Analyzing the Behavior of Distributed Virtual Environment Systems

P. Morillo, J. M. Orduña, M. Fernández , J. Duato

Resumen— **Distributed Virtual Environment systems have experienced a spectacular growth last years. However, there are several key issues in these systems that should still be improved in order to design a scalable and cost-effective DVE system. One of the key issues in the design of scalable and cost-effective DVE systems is the partitioning problem. This problem consists of efficiently assigning clients (3-D avatars) to the servers in the system, and some techniques have been already proposed for solving it.**

In this paper, we propose the correlation of the quality function proposed in the literature for solving the partitioning problem with the performance of DVE systems. Since the results show an absence of correlation, we also propose the experimental characterization of DVE systems. The results show that the reason for that absence of correlation is the non-linear behavior of DVE systems with the number of avatars in the system. Also, the results show that workload balancing mainly has an effect on system throughput, while minimizing the amount of inter-server messages mainly has an effect on system latency.

Palabras clave—**Distributed Computing, Distributed Virtual Environment, Collaborative Environments.**

I. INTRODUCTION

PR ofessional high performance graphic cards currently offer a very good frame-rate for rendering complex 3D scenes in real time. On other hand, fast Internet connections have become worldwide available at a relatively low cost. These two factors have made possible the current growth of Distributed Virtual Environment(DVE) Systems. These systems allow multiple users, working on different computers that are interconnected through different networks (and even through Internet) to interact in a shared virtual world. This is achieved by rendering images of the environment as if they were perceived by the user. Each user is represented in the shared virtual environment by an entity called *avatar*, whose state is controlled by the user input. Since DVE systems support visual interactions between multiple avatars, every change in each avatar must be propagated to the rest of the avatars in the shared virtual environment.

DVE systems are currently used in different applications [13], such as collaborative design [12], civil and military distributed training [9], e-learning [11] or multi-player games [6]. Due to their different requirements, communication rate of avatars may significantly differ among these different applications. On other hand, DVE systems are inherently hetero-geneous. Effectively, each avatar can be controlled by a different computer system, with different resources. Additionally, both clients and servers can be interconnected through different networks using different technologies. Such heterogenous systems must deal with high communication and computational requirements as the number of avatars in the DVE system increases.

One of the key issues in the design of a scalable DVE system is the *partitioning problem* [7]. It consists of efficiently assigning the workload (avatars) among different servers in the system. The partitioning problem may seriously affect the overall performance of the DVE system, since it determines not only the workload each server is assigned to, but also the inter-server communication requirements (and therefore the network traffic). Some methods for solving the partitioning problems have been already proposed [7], [8], [13]. However, there are still some features in the proposed methods that can be improved. For example, in a previous paper we proposed a heuristic search method for solving the partitioning problem, instead of using an ad-hoc solution [10]. On other hand, the quality function proposed in the literature must be correlated with system performance, in order to design actually scalable and efficient partitioning strategies.

In this paper, we present the experimental correlation of the quality function proposed in the literature with the performance of DVE systems. Since the results show an absence of correlation, we also propose the experimental characterization of DVE systems, in order to analyze the reasons for that absence of correlation. This characterization study measures the impact of different parameters of the quality function on the performance of DVE systems, and it shows that the behavior of DVE systems is non-linear with the number of avatars in the system. Therefore, in order to design an actually scalable partitioning method for DVE systems, a different (non-linear) partitioning strategy must be proposed.

The rest of the paper is organized as follows: Section II describes the partitioning problem and the existing proposals for solving it. Section III details the proposed characterization setup that allows to experimentally study the behavior DVE systems. Next, Section IV presents the correlation and evaluation results. Finally, Section V presents some concluding remarks and future work to be done.

Departamento de Informática, Universidad de Valencia. e-mail: `Juan.Orduna.@uv.es`.

J. Duato belongs to DISCA, U. Politécnica de Valencia. e-mail: `jduato@gap.upv.es`

II. The Partitioning Problem in DVE Systems

Architectures based on networked servers are becoming a de-facto standard for DVE systems [13], [8]. In these architectures, the control of the simulation relies on several interconnected servers. Multi-platform client computers join the DVE system when they are connected to one of these servers. When a client modifies an avatar, it also sends an updating message to its server, that in turn must propagate this message to other servers and clients. Servers must render different 3D models, perform positional updates of avatars and transfer control information among different clients. Thus, each new avatar represents an increasing in both the computational requirements of the application and also in the amount of network traffic. When the number of connected clients increases, the number of updating messages must be limited in order to avoid a message outburst. In this sense, concepts like areas of influence (AOI) [13], locales [1] or auras [4] have been proposed for limiting the number of neighboring avatars that a given avatar must communicate with.

