A Full Awareness Method for Peer-to-Peer Distributed Virtual Environments

P. Morillo, W. Moncho, J.M. Orduña , J. Duato

Resumen— **In recent years, large scale distributed virtual environments (DVEs) have become a major trend in distributed applications. Since architectures based on networked servers seems to be not scalable enough to support massively multiplayer applications, peer-to-peer (P2P) architectures have been proposed for these systems. However, the main challenge of P2P architectures is the awareness problem, consisting of providing each avatar with updated information about which other avatars are its neighbors. This paper presents a new awareness method based on unicast communication that is capable of providing awareness to 100% of avatars, regardless of both their location and their movement pattern in the virtual world. Therefore, it allows large scale DVEs based on P2P architectures to properly scale with the number of users.**

Palabras clave— **Distributed virtual environments, peer-to-peer architecture, awareness problem**

I. Introduction

LArge scale distributed virtual environments (DVEs) have become a major trend in distributed applications, mainly due to the enormous popularity of multiplayer online games in the entertainment industry. These highly interactive systems simulate a 3-D virtual world where multiple users share the same scenario. Each user is represented in the shared virtual environment by an entity called *avatar*, whose state is controlled by the user through a client computer. The system renders the images of the virtual world that each user would see if he was located at that point in the virtual environment. Hundreds and even thousands of client computers can be simultaneously connected to the DVE through different networks. Architectures based on networked servers have been during last years the major standard for DVE systems [1]. In these architectures, the control of the simulation relies on several interconnected servers. Client computers are assigned to one of the servers in the system. In these architectures, each new avatar represents an increase in the computational requirements of the application. Due to this increase, networked-server architectures do not properly scale with the number of existing users, particularly for the case of MMOGs [2].

The most adequate scheme in order to provide good scalability for large scale DVE systems seems to be P2P architectures, and several online games based on P2P architectures have been designed [3], [4]. Nevertheless, P2P architectures must face *awareness problem*. This problem consists of ensuring that each avatar is aware of all the avatars in its neighborhood [5]. Solving the awareness problem is a necessary condition to provide a consistent view of the environment to each participant. Effectively, if two neighbors avatars are not aware of such neighborhood, they will not exchange messages about their movements and/or changes, and therefore they will not have the same vision of the shared environment. Thus, providing awareness to all the avatars is a necessary condition to provide consistency (as defined in [6]). Awareness is crucial for MMOGs, since otherwise abnormal situations could happen. For example, a user provided with a non-coherent view of the virtual world could be shooting something that he can see although it is not actually there. Also, it could happen that an avatar not provided with a coherent view is killed by another avatar that it cannot see.

In DVE systems based on P2P architectures the neighborhood attribute must be determined in a distributed manner, in such a way that awareness is provided to all avatars during all the time. Currently, several strategies for providing awareness in DVE systems based on P2P architectures have been proposed [7], [8], [9], but either they are based on multicast communications or they do not provide full awareness. In this paper, we propose a method that provides full awareness to large scale DVE systems based on peer-to-peer architectures in an actually scalable way. Full scalability can be achieved if several kinds of neighbors follow a region-based strategy and they are allowed to increase and decrease as necessary. Performance evaluation results show that the proposed algorithm can provide full awareness in a large scale DVE system, even when avatars follow non-uniform movement patterns and they are unevenly distributed in the virtual world. Therefore, this algorithm can allow P2P architectures to become an actually efficient solution for appplications like MMOGs.

The rest of the paper is organized as follows: Section II analyzes the existing proposals for providing awareness in DVE systems based on P2P architectures. Section III describes the proposed algorithm and how it improves the weaknesses of the existing proposals. Next, Section IV presents the performance evaluation of the proposed method. Finally, Section V presents some concluding remarks and future work to be done.

This paper is supported by the Spanish MEC under Grant TIC2003-08154-C06-04

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

J. Duato is with DISCA, U. Politécnica de Valencia. e-mail: jduato@gap.upv.es

II. Background

The expansion of MMOGs has made large scale DVE systems to become usual, and networked-server architectures seem to lack scalability to properly manage the current number of avatars that these system can support (up to some hundred thousands of avatars [10]). As a result, some studies have proposed again the use of P2P architectures [7], [8], [9], since this schemes seems to be the most scalable ones. Nevertheless, before a P2P architecture can be used to efficiently support large scale DVE systems, the awareness problem must be still solved.

