A Framework for Execution and Visualization of Situated Agents Based Virtual Environments

Giuseppe Vizzari[a], Giorgio Pizzi[a], Flávio Soares Corrêa da Silva[b]

Abstract—**This document briefly describes a framework supporting the definition and implementation of virtual environment inhabited by interacting situated agents defined according to the Multilayered Multi-Agent Situated System model. The framework supports the specification and execution of visually rich 3D virtual environment endowed by the presence of mobile agents acting and interacting inside it according to a multi-agent model.**

I. Introduction

THE design and realization of virtual environments inhabited by social entities is a significant application of the conjoint results of various research areas in computer science and engineering. Virtual environments have been exploited in several ways, and in particular:

- to support computer mediated forms of human interaction, characterized by the introduction of Embodied Conversational Agents facilitating users' interactions [9] or supplying awareness information in a visually effective form [13];
- to realize operational laboratories for participatory design, supporting the effective visualization of various alternative design choices to the involved stakeholders [6][9][8];
- to provide effective instruments for the modeling, simulation and visualization of the dynamics of entities situated in a representation of an existing, planned or reconstructed environment or situation [7][12];
- for sake of entertainment, in movies, computer games or in online communities (see, e.g., Second Life[1]).

While all these applications are characterized by a strong requirement for realistic and effective visualization tools (and some of them require a thorough analysis of the system usability, due to the necessary accessibility by non-technically skilled users), they also call for expressive models supporting the specification of behaviours for the entities that inhabit these environments, as well as the interaction among them and with the environment itself. The fact that the overall performance of the system is essentially dependant on the single actions and interactions that are carried out by entities inhabiting the modeled environment leads to consider that the Multi-Agent Systems approach is particularly suited to tackle the modeling issues that are posed by this scenario. This idea is also corroborated by the fact that most of the above introduced references actually describe systems based on this approach, and by specific experiences in applying MAS approaches to specific virtual environments applications such as computer games [10].

In this vein, the main aim of this document is to show the current advancement of a long term project that provides the realization of a framework supporting the development of MAS based simulations based on the Multilayered Multi-Agent Situated System model [1] provided with an effective form of 3D visualization. The main goal of the framework is to support a smooth transition from the definition of an MMASS based model of given situation to the realization of simulation systems characterized by an effective 3D user interface. One of the possible application areas of this kind of system is related to the modeling and simulation of crowds of pedestrians to support architectural design or urban planning [2][3]. In order to have information flowing appropriately from the formal model to design professionals (e.g. architects and urban planners), the MMASS-based simulator must be supported by adequate visualization and animation tools. Such supporting tools are the core issue of the present document. For sake of space, details of the MMASS are omitted, as well as a discussion of the related works; a more through discussion of these topics can be found in an extended version of this document in this volume [15].

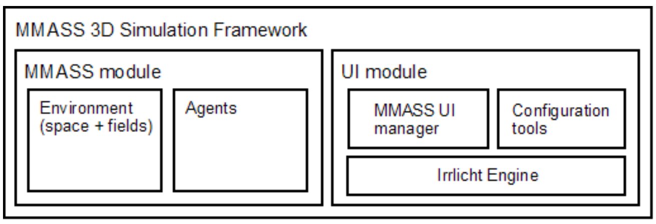

Figure 1 – Overall architecture of the framework for MMASS based virtual environments.

II. The Execution and Visualization Framework

The basic approach that was adopted for this project is to integrate an existing MAS modeling and development framework with an infrastructure supporting an effective form of 3D visualization of the dynamics generated by the model. In particular, to realize the second component we

[a]Complex Systems and Artificial Intelligence research center, University of Milano-Bicocca, via Bicocca degli Arcimboldi 8, 20126 Milano, {giuseppe.vizzari, giorgio.pizzi}@csai.disco.unimib.it

[b]Department of Computer Science, Institute of Mathematics and Statistics, Universidade de São Paulo, fcs@ime.usp.br

