



HAL
open science

Achieving Networked Virtual Environments Interoperability

Michel Soto, Hubert Lê van Gông

► **To cite this version:**

Michel Soto, Hubert Lê van Gông. Achieving Networked Virtual Environments Interoperability. 4th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS'97), Sep 1997, Darmstadt, Germany. pp.41-50, 10.1007/BFb0000338 . hal-01630555

HAL Id: hal-01630555

<https://hal.science/hal-01630555>

Submitted on 18 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Achieving Networked Virtual Environments Interoperability

Michel Soto and Hubert Lê Văn Gông

Laboratoire LIP6 – Université Pierre et Marie Curie
75252 Paris Cedex 05, FRANCE

Abstract. The interoperability between heterogeneous distributed virtual environments is one of the main problem Virtual Reality will have to face in the very close future. This article presents the main concepts we propose in NOVÆ (which stands for Networked Open VirtuAl Environment). These concepts are presented as a set of paradigms based on the semantic of a virtual world and its virtual entities.

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

1 Introduction

As the range of existent applications for synthetic environment (SE) is rapidly growing, the will of being able to interact between different users on different SEs will dramatically increase.

It is now generally accepted that multi-user networked SEs constitute a major area of interest. SEs allow users located in different geographical areas to jointly perform complex tasks. Thus, networked SEs can be applied to almost all the human domains of activities.

Today, there are several existing networked SEs and numerous other ones will appear in the close future. Current existing networked SEs can be considered as closed in the way that users of two distinct SEs are not able to work together. This situation is no longer desirable and constitutes a paradox while the World is becoming a global village where information highways will link the main places.

New networked SEs must allow users to cooperate freely or, in a more general way, to interoperate with other networked SEs. Meanwhile already existing SEs have to evolve taking into account this new requirement. Some efforts have already been done in this direction. The DIS protocol is a good example of what have been obtained in this topic of research [DIS 1993]. Nevertheless, to be successful on the way of interoperability, we believe in the need of new paradigms and in the consolidation of existing ones. Taking into account the interoperability requirement immediately raises the problem of heterogeneity. We sometimes can restrict heterogeneity but never avoid it. Therefore, successful paradigms aiming at interoperability must take it into account. Heterogeneity appears at

any level between distinct SEs: hardware, software, operating system, communication stack protocols, input/output devices. Solutions already exist (e.g., ISO and TCP/IP protocols, ODP, CORBA) to deal with these levels of heterogeneity and thus allowing interoperability. The point is that heterogeneity also appears at the application level which is the most important for networked SE. At this level, interoperability paradigms must allow to answer questions such as:

- Having users working together from their respective synthetic environments, what does their interaction capabilities become ?
- What does the behavior of a virtual entity migrating from its native synthetic environment to a new one become ?

In this paper we propose architectural paradigms to achieve interoperability between networked SEs. Section 2 depicts the proposed paradigms (virtual entity, metaphor, virtual system) and highlight their major benefits. Finally we discuss the compatibility evaluation of SE systems.

2 Architectural Paradigms

The paradigms we propose aim at providing an open architecture for virtual environments. The starting point of our reflexion is the semantic associated to a virtual entity. This semantic represents the purpose the entity has been designed for. For instance, a virtual meeting desktop may be designed to automatically duplicate the document to the meeting participants. If we immerse this entity in a virtual world created to provides meeting points, the entity semantic is adequate to the virtual world's one.

If we now consider two different virtual worlds, they possibly have two different semantic. Is it possible to use the virtual meeting desktop in both environment in order to work together ? What can I expect to do with an object imported from another environment ?

We do believe that a solution to provide virtual environments openness relies on addressing the semantic and the computing aspects of the problem. The former one having consequences on the solutions to the latter one.

Our solution relies on the distinction we make between an action and the consequences of this action. In order to do that we use the Influence/Reaction model developed in multi-agents systems by Ferber [Ferber 1995]. This distinction allows us to address the fact that it is impossible to define all the possible consequences of an action because we cannot know by advance in which virtual world the entity will be immersed.

2.1 The virtual entities

The *virtual entity* is undoubtedly a central notion in a SE. Unfortunately, its definition can slightly vary from a system to another one. For example, AVIARY [Snowdon et West 1994] implements autonomous and communicating objects

whereas NPSNET [Pratt 1993] provides semi-automated forces (SAF) and autonomous Agents (AA). We represent virtual entities as a set of modules, each one corresponding to a particular aspect of objects in virtual worlds (figure 1).