Some proposals use multicast communications to guarantee awareness [8]. Although multicast greatly improves scalability, it is hardly available on the Internet, which is the natural environment for multiplayer online games. Therefore, this scheme cannot be used in most of large-scale DVE systems. The solution proposed in [7] does not provide total awareness (it provides an awareness rate of 95%), but an awareness rate of 100% is crucial for MMOGs, since it guarantees that no faults like being killed by an "invisible" avatar will occur. The reason for this behavior is that it limits the number of neighbors that a given avatar can see, as described in [9]. Finally, the method proposed in [9] seems to provide full awareness. However, the use of Voronoi diagrams makes this method to require each avatar i to communicate with a high number of avatars. This feature suggests that this method is not scalable. In fact, this proposal lacks a performance evaluation. Therefore, the efficiency and scalability of this proposal cannot be stated.

III. A New Awareness Method: COVER

In order to provide full awareness to a DVE system based on a P2P architecture, we propose an approach based on a P2P hybrid organization where peer nodes (clients of the simulation) can play multiple roles in the DVE system. We have denoted this approach as *COVER*, since awareness for each avatar i is provided or "covered" by the avatars surrounding i.

In order to provide awareness to a given avatar i, COVER method involves all the avatars surrounding i up to the second level of neighborhood, like the method proposed in [9]. The first-level neighbors of an avatar i are those avatars in the DVE system in whose AOI avatar i appears. The second-level neighbors of i are all the neighbor avatars of the first-level neighbors of i. In order to forbid cyclic relationships (redundant messages), if a second-level neighbor is also a first-level neighbor, then it is not considered as a second-level neighbor. Each time an avatar i moves, it sends an updating message to each of its first-level neighbors. These neighbors in turn propagate the updating message of i to the second-level neighbors of i.

Unlike the methods proposed in the previous section [7], [9], COVER classifies avatars in two categories: covered or uncovered. Each avatar checks its classification each time it moves and each time that any of its neighboring avatars moves. This classification determines the behavior of P2P clients in order to get updated (conherent) information. We denote an avatar i as *covered* if its first or second-level neighbors are located in such a way that the intersections of their AOIs totally cover the AOI of i. Otherwise, it is considered as an *uncovered* avatar. Since the updating messages sent by any avatar arrives to its second-level neighbors, COVER method offers auto-awareness to covered avatars, because no avatar can approach them without being detected by their neighbor avatars. This mechanism for providing awareness to covered avatars is similar to the ones proposed in [7] and in [9].

However, it can happen that at a given moment a given avatar does not have enough neighbors around it to be provided with awareness (that avatar is uncovered). Unlike the method proposed in [7], our approach provides awareness also for uncovered avatars, by means of using *supernode* avatars. These avatars play multiple roles, acting not only as simple avatars but also as pseudo-servers [1]. Supernodes represent an upper layer in the awareness scheme, and they provide the required scalability while ensuring a awareness rate of 100%. Supernodes are responsible of providing awareness to the uncovered avatars in their surroundings, and they are initially designated by the entity in charge of the initialization of new avatars (denoted as Loader or Bootstrap server [7]) when they join the DVE system. At boot time, the loader divides the 3-D virtual scene into square sections called *regions*. For each region, the closest avatar to the geometric center of each region is selected by the loader as the supernode for that region. From that instant, supernodes are responsible of providing awareness to those uncovered avatars that are located within their regions. Uncovered avatars must send their updating messages not only to their neighbors, but also to the corresponding supernode of the region where they are located. In this way, supernodes can notice uncovered avatars when another uncovered avatar(s) cross their AOI. The auto-awareness of covered avatars ensures that before a covered avatar k can enter the AOI of an uncovered avatar i, another uncovered avatar j will cross the AOI of i. Therefore, this scheme does not require supernodes to notice uncovered avatars about the movement of covered avatars, significantly reducing the communications required for providing awareness. When any avatar either receives an updating message from its neighbors or when it moves, it checks again if it is a covered or an uncovered avatar.