Depending on their origin and destination avatars, messages in a DVE system can be intra-server or inter-server messages. Figure1 show an example of both kinds of messages. Intra-server messages are those messages whose origin and destination avatars (client computers) are both assigned (connected) to the same server. This kind of messages only need resources of a single server. Inter-server messages are those messages whose origin and destination avatars (client computers) are assigned (connected) to different servers. Inter-server messages need resources of several servers.
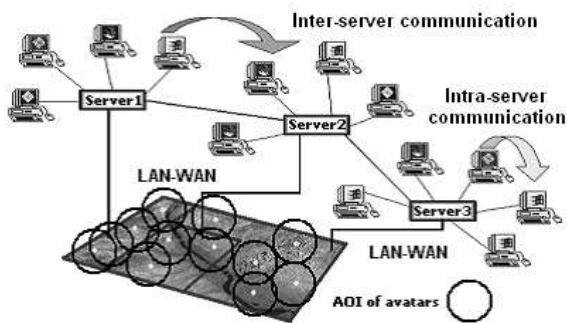

Fig. 1. Multi-server DVE system

Lui and Chan have shown the key role of finding a good assignment of clients to servers in order to ensure both a good frame rate and a minimum network traffic in DVE systems [8]. They propose a quality function, denoted as C_p, for evaluating each assignment of clients to servers. This quality function takes into account two parameters. One of them consists of the computing workload generated by clients in the DVE system, denoted as C_p^W. In order to minimize this parameter, the computing workload should be proportionally shared among all the servers in the DVE system, according to the computing resources

of each server. The other parameter of the quality function consists of the overall inter-server communication requirements, denoted as C_p^L. In order to minimize this parameter, avatars sharing the same AOI should be assigned to the same server. Quality function C_p is defined as

$$C_p = W_1\, C_p^W + W_2\, C_p^L \qquad (1)$$

where $W_1 + W_2 = 1$. W_1 and W_2 are two coefficients that weight the relative importance of the computational and communication workload, respectively. These coefficients should be tuned according to the specific features of each DVE system. Using this quality function (and assuming $W_1 = W_2 = 0.5$) Lui and Chan propose a partitioning algorithm that re-assigns clients to servers [8]. The partitioning algorithm should be periodically executed for adapting the partition to the current state of the DVE system (avatars can join or leave the DVE system at any time, and they can also move everywhere within the simulated virtual world). Lui and Chan also have proposed a testing platform for the performance evaluation of DVE systems, as well as a parallelization of the partitioning algorithm [8].

The partitioning method proposed by Lui and Chan currently provides the best results for DVE systems. However, the correlation of the quality function C_p with DVE system performance should be studied, and parameters W_1 and W_2 must be properly tuned. In this sense, the characterization of DVE systems is crucial in order to design partitioning strategies that actually improves the scalability of DVE systems.

III. Characterization Setup

We propose the characterization of generic DVE systems by simulation. The evaluation methodology used is based on the main standards for modeling collaborative virtual environments, FIPA [3], DIS [2] and HLA [5]. We have developed a simulation tool (a program written in C++) that models the behavior of a generic DVE system with a network-server architecture. Concretely, we have implemented a set of multi-threaded servers. Each thread in a server uses blocking sockets for communicating with a client. Each client simulates the behavior of a single avatar, and it is also implemented as a multi-threaded application. One of the threads of the client manages the communication with the server assigned to that client, and another thread manages user information (current position, network latency, etc.).

Our simulator model is composed of a set of S interconnected servers and n avatars. Following the approach specified in FIPA and HLA standards, one of the servers acts as the main server (called *Agent Name Service* [3] or *Federation Manager* [5]) and manages the whole system. The main server also maintains a partitioning file for assigning a given server to each new avatar. In this way, once the network address and the port number where the main server is listening are known, avatars can join the

simulation through this main server, that assigns each new avatar to one of the servers in the system. At this point, the new avatar must connect with the assigned server in order to start the simulation.

