On the Characterization of Distributed Virtual Environment Systems

P. Morillo[1], J. M. Orduña[2] and J. Duato[3]

[1] Instituto de Robótica. Universidad de Valencia. SPAIN.
[2] Departamento de Informática. Universidad de Valencia. SPAIN
[3] DISCA. Universidad Politécnica de Valencia. SPAIN
Pedro.Morillo@uv.es, Juan.Orduna@uv.es, jduato@gap.upv.es

Abstract. Distributed Virtual Environment systems have experienced a spectacular growth last years. 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 experimental characterization of DVE systems. The evaluation results show that there exists an absence of correlation between the quality function proposed in the literature for solving the partitioning problem and the performance of DVE systems. The reason for that absence of correlation is the non-linear behavior of DVE systems with the number of avatars in the system. Also, we have studied the effects of the two terms of the quality function proposed in the literature on the performance of DVE systems. 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. Therefore, in order to design actually scalable DVE systems, the quality function used for solving the partitioning problem must be properly redefined.

1 Introduction

Professional 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 many different applications [14], such as collaborative design [13], civil and military distributed training [10], e-learning [12] or multi-player games [7].

One of the key issues in the design of a scalable DVE system is the *partitioning problem* [8]. It consists of efficiently assigning the workload (avatars) among different

servers in the system. The partitioning problem determines the overall performance of the DVE system, since it has an effect not only on the workload each server in the system is assigned to, but also on the inter-server communications (and therefore on the network traffic). Some methods for solving the partitioning problem have been already proposed [8, 9, 14]. These methods provide efficient solutions even for large scale DVE systems. However, there are still some features in the proposed methods that can be improved. For example, different heuristic search methods can be used for finding the best assignment of clients to servers, instead of using ad-hoc heuristics. Moreover, the quality function proposed in the literature must be correlated with system performance, in order to design actually scalable and efficient partitioning strategies.

In previous papers, we proposed a heuristic search method for solving the partitioning problem [11]. 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) quality function must be proposed.

The rest of the paper is organized as follows: Section 2 describes the partitioning problem and the existing proposals for solving it. Section 3 details the proposed characterization setup that allows to experimentally study the behavior DVE systems. Next, Section 4 presents the correlation and evaluation results. Finally, Section 5 presents some concluding remarks and future work to be done.

2 The Partitioning Problem in DVE Systems

Architectures based on networked servers are becoming a de-facto standard for DVE systems [14, 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 avoid a message outburst. In this sense, concepts like areas of influence (AOI) [14], locales [1] or auras [5] 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. In order to design a scalable DVE systems, the number of intra-server messages must be maximized. Effectively, when clients send intra-server messages they only concern a single server. Therefore, they are minimizing

the computing, storage and communication requirements for maintaining a consistent state of the avatars in a 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, 9]. 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 [9]. The partitioning algorithm should be periodically executed for adapting the partition to the current state of the DVE system as it evolves (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 [9].

Some other approaches for solving the partitioning problem have been also proposed. One of them groups avatars following regular distributions [2]. In order to ensure good performance, this algorithm generate a number of regular distributions equal to the number of servers in the DVE system. However, this proposal does not obtain good performance when avatars are located following a non-uniform distribution. Another different approach rejects dynamic concepts associated to avatars like AOI, aura or locale [15]. Although this approach provides a fast way of solving the partitioning problem, the performance of the static partitioning is quite low when avatars show a clustered distribution. In this case, the servers controlling the areas of the clusters are overloaded, increasing the overall cost of the quality function.

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.

3 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 [4], DIS [3] and HLA [6]. 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 it is assigned to, 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* [4] or *Federation Manager* [6]) 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, 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.

In each simulation, all avatars sharing the same AOI must communicate between them for notifying both their position in the 3D virtual world and also any change in the state of the elements in that AOI. 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. The message structure used for notifying avatar movements is the *Avatar Data Unit (ADU)* specified by DIS [3].

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 the server he is attached to 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 the servers then computes the average system response for each server. Finally, the main server computes the *average system response* (ASR) for that simulation. A actually scalable DVE system must keep this measure as low as possible as the number of avatars in the system increases. Also, the system throughput is given by the maximum number of avatars the system can manage while keeping the system response 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 [9, **?**]. The distributions of avatars in the

virtual world determine the workload that each avatar adds to the server where that avatar is assigned. 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 same movement rate for all the avatars.

4   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 for both clients and servers in the DVE system are 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 each PC. We have used a 10 Mbps Ethernet as the interconnection network. This 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 (composed of 3 servers and 180 avatars) to large virtual worlds (composed of 900 avatars and 6 servers). However, we have obtained very similar results in all of them. Due to space limitations, we present in this section the results for a small DVE configuration.

   Figure 1 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.

   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.

