# Conceptual models for Describing Virtual Worlds

F. Bernier†, E. Boivin‡, D. Laurendeau‡, M. Mokhtari†, F. Lemieux†

† Distributed Synthetic Environment Group
Defence R&D Canada – Valcartier
Val-Bélair, Qc, Canada, G3J 1X5
{francois.bernier, marielle.mokhtari, francois.lemieux}@drdc-rddc.gc.ca

‡ Computer Vision and Systems Laboratory
Dept of Electrical and Computer Engineering
Laval Univ. Canada, G1K 7P4, Québec, Qc
{eboivin, laurend}@gel.ulaval.ca

## ABSTRACT

A conceptual model of a virtual world is a high-level representation of how the objects behave and how they are related to each other. The conceptual models identify the most essential elements of the reality to be simulated. This is the first and a very important step in the process of designing a virtual world. Afterwards, specific and complex models can be implemented and inserted into these conceptual models. This paper provides an overview of existing conceptual models used to design virtual worlds. A number of existing frameworks and architecture for describing virtual worlds are classified into six kinds of conceptual models: unstructured, graphic-oriented, network-oriented, object-oriented, environment-oriented and relational graph-oriented representations. The advantages and issues regarding virtual world design, management, reusability and interoperability are discussed.

## Keywords

Virtual World, Conceptual Model, Reusability, Modularity, Extensibility

## 1. INTRODUCTION

Creating a virtual world (VW) implies choosing among many alternative design approaches. One of these decisions is the structure or the kind of conceptual model on which the VW will be built. A conceptual model is a high-level representation of how objects behave and how they are related to each other. The kind of conceptual model can have an impact on the reusability, extensibility and modularity of a VW. These characteristics are important in VW for the same reason as in software domain: increase in flexibility, decrease in cost and time, etc. This paper presents categories of conceptual models VW and shows the effect of these categories on above-mentioned characteristics.

This paper is organized following the classification of the conceptual models underlying many commercials and academics VWs. A few examples of existing VWs are given for each kind of conceptual model. Finally, the issues and advantages of each kind of conceptual models in regard of the creation, the management, the reusability, extensibility and modularity of VW are presented.

## 2. Conceptual Models

Most VWs use ad hoc conceptual models while others rely on more structured and rigorous approaches. This section will present the most common kind of conceptual models used to develop VWs.



***WSCG POSTERS Proceedings***
*WSCG'2004, February 2-6, 2004, Plzen, Czech Republic.*


### 2.1 Unstructured conceptual models

Many existing VWs do not use any specific type of conceptual model and rather adopt an unstructured ad hoc model. This approach is successful when the VW is dedicated to a simple application and when it is developed by a small design team, but the design and implementation of complex applications becomes rapidly unmanageable.

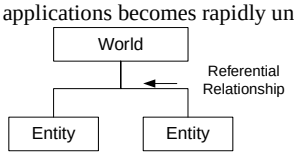


**Figure 1 Unstructured representation of a virtual world.**

As illustrated in Figure 1, when using an unstructured conceptual model, the entities populating the VW share the same virtual "space" and the task of defining their behaviour and the VW structure is left to the designers of the application. This kind of conceptual model allow the designer to optimized the VW for the current application. However, VWs based on this kind of conceptual model will be hardly extendable. For instance, the rules for routing the events to the entities or the calling sequences are tailored for the application with little opportunity left for reuse.

### 2.2 Graphic-oriented conceptual models

The scene-graph approach is a widely used modelling paradigm in virtual reality (see Figure 2). Even though scene-graphs have been mainly developed for building graphics representations of the reality, they are also used to create and animate VWs. The animation task is achieved by adding behaviours to the entities populating the VW which, in this case, consists of the scene-graph itself. Instances of the scene-graph paradigm are found in [VRML97], [OpenInventor], [Viskit], [WTK] and many other graphics rendering toolkits. In a scene-graph, each entity populating the VW corresponds to a node in the graph. A relationship between a node (entity) and other

nodes can only be achieved by referencing the nodes (e.g. by establishing a "referential relationship" in the scene-graph).

