The Design of a Virtual Environment Architecture

Manuel Fradinho Duarte de Oliveira        João Madeiras Pereira
Instituto Superior Técnico of Lisbon        Instituto de Engenharia e Sistemas de Computação
Rua Alves Redol, N°9 Apartado 13069, 1000 Lisboa, Portugal
manuel.oliveira@alfamicro.pt          jap@inesc.pt

1. Introduction

The exponential growth of the Internet has made our world a smaller place without borders. The continuous advances in communications, computer processing power and graphic capabilities has made immersive cyberspace an emerging reality.

Nowadays cyberspace exists in a crude form with the possibility of having virtual spaces where people come together to exchange ideas, work or socialise. The importance of the latter is proven by the wide adoption of on-line communities known as Multi-User Dungeons (MUD) [1].

Current cyberspace systems will gradually mature into systems with dynamic realities where people may interact with each other and their surrounding environment. These systems are designated as Large Scale Virtual Environments (LSVE) being implemented only with high-end equipment and high cost networks, therefore having a reduced user accessibility. Several factors related to the Internet and its technologies have provided the possibility of enabling common desktop computers with virtual windows to LSVE:

- The Virtual Reality Modelling Language (VRML) [2] presents itself as a standardised syntax for describing virtual worlds and associated behaviours with the embedded mechanisms for extension. Although VRML managed to introduce 3D on the World Wide Web (WWW) it does not provide means for LSVE infrastructures [3]. Several working groups exist focused on the design of VRML future extensions to provide what is necessary but it is common knowledge that the field is not mature enough to begin with standards on LSVE infrastructures.

- The Internet has been adopted as the network structure to support LSVE since it has world wide coverage and is the largest WAN available. Most of the insufficiencies currently detected in the protocol are being dealt with the revision of the standard [4]. The bandwidth insufficiency is being addressed by technology solutions such as Asymmetric Digital Subscriber Line (ADSL) and Low Earth Orbit (LEO) [5] satellites. Both technologies will pass the bottleneck from the bandwidth to elsewhere.

- Several communication protocols have been developed or are under development targeted to support LSVE such as Distributed Interactive Simulation (DIS) [6] and Virtual Reality Transport Protocol (VRTP) [7] amongst others.

- The Internet has fomented the emergence of languages with strong networking capabilities and cross-platform characteristics.

The remainder of the paper provides a brief description of Virtual Worlds, explaining the underlying concepts of the architecture design and some implementation details. The paper finalises with conclusions that states some of the future work to be done.

2. Virtual Worlds

Virtual Worlds is an open architecture that provides solutions to the common problems associated to LSVE, with the objective of equipping the common desktop computer with the capabilities of distributed virtual reality. Towards this end we adopt two very important premises that are always present:

- Independently of the amount of data a reality contains, it is impossible for any user to be omniscient at any given time.
- There are areas of interest, which represent a user's perception of the immediate reality. Anything beyond that reality is illusive and uncertain. It is possible for a user to have different auras corresponding to particular attributes such as sight and hearing.

These premises have a tremendous impact on any spatial model and consequently on the architecture of a LSVE. In Virtual Worlds a reality corresponds to a world, which is partition into regions taking into consideration the first premise. A similar approach is taken by SIMNET [8] that subdivides a simulation zone into hexagonal regions and each region has a multicast port where all the users communicate using the DIS protocol. However Virtual Worlds refines this region partitioning by employing aura management to each user. Aura research is reported in the literature, namely in systems such as Model, Architecture and System for Spatial Interaction in Virtual Environments (MASSIVE) [9,10] and Distributed Interactive Virtual Environment (DIVE) [11].
2.1 Functional Architecture

The functional architecture of Virtual Worlds is depicted in figure 1.

![Functional Architecture Diagram](image)

**Figure 1 - Functional Architecture**

The 3 different background shades in Figure 1 represent distinct tiers of the architecture:

- **World** – This corresponds to the reality in its entirety, residing on an object-relational database system, which is managed by a world server.
- **Regions** – The world is segmented into regions where each one is managed by a region manager. The information is maintained with a flat file database system but having topological structuring in memory, which provides efficient access and management. Each region manager conducts aura management with the aid of group managers and session managers.
- **Client** – The client may either be manipulated by an end-user or an autonomous agent with believable personality.

The following sections describe in further detail each of the components involved in the architecture.

2.2 Distributed Architecture

With the partitioning of the world into regions, Virtual Worlds resorts to the use of Common Object Request Broker Architecture (CORBA) [13] to provide the infrastructure for the distributed components as depicted in the block diagram of figure 2.

![Block Diagram of Distributed Architecture](image)

**Figure 2 – Block Diagram of Distributed Architecture**

2.2.1 World Server (WS)

The WS (figure 2) is the nucleus of the entire architecture responsible for the management and integrity of the data related to the world, region managers (RM) and users. The database used is object-relational due to the best trade-offs of applying object-orientation to mature relational databases.