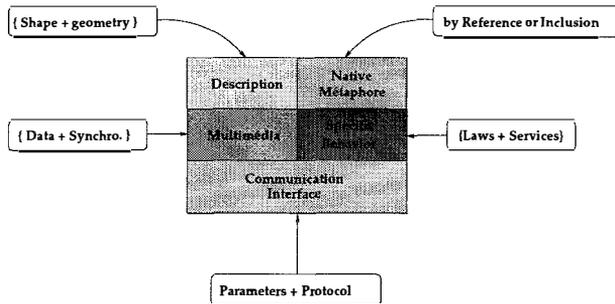


Fig. 1. A virtual Entity

- **The Appearance** corresponds to the description of the geometry of the entity. It also refers to any information related to the appearance of the entity (e.g., graphical, haptic, audio). The description of these data is performed using well-known languages (e.g. Autodesk's DXF, JPEG, VRML...) and thus can be processed by engines.

- **The Multimedia** component plays a specific role in the interactivity capacity of the entity. Embedded multimedia is already sensed as a key feature of future virtual environment [Zyda *et al.* 1993] because of the number of potential virtual applications which would benefit of multimedia capacities (e.g., distant learning, telemedicine). Several standards such as MHEG ([ISO 1994, ISO 1996]), HyTime [HyT 1992] and HyperODA [ISO 1992] enable designers to describe multimedia documents. All those formats also need specific engines to exploit their data and synchronize the flows of the different media. Some SEs already implement similar possibilities, such as NPSNET which provides *anchor* as a deposit of multimedia information [Zyda *et al.* 1993].

- **The Communication Interface** is a repository for the necessary parameters to enable the communication between the virtual entity and the rest of the virtual world. The interface also provides a communication protocol such as DIS [DIS 1993] to ensure communication possibility between virtual entities and with the environment.

- **The Specific Behavior** gives the opportunity to a virtual entity designer to modify the default behavior of the entity with another pattern. This behavior represents the peculiar actions of the entity, the rules it is the only one to follow. We represent the behavior by a set of laws using a formalism based on the Influence/Reaction model. Entities designed with language such as VRML 2.0 may have scripts containing behaviors defined with our formalism.

Some SEs already provide specific behavior to virtual entities. For instance, AVIARY [West *et al.* 1993] provides objects with specific behavior called *volitional behavior*. The benefit is to ease the definition of autonomous actors to populate the virtual worlds and to increase the quality of the simulation.

- **The Native Metaphor** is a reference to the metaphor the entity has been designed for (see definition of the term metaphor below 2.3). That means the entity was thought to fully behave according to the laws contained in the mentioned metaphor. Indeed, an entity is always designed to be used in a certain way even if our architectural paradigms intend to make possible interaction with any entity.

2.2 The laws

Two types of laws are needed to distinguish effectively between an action and the consequences of the action.

- The *behavioral law* type is used to specify the actions performed by any virtual entity.
- The *reaction law* type allows to specify how the virtual world changes (i.e., reacts) according to the current state of the virtual world and according to **all** the actions of the virtual entities. The reaction laws are included in the metaphor rulling the virtual world. So, specifying the behavior of a virtual entity, we do not have to describe all the possible consequences on the state of the virtual world.

For example, if a single virtual robot pushes a virtual table, this action will modify the current state of the virtual world since the location of the moved table will change. The final location of the table will be different if there are, now, several robots pushing simultaneously the table in different directions. The distinguishing between action and reaction allows us to focus on what is the push action in the behavior of a robot without the burden of envisaging an infinity of scenarios when the action is performed. The new state of the virtual world (i.e., the final location of the table in our example) will be calculated by the virtual system using the reaction laws of the metaphor.

The dynamic of a virtual world is based on a two phases schema: the production of influences followed by the reaction of the virtual world. The figure 2 presents this schema.

The application of both a behavioral law and a reaction law depends on conditions to be fulfilled. A pre-state defines a condition on the state of the

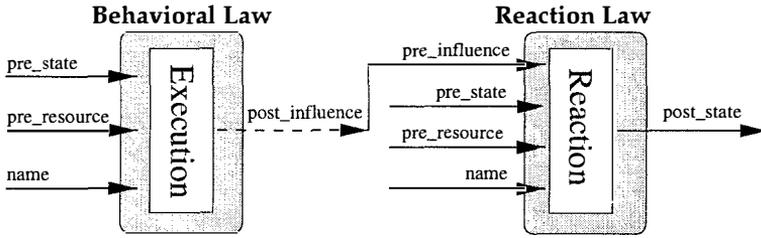


Fig. 2. The Dynamic of the Virtual World

virtual world. For example a pre-state for performing a push action on the virtual table is that the robot and the table must be in contact. The pre-resource is a special pre-state as detailed in section 2.3. When the pre-conditions of a behavioral law are satisfied, one or more post-influences are produced. A post-influence indicates an attempt to modify the state of the virtual world. For instance, a post-influence in the behavior of a robot will indicate its attempt to push the table in a given direction. The post-influences become pre-influences for reaction laws.