As an example, figure 1 shows a 2-D region containing five avatars, represented as dots, and their respective AOIs, represented as circumferences around the dots. In this region avatars B, C, D and E are uncovered avatars. Since the circumference around avatar A is totally covered by the AOIs of avatars D, C and E, avatar A is classified as a covered avatar. Also, in this case avatar A has been chosen as supernode of the region (we have represented supernodes by depicting their AOI with a thicker circumference), and therefore this avatar will receive updating messages from all the uncovered avatars in this region (the rest of the avatars).

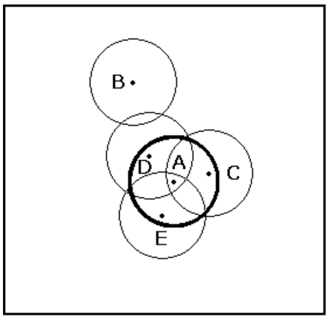

Fig. 1. An example of a covered avatar being the supernode at the same time

Although the client computers controlling the supernodes act as mirrored servers do in a server network architectures [1], they are client computers of a DVE system based on a P2P architecture. Therefore, they should not be significantly overloaded by the awareness mechanism. Since uncovered avatars add workload to their supernodes [11], COVER method uses a quad-tree segmentation of the virtual scene [12] to avoid the saturation of supernodes (thus providing scalability). Concretely, COVER limits the maximum number of uncovered avatars which are simultaneously connected to the same supernode. This parameter is called *MNUA*, (for Maximum Number or Uncovered Avatars). Whenever MNUA is exceeded, the supernode divides that region in four different subregions and computes a new supernode for each subregion, based on the criterion of geometric distance to the center of the subregion. Once the division has been performed and a new supernode is selected for each subregion, the uncovered avatars in each subregion are re-assigned to the new supernodes. When the system is running, this mechanism defines a dynamic quad-tree structure where each supernode has four sons. Two or more supernodes are *brothers* if they have been generated in the same division operation. Brother nodes are constantly monitoring each other to detect if the total number of uncovered avatars located in the four regions is under MNUA. In this case, a fusion operation is performed. In this operation the four brother subregions are joined to become a unique, larger region, and a new supernode for the new region is computed based on the same criterion of distance to the geometric center of the resulting region. Unlike the method described in [9], this mechanism provides the required flexibility for situations such as avatars heading to the same location of the virtual world, as shown below.

As an example, figure 2 shows the evolution of figure 1 when the proposed scheme is applied and the MNUA parameter has value of six avatars. Since three more avatars have joined the system and the MNUA value has been exceeded, the supernode has divided the region in four subregions. In this case, the resulting supernodes are now avatars A, D, E, and G. COVER method does not require that supernodes exchange the position of the avatars in their respective regions, like networked-server architectures do [1], [13]. This key issue allows COVER method to limit the amount of messages generated to provide awareness.

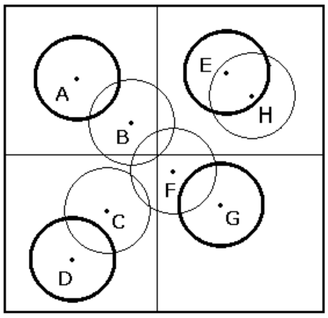

Fig. 2. Division of the virtual world shown in figure 1 into four subregions

In order to offer full awareness to those avatars located at the borders of different regions (denoted as *critical* avatars), *secondary* supernodes are used. Critical avatars are defined as those uncovered avatars whose AOIs intersect with more than a single region. Critical avatars should send updating messages not only to the supernode managing the region where they are located, but also to the supernodes managing the adjacent regions. These supernodes show uncovered avatars located in different regions, and they are considered as secondary supernodes for critical avatars. Figure 2 shows that avatar B and F must be considered as critical avatars, because the area of their AOI exceed the limits of the subregion where they are located. The solution for this situation is to force B to send the updating messages not only to supernode A, but also to supernodes D and E, as discussed below. In the same way, avatar F must send updating messages to the four supernodes.