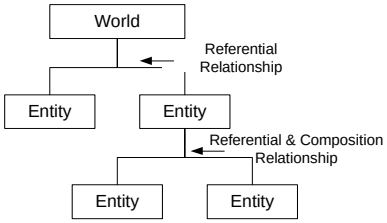


**Figure 2 Scene-graph representation of a virtual world.**

A simple instance of a scene-graph is the four wheels belonging to an automobile. Each wheel corresponds to a node in the graph. The "wheel" nodes are linked to the "car body" node by referential relationships. The relative position and orientation of the entities are stored in special nodes.

The main problem with the scene-graph paradigm is that, although it is well suited for describing the spatial relationships between entities, or more specifically, the graphics appearance of entities populating the VW, it fails in modelling non-graphical behaviours. Consequently, the VW can only manage the visual coherence of its constituents. In the scene-graph conceptual model, the behaviours of the entities are implemented using overriding or callback mechanisms. The interactions between the entities are managed by an event routing mechanism. The entities publish events related to their behaviour to a event-management service in the VW and this event-management service routes these events to the entities that have subscribed to this service. This approach is highly flexible. However, even if the service is able to transmit a wide range of events that are relevant to an entity, it cannot manage and guarantee the coherence of this entity and the interoperability between the entities participating in an interaction. This task is left to the design team and it is a serious limiting factor for VWs of medium to high complexity in which a high level of reusability, extensibility and modularity is required. For instance, a virtual pen and its cap are represented by two different scene-graphs models and integrated, at run time, into to the same VW. The following behaviours shall be expected from their interactions:

1. The cap shall hide the tip of the pen;
2. The cap and the pen shall hold together due to friction.

A VW based on scene-graphs (which was the paradigm used for designing the cap and the pen) can only support the first behaviour since it is relevant to the appearance of the pen. Even if the second behaviour can be implemented in one of the two scene graphs with an ad hoc physics-based external model for friction, it cannot recognize (physically) the entities in the second graph. This hybrid design approach (scene-graph and unstructured) succeeds for monolithic VWs with a small number of physics-based behaviours but fails for dynamic VWs with a large number of complex behaviours.

The graphical approach is an entity-centric modelling approach, since entities are more important than the interactions between them. Like for many kinds of conceptual models, problems may appear when many different types of symmetrical interactions exist between entities. For instance, object Earth attracts all the "owned" entities according to the universal law of attraction. However, this law is bi-directional, thus every "owned" entity should attract object Earth in turn. Collision deals with the same type of problem since collision is an interactive phenomenon depending on all participating entities. The reuse of these symmetrical behaviours only is not possible since it implies the reuse of many entities.

## 2.3 Network-oriented conceptual models

Several VWs are distributed over a computer network. In order to optimize network communication between virtual entities, Networked Virtual Environments (NVE) [Singhal99] or Distributed Virtual Environment (DVE) [Singhal99] like Spline [Open Community] and NetEffect [Das97] improve the previous scene-graph conceptual model by refining it with the concept of "locale".

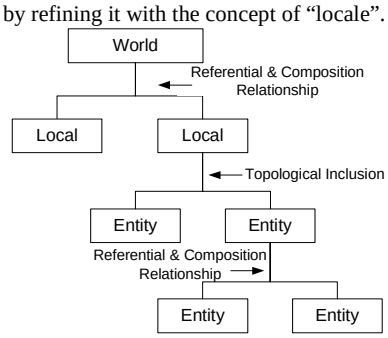


**Figure 3 Network-oriented representation of a virtual world.**

This modelling paradigm allows the entities to be managed in well-defined 3D volumes. For instance, an avatar in the VW can see only the entities sharing the same room. Open Community [Open Community] and Massive3 [Greenhalgh00] improve this concept of "locale" by allowing hierarchical entities in the different locales composing the VW.

