as compatible and thus able to interoperate. For example, if an object moves from the space managed by E1 to the space managed by E2, all the behaviors depending on the mass and weight of the object will not be rendered. Nevertheless it works and this degraded working may be sufficient.

Normalization

As mentioned before, adopting the concept of metaphor makes discussion possible about laws featuring in any peculiar metaphor. Discussion may take place at different levels such as team level or international organization. Both levels of discussion are, of course, profitable for virtual environments. Nevertheless the major benefits are obtained with discussion leaded at international level that favour emergence of standards for metaphors.

Completness

Laws defined in the metaphor allow to predict common behavior of objects. So, object behavior specification only needs to focus on the specific behavior of objects. This implies that a given metaphor called *native metaphor* is attached to any object. The notion of native metaphor also improves interoperability. Indeed, interactions abilities between two objects may be evaluated by comparing their respective native metaphors.

3.2 Virtual entity

A virtual entity plays a role in a virtual space. This role ranges from trivial to highly complex. Roughly, virtual entities can be divided in three classes. *Passive entities* standing for simple objects such as wall, chair or table have no peculiar behavior. *Semi-passive entities* exhibit more complex behaviors and involve various interactions with others objects. For instance, entities such as virtual car, plane, television need many informations to have a realistic behavior. Lastly, the main feature of *Autonomous entities* is their capacity of decision making to reach a goal and adaptation to the environment. Such a class includes human participant or autonomous robot for example. We distinguish several components in a virtual entity.

- appearance description
- behavior specification

- multimedia feature
- native metaphor
- interface communication

The two first components are usual in description of virtual objects. The multimedia component plays a specific role in the behavior and appearance of an entity. Embedded multimedia is already sensed as a key feature of future virtual environment [ZLP93]. The last two ones are required for interoperability purpose. We state in our model that any entity is specified in conformance to a given metaphor called *native metaphor*. A native metaphor allows us to define a context to which the object is dedicated. For instance, let us specify an entity "chair" with the native "physics" metaphor that contains the gravity notion. Mass and volume are required in the definition of the chair. Then, the weight parameter is implicitely associated to the entity chair by the environment. In the same way, behaviors related to the gravity notion will implicitely influence the motion of the chair in any environment running under the same metaphor. An *interface communication* is provided with each entity. Such interface allow entities to exchange messages between them and with the environment. To achieve interoperability, these interfaces has to be built using a communication protocol. Existing virtual environments already take advantages of such kind of protocol [DIS93].

3.3 Virtual space and time

Mainly two reasons make the concepts of space and time compulsory in a model of reference for virtual environments interoperability:

- Space and time are necessary concepts for experiencing a metaphor. Relationships between entities can not be managed without those concepts. Metrics are used to calculate spacial and temporal properties (e.g.; location, distance, volume and speed) of entities.
- Human perception of space and time has to be mapped onto virtual space and time. Different types of mapping have been identified by [Rob92]: alignment, displacement and scalling. Virtual space and time are ruled by the metaphor.

3.4 Distributed virtual system

The virtual system constitutes the software support of the virtual world. It contains some classical items that composes a virtual environment:

- metaphor engine
- Entities engine
- Entities management
- Collision detection
- Scenes rendering

We call *metaphor engine* the application of the laws composing the metaphor on the virtual world. *Entity engine* applies laws dedicated to one entity behavior, and *entities management* constitutes the part of the virtual environment which provides services for including entities based on the same metaphor or based on other metaphors. Distributed virtual system is obtained by distributing these parts on different networked sites in order to increase perfomances.

3.5 Cooperative virtual environment

The cooperative virtual environment gathers a set of heterogeneous virtual environments that cooperate to build and to manage a virtual world. The degree of interoperability is characterized by metaphors used in each virtual environments.

4 Services

Components of a virtual environment, previously described offer a set of services to manage entity instanciation in a virtual environment, migration and sharing of entities between environments plus usual features like interaction between entities.

Instantiation
We distinguish 2 cases in entity instanciation. First, the entity is instanciated in a space ruled by the native metaphor of the instantiated entity. Second, environment metaphor and entity native metaphor are different but compatible.