Thus a pre-influence is a condition on an attempt of modification of the virtual world. Arbitrary complex scenarios of interaction between virtual entities can be addressed in this manner. When the pre-conditions of a reaction law are satisfied one or more post-states are produced. A post-state thus defines an actual modification of the virtual world. The phases of influences production and reaction are repeated infinitely.

2.3 The metaphor

In our model, a metaphor is the expression of the virtual world's semantic. It's an intention, a goal related to a virtual world. This intention induces the specification of the world's reactions.

Hence, a metaphor is a set of laws defining the reactions of the environment to the influences generated by the entities. Those reactions may vary from an environment to another, and it's the metaphor's job to implement those characteristics. For instance, the lift up of a rocket might not produce the same result if we're in Space or in Paris.

The previous example shows that the metaphor also have to express impossibilities in order to achieve modelling of consistent virtual worlds : the space environment should forbid the propagation of soundwaves. However, it would be wrong to prevent a rocket from making noise : the engine is still functioning in the same way, but we cannot hear it. In order to deal with this problem, we propose to determine the execution of a law by the presence of resources. These resources represent a part of the world's semantic.

We define two kind of resources in a metaphor : positive and negative. The former one means that the resource exists in the virtual world. This resource represents a property of the virtual world. For example, a virtual application designed for the simulation of an ecosystem might need an *atmosphere* resource to distinguish bacterium who can't live in air from thos who can.

A negative resource means that this resource doesn't exist in the virtual world and that it is forbidden in order to maintain the world consistent. If a virtual world contains such a resource, this is to indicate that the creator of the world designed it with this incompatibility in mind.

The absence of a resource in a metaphore means that the virtual world has been modelled to be used without this resource, but that the presence of it shouldn't lead to inconsistency (at least, there's no explicit incompatibility).

To summarize, we propose the following definition :

A metaphor is the representation of a virtual world's reactions to the influences produced by its virtual entities. It is composed of a set of resources and a set of laws.

The benefits As we explain it in [Gông *et al.* 1994], the notion of metaphor is one of the key point of our architecture because it provides important benefits :

- **Consistency:** As a metaphor gathers all the laws composing the environment's reaction, the designer of a virtual world has a better overview of the global behavior of its virtual applications. Using the formalism we propose to design the laws, it would even be possible to use some formal methods to determine the consistency of the virtual world.

- **Simplicity:** Without the model we propose, if we want an object to have a consistent behavior in two different virtual worlds, we have to define the result of an action in both virtual world. Therefore, it is impossible to predict all the different consequences of an action among the virtual world an entity is immersed in. Using our model, the designer defining a particular behavior has no longer to define all the possible consequences of these actions. Obviously, the use of the metaphor represents a benefit in time (for the design of new virtual entity) and in performance.

- **Interoperability:** The main benefit for interoperability resides in the fact that the virtual world's reactions to entities influences are defined in the metaphor. This allows us to re-use entities while avoiding inconsistency problems thanks to the resource notion.

Also, the evaluation of the needs in resources (network, system and hardware ones) is more accurate using the semantic content of the metaphor. For instance, the needed network bandwidth to maintain the coherence of a virtual world populated by rapidly moving entities (e.g., planes, data stream) will be higher than if the world was populated by static virtual entities (e.g., humans) [Council 1995, Macedonia et Zyda 1995].

Using our metaphor model also eased us to develop the needed mechanisms (1) to determine levels of compatibility between different virtual environment to interoperate, (2) to calculate the resulting behavior of each virtual entity involved in an interoperability process.

2.4 The underlying virtual system

The virtual system is the base of our architecture. It contains the different managers in order to deal with the internal mechanisms. We do not impose the content of those manager, but we give their external definition.