IV. Performance Evaluation

We propose the evaluation of generic DVE systems by simulation. The evaluation methodology used is based on the main standards for modeling collaborative virtual environments, such as FIPA [14], DIS [15] and HLA [16]. Concretely, (as we did for the case of DVE systems based on networked-server architectures [11]) we have developed a standalone simulation tool, denoted as SimPeerDve (SPD), that models the behavior of a generic DVE system based on a P2P architecture as a set of independent avatars. These avatars are located within a seamless 3D virtual world [2] following three different and well-known initial distributions: uniform, skewed and clustered [17], [11]. Starting from these initial locations, in each simulation avatars perform 100 iterations. Each iteration consists of each avatar independently moving into the scene. Iterations are performed at the typical rate of 1 avatar movement every 2 seconds). We have considered three typical movement patterns in DVE systems [11].

SimPeerDve not only simulates the behavior of the avatar population in the virtual scene, but it also monitors the number and type of messages that each avatar sends or receives from the rest of elements in the DVE system. For comparison purposes, SimPeerDve simulates the execution of different awareness methods for all the avatars in the system each time an iteration has finished.

Using this simulator, we have performed experimental studies to evaluate the performance of the proposed technique. For comparison purposes, we have simulated the awareness method proposed in the previous section and also the awareness method proposed in [7], since this method currently provides the best awareness results for DVE systems based on P2P architectures (as stated above, the method proposed in [9] has not been evaluated). In order to ensure that the evaluation is performed under the worst case, SimPeerDve allows the overlapping of different avatars at the same location of the virtual environment. Although this situation would be erroneous in a real environment, it allows us to increase the number of avatars located in a given region of the virtual world beyond the limits of a real environment. If awareness is provided under such circumstances, then awareness is guaranteed under real conditions.

Figure 3 shows the evaluation results for the awareness method proposed in this paper (labeled as COVER) as well as for the awareness method described in [7] (labeled as K(x-x)), under all the possible combinations of initial distributions and movement patterns of avatars. This figure shows a representative example of the experiments performed with different DVE configurations. Concretely, it shows the results obtained for a DVE configuration of 1000 avatars. On the X-axis this figure shows the iteration number of the simulation performed, and on the Y-axis it shows the percentage of awareness. Each point in the plots represents the average awareness percentage during the last ten iterations. Since we have obtained identical plots for COVER method (a flat line at 100%, that is, full awareness during the whole simulation) for all the combinations of initial distributions and movement patterns of avatars, we have plotted a single plot for this method .

Figure 3 shows that for a large scale DVE configuration the method described in [7] only is able to provide a high percentage of awareness (around a 85%) under the uniform movement pattern. For the rest of combinations of initial distributions and movement patterns of avatars, the method proposed in [7] provides a awareness rate below 50%. The worst results of this method are provided for the combination of a clustered initial distribution of avatars and HPA movement pattern, being lower than 10% at the end of the simulation. The reason for this behavior is that for non uniform movement patterns, avatars are unevenly distributed in the virtual world most of the simulation time. Under this situation, the probability that a given avatar i has one or more unknown neighbors crossing its AOI increases, since its known neighbors of i also tend to be unevenly distributed around the AOI of i. The use of supernodes avoids inconsistencies under such situations when using COVER method.

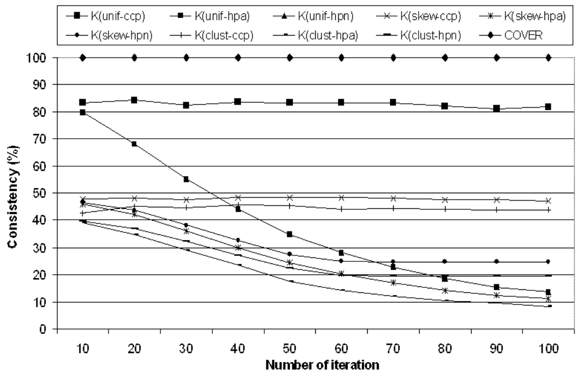

Fig. 3. Percentage of awareness provided for all combinations of initial distributions of avatars and movement patterns.