Migration
This service allows motion of entities between different environments. Migration means that one entity leaves its current environment and enter in another one. So, the management of the moving entity is also transfered to its new environment. For example, migration services allow a user to navigate between severals virtual worlds ruled by compatible metaphors.

Sharing
Sharing service enable the building of cooperating spaces. In this case, entities are managed by all the participating virtual environments. Sharing service include negotiation facilities to map the management of shared entities.

Interactions
Interaction services must be invoked when events occur in a virtual environment. Interactions service deals with different kinds of interactions. These events may have different origins: the cooperative environment and the entities.

5 Conclusion

In this paper, we have presented some of the key features that aim to improve interoperability between virtual environments. We do not impose any implementation constraint even if our model favors object oriented techniques.

The study of current virtual systems has allowed us to depict a model for interoperability between virtual environments. Our model takes into account heterogeneity of virtual environments. The concept of metaphor simplifies the modelling of objects and provides paradigm to evaluate compatibility between virtual environments. Sharing and migration of objects constitute basic services to enable open cooperation.

Our future work will focus on the accurate elaboration of the mentionned services. The next step consists in designing a virtual environment using our model and to implement layers on top of existing environments in order to validate our interoperability concepts.

References

[ACHS94] M. Andersson, C. Carlsson, O. Hagsand, and O. Stahl. *DIVE – The Distributed Interactive Virtual Environment Technical Reference Manual.* Swedish Institute of Computer Science, Box 1263, 164 28 Kista, Sweden, 1994.

[BCML92] C. Blau, C.E.Hughes, J.M. Moshell, and C. Lisle. Networked virtual environments. *Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics*, pages 157–160, 1992.

[Bri90] W. Bricken. Software architecture for virtual reality. Technical report, Human Interface Technology Laboratory, Seattle, WA, April 1990.

[CB93] P. Coiffet and G. Burdea. *La Réalité Virtuelle.* Hermes, 1993.

[CJK+92] C. Codella, R. Jalili, L. Koved, J.B. Lewis, D.T. Ling, J.S. Lipscomb, D.A. Rabenhorst, and C.P. Wang. Interactive simulation in a multi-person virtual world. In *CHI'92*, pages 329–334, May 92.

[DIS93] IEEE Standard for Information Technology – Protocols for Distributed Interactive Simulation Applications, Entity Information and Interaction . Technical Report ISBN 1-55937-305-9, The Institute of Electrical and Electronics Engineers, 1993.

[FB92] H. Fuchs and G. Bishop. Research directions in virtual environments. *Computer Graphics*, 26(3), August 1992.

[Gri91] G. Grimsdale. Distributed virtual environment system dvs. Technical report, Division Ltd, 1991.

[LBSC93] L.E.Fahlen, C.G. Brown, O. Stahl, and C. Carlsson. A space based model for user interaction in shared synthetic environments. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, and T. White, editors, *Conference on Human Factors in Computing Systems*, pages 43–48, Amsterdam,, April 1993. ACM Press. 24-29.

[LKL91] J.B. Lewis, L. Koved, and D.T. Ling. Dialogue structures for virtual worlds. In *CHI'91 Proceedings*, 1991.

[Pra93] D.R. Pratt. *A Software Architecture for the Construction and Management of Real-Time Virtual Worlds.* PhD thesis, Naval Postgraduate School, Monterey, California, June 1993.

[Rob92] W. Robinett. Synthetic experience: A proposed texonomy. volume 1, pages 229–247. MIT Press, Spring 1992.

[SLGS92] C. Shaw, J. Liang, M. Green, and Y. Sun. The decoupled simulation model for virtual reality systems. *CHI'92*, pages 321–328, May 1992.

[ZLP93] M.J. Zyda, C. Lombardo, and D.R. Pratt. Hypermedia and networking in the development of large-scale environments. In *International Conference on Artificial Reality and Tele-existence*, pages 33–39, Tokyo, Japan, July 1993.

[ZPMW92] M.J. Zyda, D.R. Pratt, J.G. Monahan, and K.P. Wilson. Npsnet: Constructing a 3d virtual world. *Symposium on Interactive 3D Graphics*, pages 147–156, 1992.