We defined the following managers as necessary in the virtual system :

- **the entities manager** is in charge of the system's structures related to the virtual entities. It is responsible for the instantiation and the destruction of entities. For instance, when an entity is imported using the migration service, the entities manager has to be informed of this new entity in order to perform its initialisation. For distributed virtual environments, the architecture of the entities database may vary from a system to another.
- **the law manager** is responsible for the behavioral laws applications. It gets the influences generated by the entities laws analyses them and finally launch the reactions. As the architecture of our virtual system may be distributed, each site has at least one law manager. On a single site, one may need to have more than one law manager because there's no official language for the modeling of a virtual entity's management. Another reason to duplicate the law manager is the fact that it represents a potential bottleneck.
- **the interoperability manager** has to deal with interoperability operations. It is able to determine the compatibility level between an entity and a metaphor or between two metaphors. The other part of the Interoperability manager (IM for short) is to decide which mechanism is to be used in order to increase the compatibility level when this level is too low. These mechanisms may be the use of virtual RPC (called VRPC) or the use of a metaphors server.
- **the SIP (Service Information Provider)** has been introduced in our model in order to inform new entities. The informations it provides are related to the services the entity may request. Indeed, for a new entity, it is important to be able to get some information concerning the different services available. This means of discovering the world will allow a better adaptation of an incoming or a new entity. The SIP is very similar to the ODP trader plus the dynamicity of the CORBA ORB. We'll note that HLA (the High Level Architecture) also admits the importance of such mechanism because it uses some elements of a CORBA implementation.

The following figure (fig 3) represents the general architecture of a virtual environment.

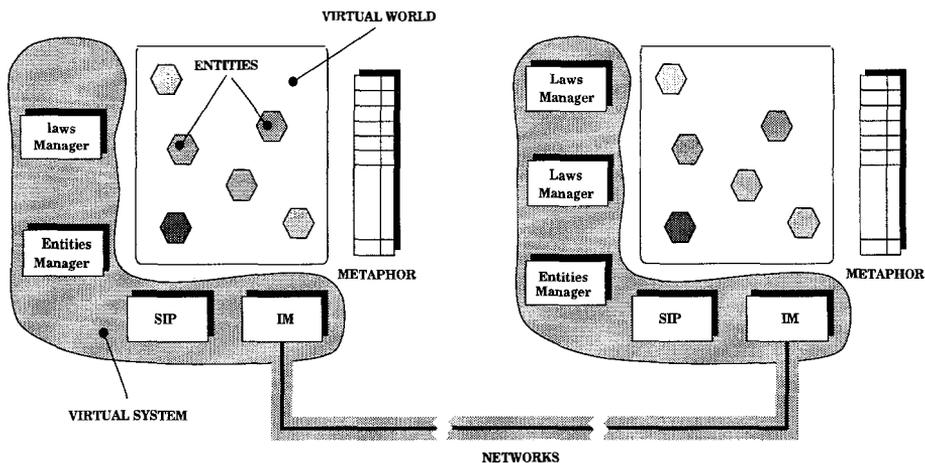


Fig. 3. The General Architecture

3 The Compatibility Evaluation

Compatibility evaluation is necessary to determine the potential of interoperability between different SEs. There are three kinds of compatibility depending on the protagonists involved:

- **Metaphor-Virtual entity:** This level of compatibility is necessary for the migration service.
- **Metaphor-Metaphor:** for sharing services.
- **Virtual entity-Virtual entity:** if the goal is to enable the interaction of two virtual entities (for instance, a virtual application that wants to deal with a resource on another virtual environment using direct sharing service).

Both compatibilities require such an evaluation to determine the quality of future interactions between the virtual entity/metaphor and the metaphor of the receiving SE. This compatibility level ranges from 0 (interoperability is impossible) to 1 (complete interoperability).

The calculus of the compatibility level is made possible by the use of the formalism we introduced to describe the laws. Using this formalism and the resources mentioned in the metaphor, we are able to determine a compatibility level and finally to predict the potential behavior of an entity in a virtual world.

Once a compatibility level has been calculated, several possibilities exist depending on the value of this compatibility level:

1. *level = 0* : the interoperability is impossible.
2. *level = 1* : the interoperability is full.

3. $0 < level < 1$: several possibilities exist, we describe them below.

In the third case, the protagonists may decide :

1. to give up the interaction because the compatibility level is too low,
2. to interact in the debased mode obtained among the compatibility level,
3. to increase the compatibility using the fourth service we proposed (Metaphor servers with virtual RPC).

A metaphor server is a repository of laws that can be exported or directly applied in a virtual environment via the use of virtual RPC (VRPC). VRPCs are quite similar to the well-known RPC but enhanced in order to meet the needs of distributed virtual environments (see [Zelesko et Cheriton 1996]).

