Simulation Modelling Practice and Theory 25 (2012) 1–16

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

A simulation modelling tool for Distributed Virtual Environments

Christos Bouras [a,b,*], Eri Giannaka [a,c], Thrasyvoulos Tsiatsos [d]

[a] Computer Engineering and Informatics Department, University of Patras, GR-26500 Rion, Patras, Greece
[b] Research Academic Computer Technology Institute, N. Kazantzaki Str., Patras University, GR-26500 Rion, Patras, Greece
[c] Athens Information Technology, 0.8 km Markopoulou Ave., 19002 Peania, Attika, Greece
[d] Department of Informatics, Aristotle University of Thessaloniki, PO Box 114, GR-54124 Thessaloniki, Greece

A R T I C L E I N F O

Article history:
Received 2 November 2010
Received in revised form 16 December 2011
Accepted 16 February 2012
Available online 14 March 2012

Keywords:
Distributed Virtual Environments
Performance monitoring
Simulation modelling
Simulator

A B S T R A C T

Distributed Virtual Environments simulate the behaviour and activities of a great number of users interacting in a virtual world over a wide area network. The size of the virtual worlds and the tremendous number of users that these environments are called to support require additional bandwidth and computational resources. For supporting large-scale Distributed Virtual Environments, extended infrastructure is needed in terms of both hardware and software. However, both researchers and application designers do not always have access to such an extended infrastructure and the assessment and evaluation of developed performance improvement techniques becomes extremely difficult. To address this issue, this paper presents a simulation modelling tool, called STEADiVE for networked servers Distributed Virtual Environments that could be used by designers for evaluating the performance of their approaches under different scenarios and system settings. The validation of the simulation modelling tool has showed that it achieves high accuracy in representing a real DVE system. STEADiVE comes to fill in the gap in the area of simulation tools for these systems.

1. Introduction

Distributed Virtual Environments (DVEs) have become a major trend in distributed applications [6]. DVEs simulate real or imaginary worlds by incorporating rich media and graphics. A large number of platforms and applications were designed and developed to support large-scale DVEs, which were gradually adopted in a wide range of both academic and industrial environments. Their most common application is still met in the entertainment area, where platforms such as Second Life [10] and World of Warcraft [11] have become extremely popular. To handle these demanding applications, existing approaches fall usually into one of the following architectures: (a) networked servers architectures and (b) peer-to-peer architectures. A lot of work has been performed towards both these directions, including algorithms for the partitioning problem [1,7], load balancing techniques [4], awareness methods [6] and other optimization techniques [5].

The requirements and characteristics of each DVE system may vary significantly among virtual worlds for different simulated scenarios. For example, in case of an educational DVE, the consistency of the world would not be significantly affected if a number of position messages (i.e., messages sent each time a user changes his/her position) were lost. However, if position messages were lost while in a virtual battlefield, where soldiers move and run, then the sense of realism, users' awareness and performance would be significantly impacted. Therefore, the decision on the techniques and approaches used is

* Corresponding author at: Research Academic Computer Technology Institute, N. Kazantzaki Str., Patras University, GR-26500 Rion, Patras, Greece. Tel.: +30 2610 960375/996951; fax: +30 2610 969016.
E-mail address: bouras@cti.gr (C. Bouras).

1569-190X/$ - see front matter
doi:10.1016/j.simpat.2012.02.006

usually related to the objectives, scope and context that each virtual world aims at supporting along with its special characteristics.