In order to show that COVER method provides total awareness without increasing the number of messages exchanged by avatars, figure 4 shows the number of messages sent during the simulations whose awareness rates are summarized in figure 3. The plots in this figure show the number of messages exchanged when using COVER method for the different combinations of initial distributions and movement patterns of avatars. Since the method proposed in [7] limits the number of neighbors that a given avatar can communicate with, the number of messages sent when using this method is not comparable with the number of messages sent when using COVER method. All the simulations whose results are shown in this figure have been performed in a DVE system composed of 1000 avatars. Each point in the plots represents the average value of messages sent by all the avatars in the DVE system during the last 10 iterations.

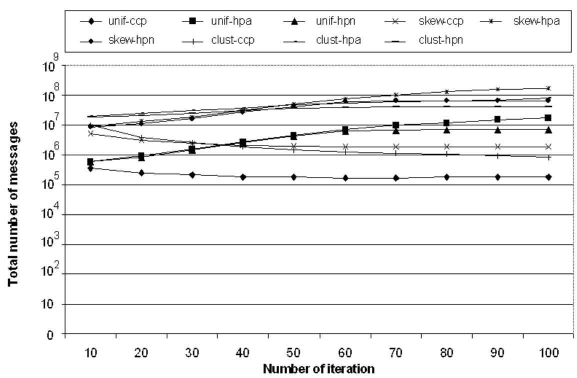

Fig. 4. Total number of messages exchanged by avatars during the simulations when using COVER method.

Figure 4 shows that there is a difference of two orders of magnitude between the two plots showing the largest (skewed-HPA) and the smallest (unif-CCP) number of messages exchanged among avatars. This difference is due to the different groupings of avatars that each combination of initial distributions and movement patterns produces. However, none of the plots shows a significant slope, meaning that COVER method does not generate more messages as more iterations are performed. On the contrary, it provides awareness while requiring low variations in the number of messages exchanged by avatars along the simulation, regardless of the combination of initial distribution and movement patterns of avatars. Although we have not made simulations with more than 1000 avatars due to the limits of our infrastructures, we have made simulations with 64, 128 and 256 avatars. Although the corresponding figures are not shown here for the sake of shortness, they contain plots with similar slopes, although they have different values. These results suggest that the more avatars in the system, the more messages are exchanged during the simulation. However, no significant differences appear as the simulation proceeds, regardless of the number of avatars in the simulation.

In order to show the scalability of COVER method, we have also studied the number of messages sent and received by supernodes under different movement patterns and for simulations performed with different numbers of avatars. For the sake of shortness, we present here the results for those combinations of initial distributions and movement patterns of avatars that show the largest (skewed-HPA) and the smallest (unif-CCP) number of messages exchanged among avatars in figure 4. Concretely, figure 5 shows the average number of messages handled (sent or received) by each supernode in the system during the simulations performed under the combination of a uniform initial distribution of avatars and CCP movement pattern. Each point in these plots is computed as follows: after each iteration in a simulation, the number of supernodes in the system as well as the number of messages sent and received by supernodes are counted, and the *average number of messages per supernode* (ANMS) is computed. Since the number of supernodes dynamically varies, when the simulation finishes, the average value of the 100 ANMS values (each simulation is composed of 100 iterations) is computed. This is the value represented in each point in the plots.

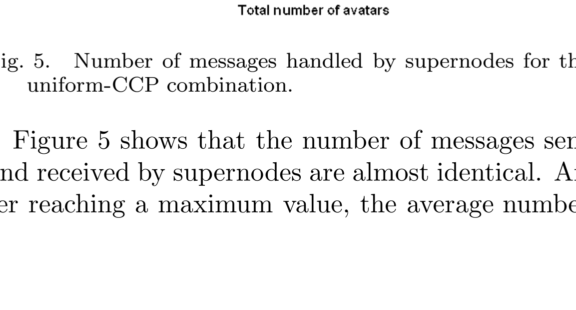

Fig. 5. Number of messages handled by supernodes for the uniform-CCP combination.

Figure 5 shows that the number of messages sent and received by supernodes are almost identical. After reaching a maximum value, the average number of messages handled (sent or received) by supernodes remains constant or even decrease as the system grows. Effectively, the number of supernodes is proportional to the number of avatars, keeping constant the average number of messages handled by each supernode. These results show that the proposed method properly balances the load generated for guaranteeing awareness.