[1] http://secondlife.com

adopted Irrlicht[2], an open-source 3D engine and usable in C++ language. It is cross-platform, it can exploit OpenGL or DirectX libraries for 3D visualization, and it provides a performance level that we considered suitable for our requirements. It provides a high level API that was adopted for several projects related to 3D and 2D applications like games or scientific visualizations. The MAS modeling and development framework we adopted is a C++ porting and relevant refactoring of the original MMASS framework [2], aimed at adapting it to the different programming language and also at optimizing some mechanisms such as commonly adopted field diffusion algorithms. The overall framework was developed and tested in the Windows XP operating system, but it can be easily ported to MacOS X or Linux.

The overall architecture of the framework is shown in Figure 1. The following subsections will discuss the basic elements of this C++ version of the MMASS framework (MMASS module in the figure) and the infrastructure interfacing this module with the 3D visualization engine (MMASS UI manager in the figure).

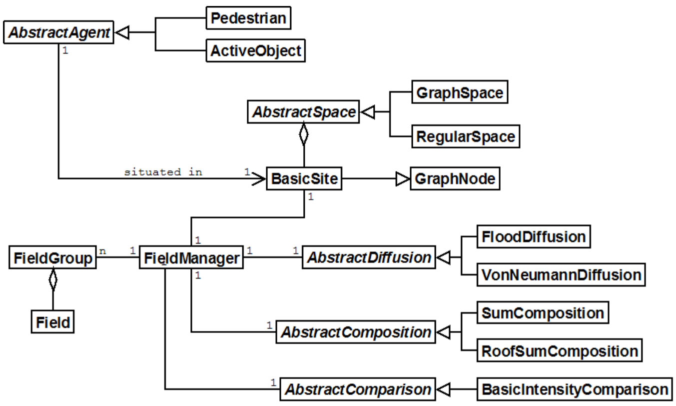

Figure 2 – Simplified class diagram of the part of the framework devoted to the realization of MMASS concepts and mechanisms.

A. Supporting and Executing MMASS Models

The MMASS framework adopted for this project is essentially a library developed in C++ providing proper classes to realize notions and mechanisms related to the SCA and MMASS models. In particular, a simplified class diagram of the MMASS framework is shown in Figure 2. The lower part of the diagram is devoted to the environment, and it is built around the BasicSite class. The latter is essentially a graph node (i.e. it inherits from the GraphNode class) that is characterized by the association with a FieldManager. The latter provides the services devoted to field management (diffusion, composition and comparison, defined as abstract classes). An abstract space is essentially an aggregation of sites, whose concretizations define proper adjacency geometries (e.g. regular spaces characterized by a Von Neumann adjacency or possibly irregular graphs).

An abstract agent is necessarily situated in exactly one site. Concrete agents defined for this specific framework are active objects (that are used to define concrete points of interest/reference to be adopted in a virtual environment) and pedestrians (that are basic agents capable of moving in the environment). Actual pedestrians and mobile agents that a developer wants to include to the virtual environment must be defined as subclasses of Pedestrian, overriding the basic behavioural methods and specifically the *action* method.

B. Integrating the Models with a Realtime 3D Engine

While the previous elements of the framework are devoted to the management of the behaviours of autonomous entities and of the environment in which they are situated, another relevant part of the described framework is devoted to the visualization of these dynamics. More than entering in the details of how the visualization library was employed in this specific context, we will now focus on how the visualization modules were integrated with the previously introduced MMASS framework in order to obtain indications on the scene that must be effectively visualized.

Figure 3 shows a simplified class diagram of the main elements of the 3D Engine Library. The diagram also includes the main classes that are effectively in charge of inspecting the state of the MMASS environment and agents, and of providing the relevant information to the SceneManager that will translate it into a scene to be visualized. The *Project* class act as a container of the 3D models providing the graphical representation of the virtual environment (*Model3D* objects), as well as the graph related to the adopted discretization of this physical space (a *Graph* object visually representing the previously discussed *physical layer*). It also includes a set of *Avatar* objects, that are three dimensional representations of *Pedestrian* objects (introduced in the previous subsection).