   Since the purpose of solving the partitioning problem is to achieve scalable DVE systems, we have studied the performance of DVE systems (ASR values) as the number of avatars increases. On other hand, in order to make a methodical characterization study, we have considered the two terms of the sum in equation 1. Given a DVE system composed with 3 servers, first we have optimized the value of C_p^W (workload balancing) (C_p^W) 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
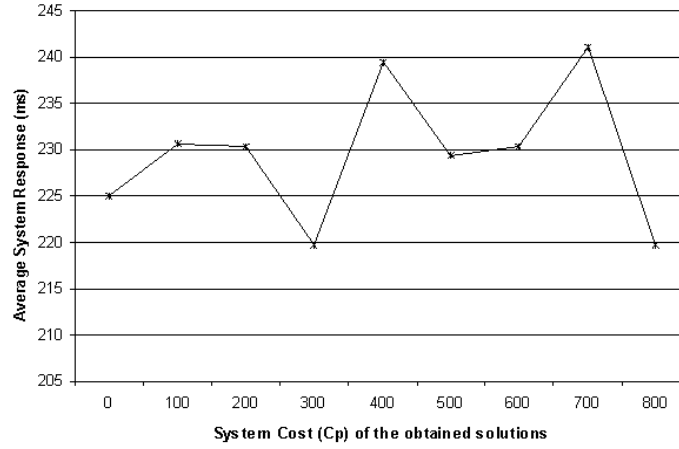
Fig. 1. Correlation of quality function C_p with average system response

obtain the worst value of C_p^W as possible, repeating the simulations for the partitions best and the worst cases of the term C_p^L.

Figure 2 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 1.

In order to determine the reason for the non-linear behavior of DVE systems, Table 1 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 2, as well as the corresponding global ASR value. 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 95% (simulation with 500 avatars). When CPU reaches more than 95% in any of the servers then the system response for that server greatly increases, also increasing global ASR accordingly. Therefore, we can conclude that the non-linear behavior of DVE systems shown in Figure 2 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.
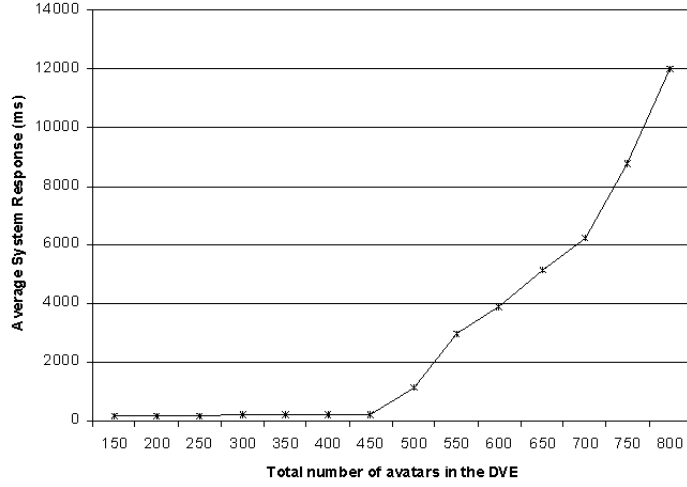
Fig. 2. Average system response as the number of avatars per server increases

No. of avatars	%CPU1	%CPU2	%CPU3	SR-S1	SR-S2	SR-S3	Average SR
350 (116,117,117)	51	54	52	193.21	191.81	196.43	193.82
400 (133,133,134)	67	69	71	249.29	245.60	247.36	247.17
450 (150,150,150)	78	76	78	279.43	281.12	280.91	280.49
500 (166,167,167)	96	99	100	545.73	892.54	1775.85	1138.04
550 (183,183,184)	100	100	100	3021.44	2811.45	2982.12	2938.34
600 (200,200,200)	100	100	100	4356.91	3645.21	3409.20	3803.77
650 (216,217,217)	100	100	100	5320.65	5114.32	5112.22	5182.40

Table 1. Average system responses and server CPU utilization in simulations shown in Figure 2

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 2 do), ASR will be very poor if the amount of avatars assigned to any server requires a CPU utilization of 95% 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 95%. This is the case for the simulations whose results are shown in Figure 1. Although we have not shown here that results due to space limitations, CPU utilization results for that simulations show that only 90% of CPU utilization is reached in the worst case.

Figure 2 shows the behavior of the DVE with the best partitions as possible. Table 2 shows the same measurements of Table 1 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 1) 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 the servers get more and more overloaded. Also, we can see that for the same number of avatars Table 2 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.