Figure 6 shows the average number of messages handled by each supernode in the system during the simulations performed under the skewed-HPA combination, the one requiring the largest number of messages exchanged among avatars. The shape of the two plots in this figure is similar to those in figure 5, showing a flat slope. The average number of messages handled by each supernode remains within a range of values between 13 and 19, showing that the proposed method scales well with the number of avatars in the system. Moreover, when comparing this figure with figure 5, we can see that this range of values is approximately the same for both figures. That is, the behavior of the proposed method does not depend on the movement pattern nor on the initial distribution of avatars in the virtual world. This feature is a key issue to provide an actually scalable awareness method, but it has not been shown in any of the currently proposed methods.

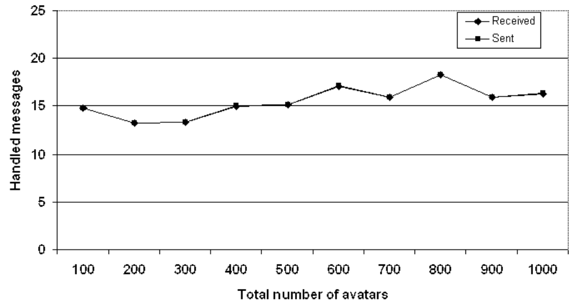

Fig. 6. Number of messages handled by supernodes for the skewed-HPA combination.

Finally, in order to show that in this experiment avatars are not put to unrealistic conditions, figure 7 shows the average and maximum densities of avatars in the AOI of each avatars as the iterations of the simulations are performed. Concretely, this figure shows the density of avatars for the case of initially skewed distribution of avatars and for the HPA movement patterns, in a DVE configuration of 1000 avatars. This combination of movement pattern and initial distribution of avatars is the one that provides the highest workload to the DVE system [18], [19].

Figure 7 shows that both average and maximum values starts with relatively low density values. As iterations are performed, these values linearly increase until a limit of 300 neighbors avatars is reached in the case of the average values (a value of 500 avatars in the case of maximum values). An average message rate of 300 messages every two seconds (and a maximum rate of 500 messages every two seconds) is easily supported by any client computer, thus working below the saturation point. Moreo-

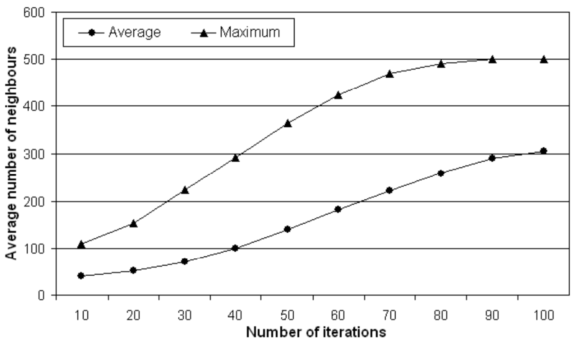

Fig. 7. Average and maximum density of avatars in the AOI of avatars.

ver, if we compare these results with the ones obtained from a similar DVE DVE system based on a networked-server architecture[19], we can see that for the Skewed+HPA combination the networked-server DVE could only perform 10 iterations before collapsing due to the workload generated by 600 avatars. On the contrary, the DVE system based on a P2P architecture does not reach the saturation point when supporting 1000 avatars. Therefore, the proposed method properly scales with the number of avatars.

V. Conclusions and Future Work

In this paper, we have proposed a new awareness method for large-scale DVE systems based on P2P architectures, denoted as COVER. Unlike the currently proposed methods, COVER uses two kind of neighbors to provide awareness, normal neighbors and supernodes. While the former ones provide awareness to avatars in the same way other methods do, the latter ones allow to provide awareness to those avatars that are isolated or not completely surrounded by other avatars. This key feature allow to provide full awareness (awareness to 100% of avatars), regardless of the distribution of avatars in the virtual world.

Performance evaluation results show that the proposed method is able to provide full awareness to large scale DVE systems composed of up to 1000 avatars, regardless of both the movement pattern and the initial distribution of avatars in the virtual world. This results have not been shown by any of the currently proposed methods.