The approach facilitates the optimization of network communication by limiting the interactions of an entity to other entities in the same locale. However, the network-oriented approach cannot manage interoperability and maintain coherence of the VW. Like for the two previous approaches, the user must use an ad hoc representation of the reality. The resulting reusability, extensibility and other interesting features are then limited because the VW itself cannot manage new added components or reuse existing one into another context. Like for the graphic-oriented conceptual model, this entity centric apporach limits the reuse of behaviours alone.

## 2.4 Object-oriented conceptual model

The object-oriented (OO) software design approach is often used as a kind of conceptual model. In the design of VWs, the Object-Oriented Physical Modelling approach ([OOPM] [Fishwick96]) is an example of this paradigm. It proposes a way of integrating geometry and dynamics of entities.

Figure 4 shows the structure of a simple model built using the object-oriented approach.

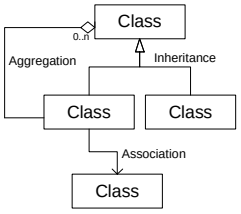


**Figure 4 Object-oriented (OO) representation for a virtual world.**

The OO approach can improve the reusability, extensibility and modularity of a VW. A VW using this kind of conceptual model can define a set of generic classes. The VW knows how these classes interact together. The VW can be extended with any entity inheriting from one or many pre-defined base classes. For instance, a cap of pen inherits form a Visible and Collisionable class. When the cap will hit a pen (another visible and collisionable object), the VW will be able to manage friction and visual occlusion between these two entities. Another advantage of OO programming is that a class does not necessarily represents an entity. For example, a class could implement a collision detection behaviour. This behaviour could then be reused without any entity.

However, this paradigm contains many pitfalls that prevent to achieve efficient modelling of highly dynamic virtual worlds. These issues a caused by inadequate aggregation, intrinsic limitation of inheritance and continuous refactoring of inheritance tree (fragile base class problem [Mikhajlov98]). For instance, aggregation in most OO languages cannot be specified. "The car is in the garage", "the garage is made of wood" and "the garage has a door" are all examples of aggregation. The meaning of the aggregation is different in these three examples. If this meaning were kept, it would help to manage coherency in a VW. However, it is lost in C++ or Java languages. The impossibility to reuse child classes is another limitation of OO approach. The inheritance approach is based on the concept of programming by difference [Johnson88] where a child class specializes its parent class. However, this specialization can rarely be reused. For instance, in a VW containing a class of entities Chair and a class of entities Table, a chair with wheels is to be created. For this purpose, the VW designer can extend the Chair class by creating a WheelChair class. If a table with wheels has to be created, a natural approach would be to reuse the wheels in the WheelChair class. However, because of the intrinsic nature of inheritance, class WheelChair cannot be used outside the context of Chair. The only possible way to create a table with wheels is to cut and paste the code of the class WheelChair into a new class WheelTable, which is a subclass of Table and to adapt the code for the "wheel" property to the Table context. If the wheels were aggregated into the class of entities Chair, it would be easier to aggregate them into another class. A conceptual model blocking inheritance and allowing aggregation only could solve these problems.

## 2.5 Environment-oriented

This modelling approach places entities into environments (see Figure 5). As for all other modelling approaches, the environmental approach allows attributes and behaviour to be defined for and assigned to an entity. Contrary to the locale approach, the environment approach is more general than the simple topological inclusion relationship. The environmental inclusion relationship allows defining a common set of environmental attributes and behaviours to be defined for all entities belonging to a given environment. For instance, entities being part of the "Earth" environment will be subjected to Earth's gravity while entities located on Mars will obey to Mars' gravity law.

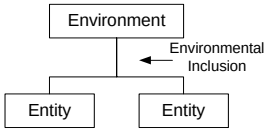


**Figure 5 Hierarchical environment-oriented representation of a virtual world.**

The environmental inclusion approach can also be extended to hierarchical entities (see Figure 6). In this case, an entity can act as an environment for another entity.