Avatars can join the simulation through this main server, that assigns each new avatar to one of the servers in the system. In each simulation, all avatars sharing the same AOI must communicate between them for notifying their position in the 3D virtual world.The message structure used for notifying avatar movements is the *Avatar Data Unit (ADU)* specified by DIS [2]. For the purpose of characterization of DVE systems, each client only simulates a given rate of avatar movements through the virtual world, and assumes that no changes are produced in any element of the AOI. This simplifying assumption reduces the system workload, but does not change the behavior of the system.

A simulation consists of each avatar performing 100 movements, at a rate of one movement every 2 seconds. Each time an avatar performs a movement, he notifies that movement to his server by sending a message with a timestamp. That server must then notify that movement to all the avatars in the same AOI of the sender avatar. When that notification arrives to these avatars, they return an ACK message to the server, that in turn propagates that ACK messages to the sender avatar. When an ACK message arrives, the sender avatar computes the round-trip delay for communicating with each neighbor avatar. We have denoted this round-trip delay (measured in real-time) as the *system response*. When a simulation ends, each avatar has computed the average system response for the avatars in its AOI. At this point, all avatars send these average system responses to their respective servers, and each server then computes its own average system response. Finally, starting from the average value provided by each server the main server computes the average system response for the entire system, denoted as the *average system response* (ASR). An ASR value is computed after each simulation. An actually scalable DVE system must keep this measure as low as possible as the number of avatars in the system increases. On other hand, the system throughput is given by the maximum number of avatars that the system can manage while keeping the ASR below a certain threshold. Therefore, we have considered ASR as the main performance measure for characterizing the behavior of DVE systems.

In order to evaluate the performance of each partitioning method, usually 3 different distributions of avatars in the virtual world are proposed in the literature: uniform, skewed and clustered distributions of avatars [8], [10]. The distributions of avatars in the virtual world determine the workload that each avatar adds to his server. However, both the movement rate and also the AOI of avatars can also be adjusted in order to make that workload to be independent of the distribution of avatars in the virtual world. Therefore, for the sake of simplicity we have

only considered a uniform distribution of avatars, with the same AOI and the rate of movements for all the avatars in the system.

IV. SIMULATION AND CORRELATION RESULTS

In this section, we present the correlation and simulation results obtained for the DVE model described in the previous section. The hardware platform used for simulating both clients and servers in the DVE system have been PC's with processor Pentium IV at 1.7 GHz, with 256 Mbytes of RAM and with NVidia MX-400 graphic cards. Each server has been implemented in a single PC, while up to 50 clients have been allocated in the same PC. We have used a 10 Mbps Ethernet as the interconnection network. The simulation tool has been executed on a Windows 2000 Professional operating system.

We have tested a great number of different DVE configurations, ranging from small virtual worlds (3 servers and 180 avatars) to large virtual worlds (900 avatars and 6 servers). Since we have obtained very similar results in both of them, we present the results for a small DVE configuration.

Figure 2 shows the performance results for a small DVE system composed of 3 servers and 180 avatars when different partitions (showing different values of the quality function C_p) are simulated. This Figure shows on the X-axis the values of C_p obtained for different partitions (assignments of avatars to the servers in the system). The Y-axis shows ASR values for the simulations performed with these partitions. Each point in the plot represents the average value of the ASR obtained after 30 simulations of the same DVE system. The standard deviation for any of the points shown in the plot was not higher than 25 ms. in any case.
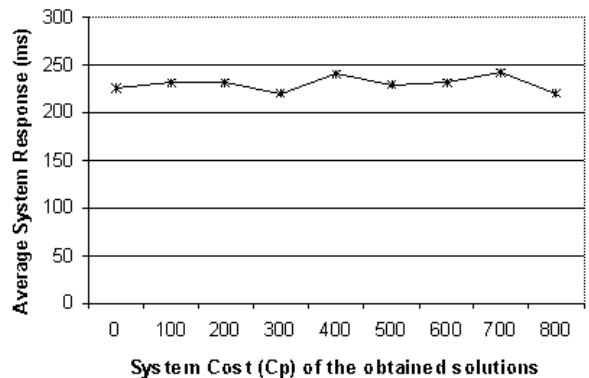

Fig. 2. Correlation of quality function C_p with average system response

This Figure clearly shows that C_p does not correlate with DVE system performance. ASR ranges in a 5% of its initial value (sometimes decreasing) as C_p values are greatly increased. Thus, for example, the ASR value obtained for a partition with a value of C_p equal to 800 (the worst value of C_p) is even lower (better) than the ASR values obtained for partitions with C_p values ranging from 400 to 700. There not exists a linear correspondence between the values of

the quality function C_p and ASR. Therefore, a characterization study is needed in order to capture the behavior of DVE systems.