However, running and measuring the effectiveness of a large-scale DVE system is a difficult problem. On the one hand, one cannot possibly find thousands of real users to participate. On the other hand, an extended infrastructure is needed in terms of both hardware and software to support large-scale DVEs and both researchers and application designers do not always have access to such an extended infrastructure. To address this issue, this paper presents a simulation modelling tool for networked servers DVEs that comes to fill in a gap in the area of simulation tools for this kind of system. The main goal of the simulation model presented in this paper is to provide a simulation framework that can be used by application designers in order to select and evaluate the appropriate scheme, algorithm and technique for the virtual world scenario they are called to simulate, without the need for extensive software and hardware infrastructures. The simulation model allows for full customization for both generic and scenario-specific DVE systems. The formulation of various DVE systems in combination with a number of existing performance improvement algorithms and techniques that the model incorporates can provide insight and valuable information for the behaviour and performance of the DVE under different approaches and system settings. More specifically, the simulation model has been designed in order to include most of the entities, processes, parameters and techniques used in the majority of both generic and scenario-specific DVE systems. The design and implementation of the simulation model derives from a definition for the issue of "optimized" performance for DVE systems as an operational management problem. Therefore, one of the innovative aspects of the simulation model is that it is based on "translating" the DVE system's requirements to the concepts of operational management. Thus, the design and implementation of the framework is conducted with SIMUL8 [12], an integrated environment for working with simulation models, widely used in operational management. The simulation model has been validated through a comparison with the results produced by a real DVE system. The results of the validation showed that the simulation model achieves high level of accuracy in representing and modelling the behaviour and performance of a real DVE system.

The rest of the paper is structured as follows: Section 2 presents the motivation for the work presented as well as related work in the area of simulation tools for evaluating DVEs' performance, while Section 3 describes the main concepts, entities, processes and characteristics of generic DVEs along with the performance issues that these systems need to address. Section 4 presents the simulation framework in terms of its basic entities and their "mapping" to SIMUL8 objects. Section 5 presents the STEADiVE simulation model in terms of the algorithms and techniques integrated for simulating DVEs' behaviour along with the main results produced by the model. Section 6 presents the model validation for proving the model's accuracy and validity and Section 7 presents an indicative experiment conducted in order to demonstrate the way that the simulation tool can be used for assessing different techniques. Finally, Section 8 concludes the paper and Section 9 presents some planned next steps.

2. Motivation and related work

One of the basic problems when designing and implementing algorithms, methods and techniques for large-scale DVEs is the way that their efficiency could be examined. In most cases, the evaluation is based on theoretical models that often fail to meet the conditions met in real DVE systems. More specifically, an extended infrastructure is needed in terms of both hardware and software to support large-scale DVEs. As both researchers and application designers do not always have access to such an extended infrastructure, the assessment and evaluation of developed techniques becomes extremely difficult.

The problems and challenges that a DVE system needs to address as well as the algorithms and techniques for performance improvement are strongly related to the architectural solution that the system adopts. As mentioned above, the architectural approaches for large-scale DVEs fall usually into one of the following architectures: (a) networked servers architectures and (b) peer-to-peer architectures. Regarding peer-to-peer (P2P) DVE systems, there is a range of simulators available that simulate the behaviour of these systems and allow for performance monitoring of different techniques. In [16] the authors present a simulator for P2P DVE systems that reproduces the behaviour of a P2P DVE system and is capable of performing the evaluation, parameterization and result acquisition of DVEs. The simulator is composed of two different kind of applications, written in C++ that implement the clients and the central manager to whom the clients should connect to in order to join the system. Both type of applications use different threads for managing the connections that should be established, while the communications are implemented by means of sockets. In the direction of P2P DVE systems Adam [17] is another simulator developed for monitoring and assessing the performance and behaviour of this type of systems. Adam is built on top of the discrete event simulator OMNet++ [18] in conjunction with the MRIP engine Akaroa [19] and comprises a large set of measuring tools to compare the effect of different algorithms and to allow for optimization of these algorithms. Finally, VAST [20] is a light-weight network library for simulating P2P DVE systems and is also designed to be an experimental middleware consisting of three major components: network-layer, protocol and simulator. This allows new algorithms or simulators to be tried and tested for assessing their performance.

However, for the cases of networked servers DVE systems the simulation tools for representing the DVE and for monitoring and assessing its performance are rather limited. In most cases, both application designers and researchers adopt specialized methods [4] for evaluating different techniques. More specifically, most of these simulations aim at finding the effect of a specific algorithmic change or of an algorithmic parameter in an otherwise monolithic application and are, in their vast majority, completely non-transferable to other scenarios or other algorithms. A more unified approach for a simulation