The server has solely a passive relationship with client applications, which are subject to an authentication procedure by means of the Log component. If validation of the client succeeds then a Command component is attributed to the client, allowing the end-user to perform a series of off-line operations or to enter immersive mode. Changing an avatar’s attributes or adding registered users are examples of the available off-line operations. Naturally the security mechanisms determine the accessibility to operations defining 4 different types of users: administrator, power user, user, visitor.

As depicted in the above illustration the WS also handles the delegation of the management of a region to the appropriate region manager. This is possible through the RM Trader that keeps updated information of machine availability and the regions that are being managed. The process of a machine registering as a qualified RM requires off-line intervention just like in other WWW services such as Domain Name Service (DNS). Naturally authentication mechanisms are used to avoid attempts by rogue machines.
2.2.2 Region Manager (RM)

The first task of a RM (figure 2) when starting is announcing its availability to the WS though the RM Request component. After this action the RM only becomes active when the WS responds by delegating a region. This requires the RM to decompose the region and obtain the pre-processed data that will be the functioning basis for the Session Managers (SM) and the Group Managers (GM). When a client requests entrance to the region, the RM notifies the WS of the user entrance, liberating any remote resources that might have been pending.

When the user is the first to enter then a GM is instantiated with all the relevant data and the necessary multi-cast ports. On the other hand, if other users already exist in the region then calculations are performed to analyse the need of creating a new GM or delegating the user to an existing GM.

Every client is associated to a Session Manager (SM) that is in charge of maintaining the illusion of seamless reality, despite of any partitioning scheme.

2.2.3 Client

As stated previously there exist 4 main types of users, each with well-defined roles. Virtual Worlds allows a programmer to extend the base users should it be required. The current types of users are:

- **Administrator** – These users hold the maximum privileges and are responsible for the maintenance and integrity of a reality.
- **Power User** – These users have access to perform some system tasks such as adding and maintaining regions, objects and autonomous entities. The objective, behind the existence of these users, is to alleviate the workload of the administrators.
- **User** – This class corresponds to the average user that has full control of its avatar.
- **Visitor** – These users are non-registered in the system and thereby do not exist in the reality. Therefore they do not possess an embodiment and may be restricted from certain areas of a world. These users are designated Ghosts due to their ephemeral state.

After a user gains the validation with the world server he/she may interact with the system through the Command component. The user is dispatched to the RM responsible for the particular region where the he/she is located only when intent of entering the environment is presented. Once the information concerning the GM and SM is available, the user can start interacting with the environment.

2.3 Aura Management

Aura Management is done on a region basis by means of a GM and the clients with their associated SM.

2.3.1 Group Manager (GM)

The GM is responsible for creating multicast ports for each aura that is created. In Virtual Worlds only the sight aura is considered however due to efficiency requirements it was necessary to separate the motion updates and object updates into two independent multicast ports. In addition two more ports were necessary for text transmission and application sharing control.

The GM is responsible for the consistency of the area it manages and has a Group Monitor that is the default owner of all existing objects. When a user picks up an object then the Group Monitor relinquishes ownership to the corresponding client. This is important since only an object’s owner determines its state, guaranteeing overall consistency. This avoids clients having objects with contradictory information.

2.3.2 Session Manager (SM)

There is a relationship of one-to-one between the SM and the corresponding client. It is responsible for monitoring the user’s navigation and triggering the necessary commands to remotely control the scene graph of the client through the usage of Line Of Sight algorithms and pre-processed data with regard to the corresponding VRML data. This requires the SM to hook up on the multicast port of the corresponding GM concerning motion updates. The control messages are then sent to the client indicating what is the command to be executed. Communication between the SM and the client is done with both UDP and TCP. The former is dedicated for light-weight interactions such as commands while the latter is dedicated for heavy-weight interactions such as downloading of more VRML data.

The SM is also responsible to indicate to the client when to change GM, updating the information with regard to communication ports.

For the tasks mentioned above it is important for the SM to use predictive algorithms that compensate the downloading times and makes navigation a smooth process.

2.3.3 Client

The client contributes to aura management by emitting periodic state updates that are necessary for the SM and other clients. However to avoid inundating the network, dead-reckoning algorithms have been adapted to handle situations that are less predictable than simulations as is the case of the DIS protocol.

3. Conclusions

We believe that Virtual Worlds through the conjunction of world partitioning and aura management make LSVE a reality for the desktop computer, using Java as the underlying language.
The following paragraphs indicate some of the immediate future work:

- Currently the area that is attributed to a Group Manager is statically determined off-line based on pre-processed information. We foresee the possibility of making this process dynamic.
- Virtual Worlds does not allow dynamic objects to reside in overlapping areas near adjacent region frontiers. This puts a necessary constraint in world building, which we plan to remove in the near future.
- At the time of writing, the system is being developed on a LAN due to the inability of the Internet to support multicast. However we plan to develop multicast software bridges until the next generation of the Internet protocol becomes effective.
- Currently we have adopted the use of the External Authoring Interface (EAI) [16], however we foresee the possibility of adopting Java3D [17] for visualisation.

REFERENCES