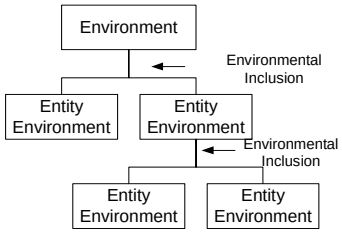


**Figure 6 Environment-oriented representation of a virtual world.**

The environmental inclusion approach is used at various levels in DEVA3 [Pettifer00], NPSNET [Macedonia94], Bamboo [Watsen98] and VEOS [Bricken93]. For instance, DEVA3 supports different behaviour assignment approaches: a behaviour can be specific to a single entity, to a group of entities, or can be shared by all entities belonging to an environment.

The environmental inclusion relationship (like most others previously mentioned kinds of relationships) is not sufficient to describe complex VWs. If more relationships were defined and recognized, it would be possible, for instance, to apply a dynamic chain solver when two or more links were connected together.

## 2.6 Relational graph-oriented

Relational graphs are a general modelling approach. In a relational graph, every entity in the VW is linked to the other entities ovia one or more relationships. Figure 7 shows a simple diagram of this modelling approach. When entities can be typed (e.g. classes of entities can be defined), the relational graph is a generalization of the previously described OOM approach.

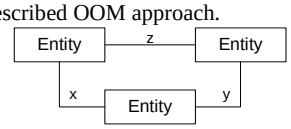


**Figure 7 Relational graph-oriented representation of a virtual world.**

URBI & ORBI [Fabre00], BrickNet [Singh95] and Daubrenet et al [Daubrenet00] are based on this approach. The first one, URBI & ORBI, defines a set of entity types but does not impose relationships. Relationships such as

"is composed of", "is adjacent to", or "activates" can be defined dynamically. BrickNet uses semantic relationships to link entities together. Depending of the configuration of the VW worlds, some behaviour can be enabled. For instance, as soon as both legs are attached to a robot, it can start to walk. This kind of behaviour can make easier the extensibility of the VW since the robot legs can be the result of an extension.

However, the global task of managing these relationships is not always or clearly defined as performed by the VW manager. Instead, this task is often divided into individual tasks to be managed by each involved entity. Consequently, the use of such a flexible modelling technique cannot guarantee the required characteristics.

## 3. CONCLUSION

The kind of conceptual model influences the level of reusability, extensibility and modularity of a VW. Unstructured (or ad hoc) conceptual models rely on the user to guarantee these characteristics. If the project is and remains small sized, it may be the most appropriate approach. For projects that are likely to be extended and constructed by modules, the development of a VW trying to improve these characteristics is essential. The appropriate kind of conceptual mode must be chosen carefully. If the object-oriented approach is selected, the level of reusability and other characteristics will be limited to the predefined set of classes. The relational graphs are really flexible but their generality makes them difficult to be understood by the framework (see cable example). If a VW cannot recognize a new relationship, it will not be able to manage its effect on existing entities of the VW. Standard integration rules (standard language) would help to define the interaction and the integration rules of entities populating a VW. Before a new entity is added, its integration rules would be defined.

Based on the lessons learned from existing kinds of conceptual models, many other approaches could be proposed. For instance, the APIA [Bernier00] proposes a kind of conceptual model with an integration language solving some of the problems encountered with the object-oriented and relational graph oriented VWs. This approach remains to be validated by real applications.

## 4. REFERENCES


[Bernier00] F. Bernier, D. Poussart, D. Laurendeau and M. Simoneau-Drolet. Interaction-Centric Modelling for Interactive Virtual Worlds: the APIA Approach. In *Proceedings of ICPR 2002*. Quebec, Canada. 2002.

[Bricken93] W. Bricken and G. Coco. *The VEOS Project.* Technical Report, Human Interface Technology Laboratory, University of Washington. 1993.