In order to make a methodical characterization study, we have studied the behavior of DVE systems as the two terms of the sum in equation 1 vary. Given a DVE system composed of 3 servers, first we have optimized the value of C_p^W (workload balancing) and we have simulated both a partition with optimum (minimized) value of C_p^L (inter-server communication requirements) and also a partition with the worst value of C_p^L as possible. Also, for the same DVE system we have unbalanced the partitions to obtain the worst value of C_p^W as possible, performing the same simulations with the partitions showing both the best and the worst values of the term C_p^L.

Figure 3 shows the obtained ASR values for a DVE configuration of 3 servers as the number of avatars in the system increases. The value of C_p associated to the simulated partitions was zero in all the cases (perfect workload balancing and absence of inter-server messages, that is, C_p^L and C_p^W equal to zero). ASR seems to be invariant with the number of avatars (ASR plot has the shape of a flat line) until it reaches a saturation point. From that saturation point, ASR values greatly increase as new avatars are added to the simulation. These results clearly show that the behavior of DVE system is non linear with the number of avatars in the system. Since C_p is defined (equation 1) as a linear function of both workload balancing and inter-server communications, this non-linear behavior can explain the absence of correlation shown in Figure 2.
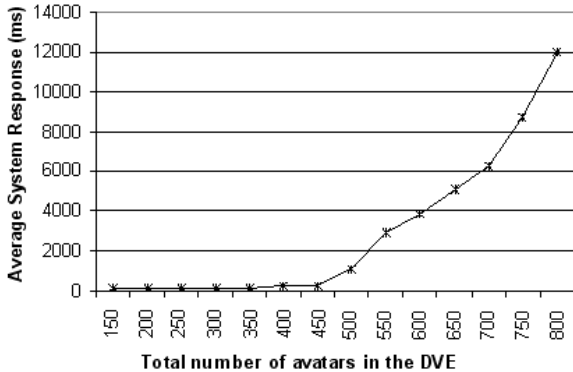

Fig. 3. Average system response as the number of avatars per server increases

In order to determine the reason for the non-linear behavior of DVE systems, Table I shows the CPU utilization and the average system response in milliseconds (SR-Sx) obtained for each server corresponding to the simulation results shown in Figure 3. This table shows that the DVE system reaches the saturation point when *any* of the servers in the system reaches a CPU utilization of 100%. All the SR-Sx values increase as the respective CPU utilization so does, and there are not significant differences between the three SR-Sx values while the CPU utilization remains under 99% (simulation with 500

avatars). When CPU utilization reaches more than 99% in any of the servers then the system response for that server greatly increases, also increasing ASR in the rest of the servers accordingly, as well as global ASR. Therefore, we can conclude that the non-linear behavior of DVE systems shown in Figure 3 is due to the limit of 100% in any CPU utilization. Since quality function C_p does not take into account CPU utilization in order to measure the quality of a partition, it cannot take into account the non-linear behavior of DVE systems as the number of avatar increases.

Avatars	%CPU1	%CPU2	%CPU3
350 (116,117,117)	51	54	52
400 (133,133,134)	67	69	71
450 (150,150,150)	78	76	78
500 (166,167,167)	96	99	100
550 (183,183,184)	100	100	100
600 (200,200,200)	100	100	100
650 (216,217,217)	100	100	100
Avatars	**ASR-S1**	**ASR-S2**	**ASR-S3**
350 (116,117,117)	193.21	191.81	196.43
400 (133,133,134)	249.29	245.60	247.36
450 (150,150,150)	279.43	281.12	280.91
500 (166,167,167)	545.73	892.54	1775.85
550 (183,183,184)	3021.44	2811.45	2982.12
600 (200,200,200)	4356.91	3645.21	3409.20
650 (216,217,217)	5320.65	5114.32	5112.22

TABLA I

AVERAGE SERVER RESPONSES AND CPU UTILIZATION IN SIMULATIONS SHOWN IN FIGURE 3

These results means that although a partition can provide perfect workload balancing and minimum inter-server communication requirements (thus providing a optimum value of C_p, as the partitions in Figure 3 do), ASR will be very poor if the amount of avatars assigned to any server requires a CPU utilization of 99% or more. Also, given a DVE system and given a certain amount of avatars in the system, the performance of that DVE system will remain practically invariant with C_p if none of the servers reaches a CPU utilization of 99%. This is the case for the simulations whose results are shown in Figure 2, where only 90% of CPU utilization was reached in the worst case.