No. of avatars	%CPU1	%CPU2	%CPU3	SR-S1	SR-S2	SR-S3	Average SR
350 (116,117,117)	61	64	65	221.45	214.62	217.81	217.96
400 (133,133,134)	76	75	79	281.41	291.31	289.91	287.94
450 (150,150,150)	80	82	83	350.13	333.73	342.39	342.08
500 (166,167,167)	98	100	100	942.45	1792.54	1675.85	1470.28
550 (183,183,184)	100	100	100	3674.56	3844.17	3364.22	3627.68
600 (200,200,200)	100	100	100	4991.34	4565.71	4398.83	4651.96
650 (216,217,217)	100	100	100	6541.34	6199.45	5853.37	6198.05

Table 2. Average system responses and server CPU utilization in simulations shown in Figure 2

Additionally, Tables 3 and 4 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 3 minimizes inter-server communication requirements (the term C_p^L), while the partition whose results are shown in Table 4 maximizes this term. In a line-by-line comparison of both tables, we can see that CPU utilization are very similar in both tables. Thus, 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. When a given server reaches saturation (third line, server S2), if the amount of inter-server messages are minimized (Table 3) avatars assigned to other servers do not use this server, and therefore this saturation does not affect the average latencies obtained for servers S1 and S3. On the contrary, if C_p^L is maximized (Table 4) then the saturated server is used by avatars assigned to other servers, and therefore system responses for the rest of the servers are also greatly increased. The result is a huge increase in ASR as the amount of avatars in the system slightly increases.

A comparison of Tables 3 and 4 with Tables 1 and 2 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 1 and 2, while reaching saturation with only 425 avatars in Tables 3 and 4. However, there not exists any significant difference in ASR values between Tables 1 and Tables 3 in the first two lines, neither between Tables 2 and 4 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

No. of avatars	%CPU1	%CPU2	%CPU3	SR-S1	SR-S2	SR-S3	Average SR
375 (125,125,125)	72	74	71	221.23	233.57	225.71	226.84
400 (125,150,125)	71	81	73	219.36	285.21	218.77	241.11
425 (125,175,125)	73	100	70	228.74	1725.84	235.54	730.04
450 (125,200,125)	72	100	69	225.67	6308.87	218.99	2251.18
475 (125,225,125)	70	100	72	226.54	11554.86	221.42	4000.94

Table 3. Average system response and CPU utilization for each server when C_p^L is minimized

No. of avatars	%CPU1	%CPU2	%CPU3	SR-S1	SR-S2	SR-S3	Average SR
375 (125,125,125)	69	72	73	217.89	241.13	233.45	230.82
400 (125,150,125)	71	83	72	251.41	323.22	218.31	264.31
425 (125,175,125)	69	100	73	1290.21	2992.85	1432.80	1905.29
450 (125,200,125)	72	100	70	5678.51	8861.91	6007.82	6849.41
475 (125,225,125)	71	100	72	9998.83	16873.21	10254.31	12375.45

Table 4. Average system response and CPU utilization for each server when C_p^L is maximized

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 minimized, but the impact of C_p^L in the system throughput is higher.

5 Conclusions

One of the key issues in the design of scalable and efficient DVE systems is the partitioning problem. 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. Since results show an absence of correlation, 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 servers. When this limit is reached, then average system response greatly increases as new avatars are added to the system.

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 minimizing the amount of inter-server messages mainly has an effect on system latency. However, the amount of inter-server messages also has a significant effect on system throughput when workload balancing is optimized.

Therefore, we can conclude that in order to design actually scalable and efficient partitioning methods, the quality function used for solving the partitioning problem

must take into account the non-linear behavior of DVE systems with the number of avatars in the system.

As a future work to be done, we are currently working in the definition of a new quality function with the features described above. In a long term future, we plan to design partitioning methods that can provide quality of service. Effectively, the purpose of solving the partitioning problem may be different depending of the state of a DVE system. If the workload is high and the DVE system is close to saturation, then the purpose of the partitioning problem may be maximizing the DVE throughput, in order to support the highest number of clients as possible. However, if the DVE system has a low workload, then the purpose of the partitioning problem may be offering certain quality of service (for example, improving latency for those clients with the lowest communication bandwidth). In the latter case, the partitioning method should be adapted to that circumstances.

References

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. P. Barham, T.Paul, "Exploiting Reality with Multicast Groups", in *IEEE Computer Graphics & Applications*, pp.38-45, September 1995.
3. DIS. 1278.1 IEEE Standard for Distributed Interactive Simulation-Application Protocols (ANSI). DMSO. DoD High Level Architecture. 1997.
4. FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2000. Available at `http://www.fipa.org/specs/fipa00023/`
5. 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.
6. 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.
7. Michael Lewis and Jeffrey Jacboson, "Game Engines in Scientific Research", in *Communications of the ACM*, Vol 45. No.1, January 2002.
8. 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.
9. 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
10. D.C.Miller, J.A. Thorpe, "SIMNET: The advent of simulator networking", in *Proceedings of the IEEE*, 83(8), pp. 1114-1123. August, 1995.
11. 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.
12. Tohei Nitta, Kazuhiro Fujita, Sachio Cono, "An Application Of Distributed Virtual Environment To Foreign Language", in *IEEE Education Society*, October 2000.
13. 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.
14. S.Singhal, and M.Zyda, "Networked Virtual Environments", *ACM Press, New York, 1999.*
15. P.T.Tam, "Communication Cost Optimization and Analysis in Distributed Virtual Environment", *M. Phil second term paper, Technical report RM1026-TR98-0412*. Department of Computer Science & Engineering.The Chinese University of Hong Kong. 1998.