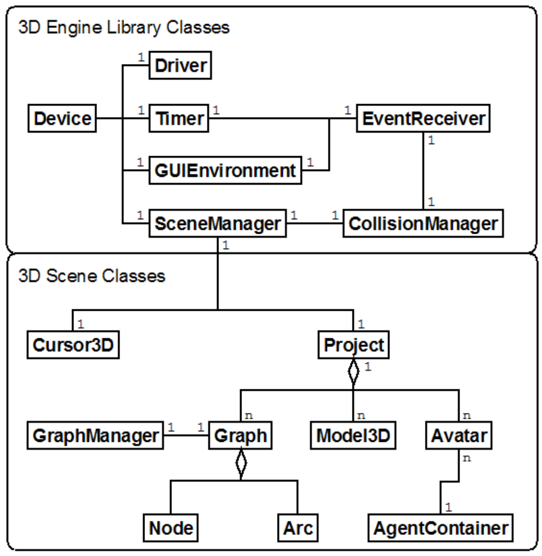

Figure 3 – Simplified class diagram of the part of the framework devoted to the management of the visualization of the dynamics generated by the model.

The framework must be able to manage in a coordinated way the execution of the model defined for the specific virtual environment and the updating of its visualization. To manage this coordinated execution of different modules and procedures three main operative modes have been defined and are supported by the framework. The first two are characterized by the fact that agents are not provided with a