Figure 3 shows the behavior of the DVE with the best partitions as possible. Table II shows the same measurements of Table I taken on the same DVE when simulating partitions with the worst C_p^L as possible (the maximum number of inter-server messages). When comparing these two tables in a line-by-line analysis, we can see that a reduction in the amount of inter-server messages (Table I) results in a fixed reduction in ASR while the system is not saturated (the difference between ASR values of both tables in the first 3 lines remains constant, about 40 ms.). However, the effect of inter-server messages in ASR becomes more important as servers approach saturation. Also, we can see that for the same number of avatars Table II shows higher CPU utilization. Therefore, in order to maximize the throughput of a DVE system (the amount of avatars a given DVE system can manage without enter saturation) inter-

server messages must also be minimized.

Avatars	%CPU1	%CPU2	%CPU3
350 (116,117,117)	61	64	65
400 (133,133,134)	76	75	79
450 (150,150,150)	80	82	83
500 (166,167,167)	98	100	100
550 (183,183,184)	100	100	100
600 (200,200,200)	100	100	100
650 (216,217,217)	100	100	100
Avatars	SR-S1	SR-S2	SR-S3
350 (116,117,117)	221.45	214.62	217.81
400 (133,133,134)	281.41	291.31	289.91
450 (150,150,150)	350.13	333.73	342.39
500 (166,167,167)	942.45	1792.54	1675.85
550 (183,183,184)	3674.56	3844.17	3364.22
600 (200,200,200)	4991.34	4565.71	4398.83
650 (216,217,217)	6541.34	6199.45	5853.37

TABLA II

AVERAGE SERVER RESPONSES AND CPU UTILIZATION WHEN C_p^L IS MAXIMIZED

Additionally, Tables III and IV show CPU utilization and server system responses for the same DVE system (composed of 3 servers) when the term C_p^W is maximized (worsened). The partitioning strategy whose results are shown in Table III minimizes inter-server communication requirements (the term C_p^L), while the partition whose results are shown in Table IV maximizes this term. In a line-by-line comparison of both tables, we can see that CPU utilization are very similar in both tables. This result shows that the term C_p^L does not have any effect on the system throughput when the term C_p^W is worsened (the system reaches saturation with 425 avatars in both tables). However, the system responses obtained for each server greatly differs between the two tables.

No. of avatars	%CPU1	%CPU2	%CPU3
375 (125,125,125)	72	74	71
400 (125,150,125)	71	81	73
425 (125,175,125)	73	100	70
450 (125,200,125)	72	100	69
475 (125,225,125)	70	100	72
No. of avatars	SR-S1	SR-S2	SR-S3
375 (125,125,125)	221.23	233.57	225.71
400 (125,150,125)	219.36	285.21	218.77
425 (125,175,125)	228.74	1725.84	235.54
450 (125,200,125)	225.67	6308.87	218.99
475 (125,225,125)	226.54	11554.86	221.42

TABLA III

AVERAGE SERVER RESPONSES AND CPU UTILIZATION WHEN C_p^L IS MINIMIZED

A comparison of Tables III and IV with Tables I and II shows that minimizing the term C_p^W results in improving the throughput (the scalability) of DVE systems. Effectively, the same DVE system reaches saturation with 500 avatars in Tables I and II, while reaching saturation with only 425 avatars in Tables III and IV. However, there not exists any significant difference in ASR values between Tables I and Tables III in the first two lines, neither between Ta-

bles II and IV on the first two lines. These results show that the term C_p^W does not have an important effect on ASR. On the contrary, the term C_p^L mainly affects to ASR values (system latency), and this effect is amplified when the term C_p^W is worsened. However, if the term C_p^W is optimized then the impact of C_p^L in ASR is not significant.