Due to the quad-tree segmentation algorithm used to select new supernodes, neither the movement pattern of avatars nor the initial distribution of avatars have a significant effect on the number of messages sent by avatars as simulations proceed. Evaluation results also show that this number remains with a flat slope, even for those movement patterns that in the last iterations tend to group avatars in certain points of the virtual world. This result indicates that the proposed method properly balances the workload generated to provide awareness to all avatars.

Also, performance evaluation results show that the number of messages handled by supernodes does not increase as new avatars are added to the DVE system. This scalability is achieved by selecting new supernodes when MNUA parameter is exceeded and merging several supernodes into a single supernode when adjacent supernodes manage less than MNUA avatars.

As a result, we can conclude that COVER method provides full awareness to large-scale DVE systems based on P2P architectures in a scalable and efficient way.

Referencias

[1] S. Singhal and M. Zyda, *Networked Virtual Environments*, ACM Press, 1999.

[2] T. Alexander, *Massively Multiplayer Game Development II*, Charles River Media, 2005.

[3] S. Mooney and B. Games, *Battlezone: Official Strategy Guide*, BradyGame Publisher, 1998.

[4] L. Gautier and C. Diot, "Design and evaluation of mimaze, a multi-player game on the internet," in *Proceedings of IEEE Multimedia Systems Conference*, 1998.

[5] Randall B. Smith, Ronald Hixon, and Bernard Horan, *Collaborative Virtual Environments*, chapter Supporting Flexible Roles in a Shared Space, Springer-Verlag, 2001.

[6] S. Zhou, W. Cai, B. Lee, and S. J. Turner, "Time-space consistency in large-scale distributed virtual environments," *ACM Transactions on Modeling and Computer Simulation*, vol. 14, no. 1, pp. 31–47, 2004.

[7] Y. Kawahara, T. Aoyama, and H. Morikawa, "A peer-to-peer message exchange scheme for large scale networked virtual environments," *Telecommunication Systems*, vol. 25, no. 3, pp. 353–370, 2004.

[8] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, "Peer-to-peer support for massively multiplayer games," in *Proceedings of IEEE InfoCom'04*, 2004.

[9] S. Y. Hu and G. M. Liao, "Scalable peer-to-peer networked virtual environment," in *Proceedings ACM SIGCOMM 2004 workshops on NetGames '04*, 2004, pp. 129–133.

[10] ," Everquest: http://everquest.station.sony.com/.

[11] P. Morillo, J. M. Orduña, M. Fernández, and J. Duato, "Improving the performance of distributed virtual environment systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 7, pp. 637–649, 2005.

[12] J. Smith and S. Chang, "Quad-tree segmentation for texture-based image query," in *Proceedings of the second ACM International Conference on Multimedia*, 1994, pp. 279–286.

[13] N. Beatrice, S. Antonio, L. Rynson, and L. Frederick, "A multiserver architecture for distributed virtual walkthrough," in *Proceedings of ACM VRST'02*, 2002, pp. 163–170.

[14] FIPA, "Fipa agent management specification," 2000, Available at http://www.fipa.org/specs/fipa00023/.

[15] IEEE, *1278.1 IEEE Standard for Distributed Interactive Simulation-Application Protocols (ANSI)*, 1997.

[16] F. Kuhl, R. Weatherly, and J. Dahmann, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, Prentice-Hall PTR, 1999.

[17] John C.S. Lui and M.F. Chan, "An efficient partitioning algorithm for distributed virtual environment systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, 2002.

[18] P. Morillo, J. M. Orduña, M. Fernández, and J. Duato, "An adaptive load balancing technique for distributed virtual environment systems," in *Proc. of Intl. Conf. on Parallel and Distributed Computing and Systems (PDCS'03)*. IASTED, 2003, pp. 256–261, ACTA Press.

[19] P. Morillo, J. M. Orduña, M. Fernández, and J. Duato, "A fine-grain method for solving the partitioning problem in distributed virtual environment systems," in *Proc. of Intl. Conf. on Parallel and Distributed Computing and Systems (PDCS'04)*. IASTED, 2004, pp. 292–297, ACTA Press, Best paper award in the area of load balancing.