[2] http://irrlicht.sourceforge.net/

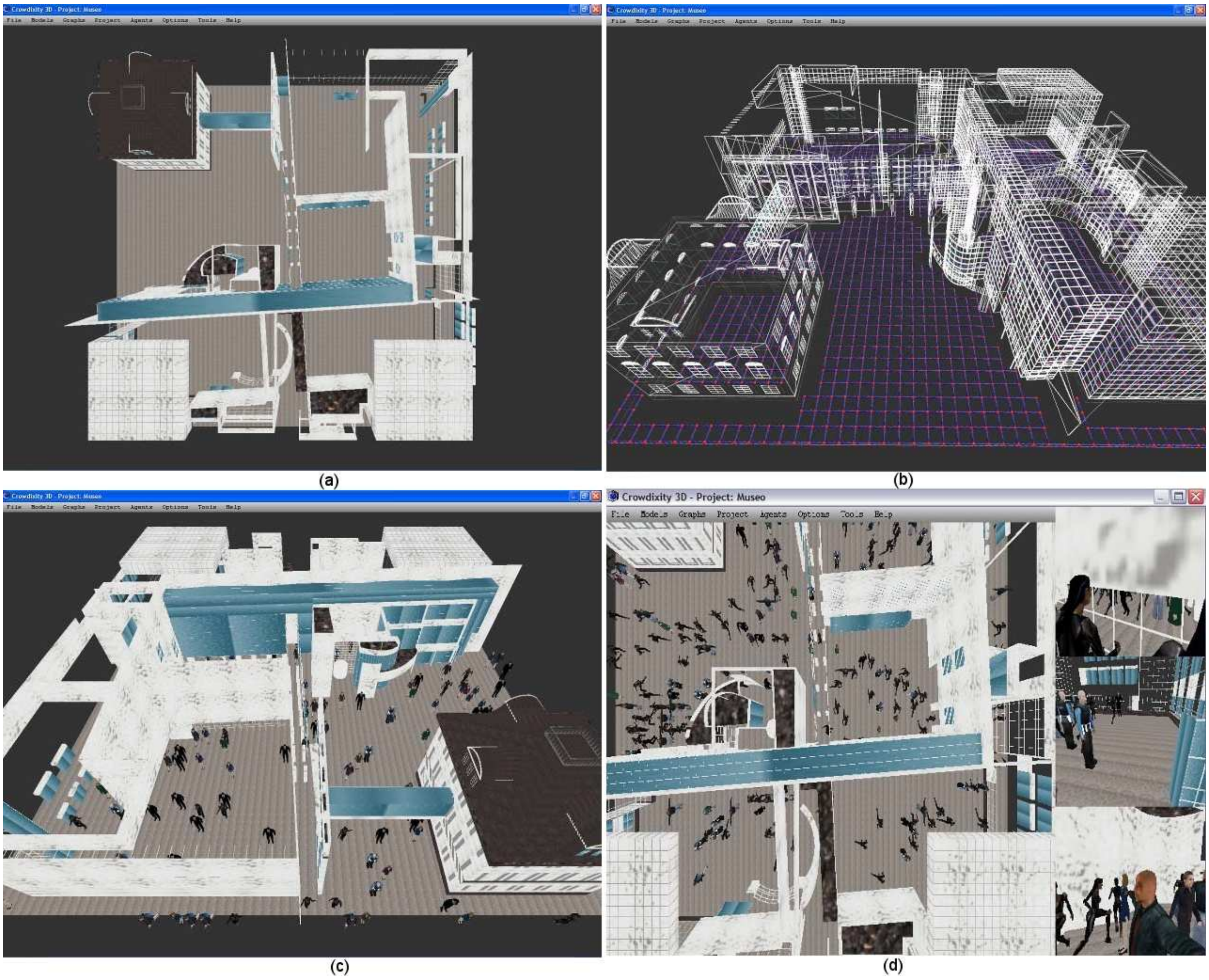

Figure 4 – Four screenshots of the virtual museum application, showing the structure of the environment - (a) and (b) – a perspective view of the evacuation and also a 'bird's eye' view of the environment coupled with three 'first-person' perspectives of agents - (c) and (d).

thread of control of their own. A notion of *turn* is defined and agents are activated to execute one action per turn, in a *sequential* way or in a conceptually *parallel* way (as for a Cellular Automaton). In this case, respectively after each agent action or after a whole turn the scene manager can update the visualization. On the other hand, agents might be *associated with a thread of control* of their own and no particular fairness policy is enforced. The environment, and more precisely the sites of the MMASS space, is in charge of managing possible conflicts on the shared resource. However, in order to support a fluid visualization of the dynamics generated by the execution of the MAS, the *Pedestrian* object before executing an action must coordinate with the related *Avatar*: if the previous movement was still not visualized, the action is temporarily blocked until the visualization engine has updated the scene. It must be noted that in all the introduced activation modes the environment is in charge of a regulation function [5] limiting agents' autonomy for sake of managing the consistency of the overall model or to manage a proper form of visualization.

III. SAMPLE APPLICATIONS

The aim of this section is to present a sample application to show how the framework supports the definition of MMASS models and the realization of an effective three dimensional visualization. The application was also chosen to show the potential of the framework in terms of execution of a large number of agents. Tests were carried out on a notebook on which the Windows XP Professional operating system was installed; the notebook was provided with an Intel Pentium IV 2.4 GHz processor, with 320 MB RAM and an ATI Raedon IGP graphic card with 128 MB (shared system memory).

The sample application is about the movement of agents inside a virtual museum; the aim of the agents in this scenario is to move outside the buildings to gather in specific areas, as in case evacuation. In this case the environment comprises around 2000 sites (a gross discretization of the represented environment) with around 6000 arcs connecting them; 500 agents were randomly positioned inside buildings, and they were provided with a thread of control of their own. Both the environment and agents were characterized by a 3D visual model, with textures; some relevant screenshots of this sample application are shown in Figure 4. Once again, the analytical results of this simulation are not relevant, since the agent models were extremely simple and they were not calibrated against real data. The simulation was executed and visualized with a number of FPS constantly above 30.