There are several reasons why a compatibility level can be less than 1. One of the more intuitive reason is that some properties of the virtual entity need some laws that do not exist in the recipient SE. We solve this problem by transferring those laws over the network. BrickNet implements a similar solution by transferring the methods to generate the entity's behavior [Singh *et al.* 1995])

However, for complex entities such as big applications over wide network bandwidth, performance will gain using virtual RPCs. They consist in asking the remote environment (or a metaphor server) to apply those laws on the local virtual world.

4 Conclusion and Further Works

We proposed architectural paradigms to take into account the unavoidable heterogeneity of networked SEs. They rely on the central concepts of metaphor defined as a set of laws and resources, and virtual entity. These concepts allow us to calculate the potential degree of interoperability between involved SE systems. The metaphor is also used during the interoperability session to calculate the resulting behavior of virtual entities.

To enable interoperability, we exhibit in [Bréant *et al.* 1994], [Gông *et al.* 1994] the need of 3 types of services: migration, cooperative sharing and direct sharing which cover all the possible VR applications. Our concept and services lead to achieve interoperability between heterogeneous networked SEs.

Those paradigms allow us to propose :

- A formalism to represent virtual environment and entities semantic. We do believe this is a rather new possibility.
- A methodology to develop open virtual environment.
- A way to predict the behavior of an entity in another environment.

We currently implement the architecture we propose. The kernel of the virtual system, the law manager has been developped in C. It implements the Influence/Reaction model and use the formalism we built in order to represent behavioral and reaction laws.

A VRML 2.0 parser is also being developed in order to load VRML2-compatible worlds. This parser is part of the entities manager. The next step is to develop the compatibility evaluation and the interoperability manager.

In the close future, we will use the semantic to extract criteria for an efficient load balancing of the distributed virtual system.

References

- [Bréant *et al.* 1994] F. Bréant, H. Lê Van Gông, et M. Soto. Towards open virtual environments interoperability. Tokyo, Japan, July 1994. International Conference on Artificial Reality and Tele-Existence.
- [Council 1995] National Research Council. *Virtual Reality, Scientific and Technological Challenges*. National Academy Press, 1995.
- [DIS 1993] *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.
- [Ferber 1995] J. Ferber. *Les Systemes Multi-Agents, vers une intelligence collective*. 1995.
- [Gông *et al.* 1994] H. Lê Van Gông, F. Bréant, et M. Soto. Architecture for virtual environments cooperation. San Antonio, Texas, USA, October 1994. IEEE International Conference on Systems, Mans and Cybernetics.
- [HyT 1992] Hytime - hypermedia time based structuring language. (10744), 1992.
- [ISO 1992] ISO. Hyperoda - hypermedia office document architecture. (8613/PDAM 7-10), October 1992.
- [ISO 1994] ISO. Mheg: Information technology - coding of multimedia and hypermedia information, part 1: Mheg object representation, base notation (asn.1). 1994.
- [ISO 1996] ISO. *MHEG: Information Technology - Coding of Multimedia and Hypermedia Information, Part 2: alternate notation (SGML)*. Technical report, 1996.
- [Macedonia et Zyda 1995] M. Macedonia et M. Zyda. A taxonomy for networked virtual environments. In *Workshop on Networked Realities*, Boston, MA, October 1995.
- [Pratt 1993] 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.
- [Singh *et al.* 1995] G. Singh, L. Serra, W. Png, A. Wong, et H. Ng. Bricknet: Sharing objects behaviors on the net. In IEEE Computer Society Press, editor, *Virtual Reality Annual International Symposium '95*, March 1995.
- [Snowdon et West 1994] D. Snowdon et A. West. The aviary distributed virtual environment. In Robin Hollands, editor, *Proceedings of the 2nd UK VR-SIG Conference*, pages 39–54, 1994.
- [West *et al.* 1993] A.J. West, T.L.J. Howard, R.J. Hubbard, A.D. Murta, D.N. Snowdon, et D.A. Butler. Aviary - a generic virtual reality interface for real applications. In R.A. Earnshaw, M.A. Gigante, et H. Jones, editors, *Virtual Reality Systems*, chapter 15, pages 213–226. Academic Press, March 1993.
- [Zelesko et Cheriton 1996] M. Zelesko et D. Cheriton. Specializing object-oriented rpc for functionality and performance. In *International Conference on Distributed Computing Systems, ICDCS*, 1996.
- [Zyda *et al.* 1993] M.J. Zyda, C. Lombardo, et 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.