[Dahnmann99] J. Dahmann et al. HLA and Beyond: Interoperability Challenges. In *Simulation Interoperability Workshop*. Orlando, FL: IEEE. 1999.

[Daubrenet00] S. Daubrenet, S. Pettifer and A. West. Relationships: providing structure and behaviour for shared virtual environments. In *Proceedings of 7th UKVRSIG Conference*, pages 117-126. University of Strathclyde, Strathclye, Scotland. 2000

[Das97] T.K. Das, G. Singh, A. Mitchell, P.S Kumar, and K. McGee. NetEffect: A network architecture for large-scale multi-user virtual worlds. In *proceedings de la conference ACM Symposium on Virtual Reality Software and Technology (VRST 1997)*. pages 157-163. Lausanne, Suisse. 1997.

[Fabre00] Y. Fabre, G. Pitel, L. Soubrevilla, E. Marchand, T. Géraud and A. Demaille. A Framework to Dynamically Manage Distributed Virtual Environments. In *Proceedings of the Second International Conference on Virtual Worlds*, pages 54-64. Springer-Verlag. 2000

[Fishwick95] P. A. Fishwick. *Simulation Model Design and Execution: Building Digital Worlds*. Pearson Education POD, Prentice Hall, 1995.

[Fishwick96] P. A. Fishwick. Extending Object-Oriented Design for Physical Modeling. Submission for special issues of *ACM Transactions on Modeling and Computer Simulation.*

[Johnson88] A. Johnson, M. Roussos, J. Leigh, C. Barnes, C. Vasilakis and T. Moher. The NICE Project: Learning Together in a Virtual World. In *Proceedings of VRAIS '98*, pages 176-183. Atlanta, Georgia. 1998.

[Greenhalgh00] C.M. Greenhalgh, J. Purbrick, and D. Snowdown. Inside MASSIVE-3: Flexible support for data consistency and world structuring. In *Proceedings of the conference Collaborative Virtual Environments 2000*, pages 119-127. San Francisco, CA. 2000.

[Macedonia94] M. R. Macedonia, M. J. Zyda, D. R. Pratt, P. T. Barham and S. Zeswitz. NPSNET: A Network Software Architecture for Large Scale Virtual Environments, *Presence*, 3(4). 265–287. 1994.

[Mikhajlov98] L. Mikhajlov and E. Sekerinski. A Study of The Fragile Base Class Problem. In *Proceedings of the 12th European Conference on Object-Oriented Programming*, pages 355-382. Brussels, Belgium. 1998.

[OpenInvenor] Open inventor – SGI http://www.sgi.com /software/inventor/

[OOPM] OOPM - Object-Oriented Physical Modeling - University of Florida, http://www.cis.ufl.edu/~fishwick/tr/tr96-026.html

[Pettifer00] S. Pettifer, J. Cook, J. Marsh and A. West. DEVA3: Architecture for a Large-Scale Distributed Virtual Reality System, In *Proceedings of ACM Conference on Virtual Reality Software and Technology (VRST'00)*, pages 33-40. Seoul, Korea. 2000.

[Singh95] G. Singh, L. Serra, W. Png, A. Wong, and H. Ng. BrickNet: sharing object behaviors on the Net. In *Proceedings IEEE VRAIS '95*, pages 19-27. 1995.

[Singhal99] S. Singhal and M. Zyda. *Networked Virtual Environments - Design and Implementation.* Addison Wesley. 1999.

[Viskit] VisKit - Paradigm Entertainment, http://www.viskit.com

[Vrml97] VRML 97 - Web3D http://www.web3d.org/Specifications/VRML97/

[Watsen98] K. Watsen and M. Zyda. Bamboo - A Portable System for Dynamically Extensible, Realtime, Networked, Virtual Environments. In *Proceedings of the Virtual Reality Annual International Symposium (VRAIS'98)*, pages 139-146. Atlanta, GA. 1998.

[WTK] WorldToolkit - Sense8, http://www.sense8.com/products/wtk.h