No. of avatars	%CPU1	%CPU2	%CPU3
375 (125,125,125)	69	72	73
400 (125,150,125)	71	83	72
425 (125,175,125)	69	100	73
450 (125,200,125)	72	100	70
475 (125,225,125)	71	100	72
No. of avatars	SR-S1	SR-S2	SR-S3
375 (125,125,125)	217.89	241.13	233.45
400 (125,150,125)	251.41	323.22	218.31
425 (125,175,125)	1290.21	2992.85	1432.80
450 (125,200,125)	5678.51	8861.91	6007.82
475 (125,225,125)	9998.83	16873.21	10254.31

TABLA IV

AVERAGE SYSTEM RESPONSE AND CPU UTILIZATION FOR EACH SERVER WHEN C_p^L IS MAXIMIZED

V. CONCLUSIONS

In this paper, we have proposed the experimental correlation of the quality function proposed in the literature for solving the partitioning problem with the performance of DVE systems. The results show a total absence of correlation. Therefore, we have also proposed a characterization study of DVE systems.

DVE systems show a non-linear behavior with the number of avatars in the system. Average system response (round trip-delay of messages notifying movements of avatars) remains practically invariant with the number of avatars in the system until the DVE system reaches a saturation point. This saturation point is given by the limit of a CPU utilization of 100% in any of the DVE servers. When this limit is reached, then average system response greatly increases as new avatars are added to the system. The reason for this behavior is that the saturated server becomes the system bottleneck, regardless of the state of the other servers.

We have also studied the effects of the two terms of the quality function proposed in the literature (workload balancing and the amount of inter-server messages) on the performance of DVE systems. The results show that workload balancing mainly has an effect on system throughput, while the amount of inter-server messages mainly has an effect on system latency. However, if none of the servers reaches saturation then the amount of inter-server messages does not have any significant effect on system latency, regardless of the amount of inter-server messages. Therefore, in order to design an efficient and scalable DVE system, the partitioning method should be targeted to balance the workload among the servers in the system in such a way that none of them reaches 100% of CPU utilization.

REFERENCIAS

[1] D.B.Anderson, J.W.Barrus, J.H.Howard, "Building multi-user interactive multimedia environments at MERL", in *IEEE Multimedia*, 2(4), pp.77-82, Winter 1995.

[2] DIS. 1278.1 IEEE Standard for Distributed Interactive Simulation-Application Protocols (ANSI). DMSO. DoD High Level Architecture. 1997.

[3] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000. Available at `http://www.fipa.org/specs/fipa00023/`

[4] J.C.Hu, I.Pyarali, D.C.Schmidt, "Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance Over High-Speed Networks", *Proc. of the 2nd. IEEE Global Internet Conference*, November.1997.

[5] Kuhl, F., Weatherly, R, Dahmann, J., "Creating Computer Simulation Systems: An Introduction to the High Level Architecture", Prentice-Hall PTR, Upper Saddle River, NJ, 1999.

[6] Michael Lewis and Jeffrey Jacboson, "Game Engines in Scientific Research", in *Communications of the ACM*, Vol 45. No.1, January 2002.

[7] John C.S. Lui, M.F.Chan, Oldfield K.Y, "Dynamic Partitioning for a Distributed Virtual Environment", *Department of Computer Science*, Chinese University of Hong Kong, 1998.

[8] Jonh C.S. Lui, M.F. Chan, "An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems", *IEEE Trans. Parallel and Distributed Systems*, Vol. 13, March 2002

[9] D.C.Miller, J.A. Thorpe, "SIMNET: The advent of simulator networking", in *Proceedings of the IEEE*, 83(8), pp. 1114-1123. August, 1995.

[10] P. Morillo, M. Fernández, J.M. Orduña "An ACS-Based Partitioning Method for Distributed Virtual Environment Systems", *Proc. of 2003 Int. Parallel and Distributed Processing Symposium Workshops (IPDPS' 2003)*, Nice, France. April, 2003.

[11] Tohei Nitta, Kazuhiro Fujita, Sachio Cono, "An Application Of Distributed Virtual Environment To Foreign Language", in *IEEE Education Society*, October 2000.

[12] J.M.Salles Dias, Ricardo Galli, A. C. Almeida et al. "mWorld: A Multiuser 3D Virtual Environment", in *IEEE Computer Graphics*, Vol. 17, No. 2, March-April 1997.

[13] S.Singhal, and M.Zyda, "Networked Virtual Environments", *ACM Press, New York, 1999*.