We also executed a stress test on a different hardware configuration, to verify the scalability of the framework; the workstation was based on Windows XP Professional operating system, with an Intel Pentium Core 2 Duo 2.4 GHz, 2 GB RAM and a NVIDIA Quadro FX 3450 graphic card with 256 MB. The test environment was constituted by 11000 sites, connected by around 44000 arcs; 10000 agents, sequentially activated, were positioned in this environment. Their behaviour was simply to move towards the closest source of an 'exit' field; agents reaching the source were removed from the environment. The system was able to execute and visualize the simulation with 22 FPS, when the structure of the environment was hidden (reducing the number of displayed triangles), and with 3 FPS when it was visualized.

REFERENCES

[1] S. Bandini, S. Manzoni, C. Simone. Heterogeneous Agents Situated in Heterogeneous Spaces. Applied Artificial Intelligence, 16(9-10):831–852, 2002.

[2] S. Bandini, S. Manzoni, G. Vizzari. Situated Cellular Agents: a Model to Simulate Crowding Dynamics. IEICE - Transactions on Information and Systems: Special Section on Cellular Automata, Vol.E87-D(3):669-676, 2004.

[3] S. Bandini, S. Manzoni, G. Vizzari. Multi Agent Approach to Localization Problems: the Case of Multilayered Multi Agent Situated System. Web Intelligence and Agent Systems, IOS Press, 2(3):155-166, 2004.

[4] S. Bandini, S. Manzoni, G. Vizzari. Towards a platform for Multilayered Multi Agent Situated System based simulations: focusing on field diffusion. Applied Artificial Intelligence, Taylor & Francis, 20(4-5):327-351, 2006.

[5] S. Bandini, G. Vizzari. Regulation Function of the Environment in Agent-Based Simulation. Environments for Multi-Agent Systems III, Third International Workshop, E4MAS 2006, vol. 4389 of Lecture Notes in Computer Science, Springer-Verlag, pp. 157-169, 2007.

[6] M. Batty, A. Hudson-Smith. Urban Simulacra: From Real to Virtual Cities, Back and Beyond, *Architectural Design*, 75 (6):42-47, 2005.

[7] J. Dijkstra, H. P. J. Timmermans. Towards a multi-agent model for visualizing simulated user behavior to support the assessment of design performance. Automation in Construction 11:135-145, Elsevier, 2002.

[8] J. Dijkstra, J. Van Leeuwen, H. J. P. Timmermans. Evaluating Design Alternatives Using Conjoint Experiments in Virtual Reality. Environment and Planning B 30(3):357–367, 2003.

[9] T. Ishida, Y. Nakajima, Y. Murakami, H. Nakanishi. Augmented Experiment: Participatory Design with Multiagent Simulation. IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 1341-1346, 2007.

[10] M. Mamei, F. Zambonelli. Motion Coordination in the Quake 3 Arena Environment: A Field-Based Approach. Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, vol. 3374 of Lecture Notes in Computer Science, Springer-Verlag, pp. 264-278, 2005.

[11] H. Nakanishi, S. Nakazawa, T. Ishida, K. Takanashi, K. Isbister. Can Software Agents Influence Human Relations? - Balance Theory in Agent-mediated Communities. *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, ACM press, pp. 717-724, 2003.

[12] P. Nugues, S. Dupuy, A. Egges: Information Extraction to Generate Visual Simulations of Car Accidents from Written Descriptions. In: Computational Science and Its Applications - ICCSA 2003, vol. 2667 of Lecture Notes in Computer Science, Springer-Verlag, pp. 31-40, 2003.

[13] F. Nunnari, C. Simone. Perceiving awareness information through 3D representations. Proceedings of the working conference on Advanced Visual Interfaces, AVI 2004, ACM Press, pp. 443-446, 2004.

[14] G. Papagiannakis, S. Schertenleib, B. O'Kennedy, M. Arevalo-Poizat, N. Magnenat-Thalmann, A. J. Stoddart, D. Thalmann: Mixing virtual and real scenes in the site of ancient Pompeii. Journal of Visualization and Computer Animation 16(1):11-24, 2005.

[15] G. Vizzari, G. Pizzi, F. Soares Corrêa da Silva. A Framework for Interacting Situated Agents in Virtual Environment, WOA 2007 (this volume).