Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220332

An Agent-Based Simulation Framework for Firefighters Training

Jordi SABATER-MIR [a,1], Ignasi CAMPS-ORTIN [a] and Cristian COZAR-ALIER [a]
[a] *IIIA, CSIC, Campus UAB, E-08193. Bellaterra, Catalonia (Spain)*

Abstract. In this paper, we introduce RHYMAS, a simulation framework that uses a multiagent approach and allows the creation of simulated emergency scenarios that mix artificial agents and humans. Those humans interact with the simulation using different types of immersive technologies, including virtual reality, simtables or virtual cave environments. The RHYMAS framework is being designed specifically for training firefighters and is being developed in collaboration with the "Escola de bombers i protecció civil de Catalunya".

Keywords. computer-based simulation, multiagent systems, autonomous agents

1. Introduction

The use of computer simulations as a tool for training qualified professionals on the skills needed in their jobs has a long tradition in many areas (healthcare, military, emergencies, transport, etc.). In general, these are areas where reproducing the training scenario in the real world is costly, dangerous or directly unfeasible. Recently, the interest in these kinds of simulations has greatly increased due to the improvement and cost reduction of immersive technologies like Virtual Reality (VR) and Augmented Reality (AR) that reduce the distance between reality and the simulation.

The RHYMAS framework is a framework under development in collaboration with the "Escola de bombers i protecció civil de Catalunya" specifically designed for firefighters training. The framework is being designed to allow for large training simulations that combine different tactical and strategic levels and that at the same time are easy to enact, easy to control and with a high training capacity. This is achieved approaching the simulations using a multi-scale paradigm and adding autonomous agents and multiagent systems technology to reduce the amount of human (simulation operators) intervention during the training sessions.

2. Example scenario

In this section we will present an example of the kind of complex training scenarios that RHYMAS is targeting. In this example, the emergency situation (based on a real case)

[1]Corresponding Author: Jordi Sabater-Mir, IIIA, CSIC, Campus UAB, E-08193. Bellaterra, Catalonia (Spain); E-mail: jsabater@iiia.csic.es.

starts at the Barcelona's harbour where a passenger ship hits a harbour crane that falls on a group of containers and starts a fire. The ship's crew manoeuvres quickly and takes the ferry to a safer area as far away from the scene as possible but with many injured among the passengers. In the meantime, a smoke column comes out of the fire in the containers, creating a toxic cloud that is spreading over the city.

At this point is where the training scenario starts. Initially there are two sectors: in sector 1 there is the ship with 120 injured people and in sector 2 there is the fire in the containers with the smoke column. A scenario like this has an enormous complexity. Figure 1 shows a recreation of the different human actors that could participate in the simulation.

Figure 1. Recreation of a human-in-the-loop complex emergency simulation.

At the first area (number 1 in Figure 1) you have groups of firefighters that are dealing face-to-face with the emergency (injured people, fire, etc.). At this level, the simulations are localised in a small area but need a lot of detail (convincing avatars and fire recreation, high graphical detail, etc.). Those are the firefighters that in the example would be dealing with the injured people and the ship in sector 1 and would be in front of the fire in sector 2.

The second area is populated by officers that direct groups of firefighters (number 2 in Figure 1). The groups of firefighters these officers are leading are humans participating in the simulation in the first area or groups of avatars controlled by the simulation engine. In the example scenario there would be at least one officer for each sector but if things go wrong, and the number of firefighters has to increase, a new officer layer is added to coordinate the different groups. The perception of the emergency scenario that the officers have at this level is from a certain distance but they are still located at the place of the emergency. Simulation detail needs to be quite high but not so detailed as it is in the first area.

In the example scenario there is also the added problem of a toxic smoke column that, after a while, spreads dangerously over the city. This requires the creation of a third level, a new command centre that will deal with that aspect of the emergency (number 3 in Figure 1). In this case, the perspective those officers have of the emergency is at a map level. By looking at the evolution of the smoke, the officers at this level provide instructions to the officers at level 2.

Finally, the last simulation level (number 4 in Figure 1) is composed by high rank officers. They receive reports from the other levels and determine the main course of action.

3. The RHYMAS architecture

Figure 2 shows the main elements of the RHYMAS architecture. The whole simulation is composed by a set of federated simulations. Each one of these simulations has its own independent simulation engine that is able to exchange information about its state with the other simulations. The different simulations are supervised by a sim operator that is responsible of high level management tasks.

A federated simulation is, at the same time, composed by the backend and one or several frontends. The backend implements the simulation logic and is responsible of communicating with/receiving information from the other simulations Backends. The frontends have two main functions: they provide different user views on that simulation (showing to the user the elements of the simulation that are relevant for him/her) and they can function as solvers for some specific aspects of the simulation, specifically physics, movement and perception. Separating this two aspects of the simulation has several advantages like the possibility of using state of the art game engines for the frontends but without conditioning the logic of the simulation to the game engine's idiosyncrasy.

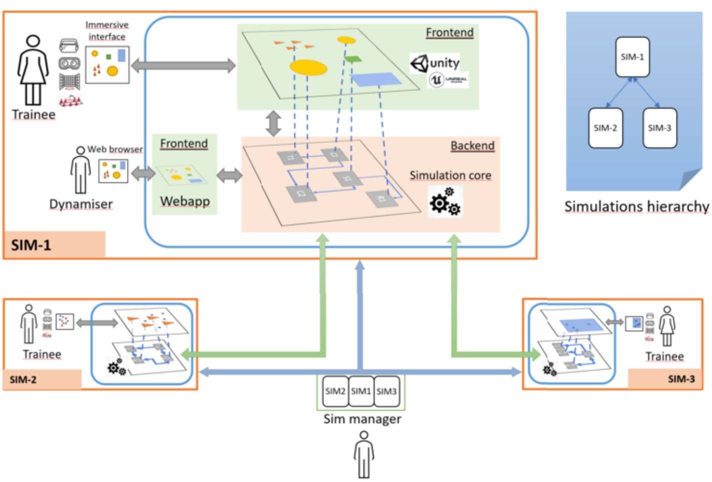

Figure 2. General RHYMAS architecture.

The counterpart of this separation between the Backend and the Frontends is that it increases the complexity of the architecture. An entity that is part of a simulation is split in two parts, the one in the Backend that we can see as the 'brain' of the entity and the one in the Frontend that we can see as the 'body' (sensors and actuators). The two parts of the same entity communicate between them using messages. So for example, when the Frontend part detects a collision with another object, if this is relevant for the simulation, it will send a message to the Backend part notifying it, probably indicating the ID of the other entity and other relevant information regarding that collision. On the other hand, if the Backend part decides that the entity has to move to a specific location, it will send a message to the Frontend part with the target location. The Frontend part will start the movement in the 3D environment taking care of the details, like for example which animation needs to be played or the calculation of the pathfinding. The Backend will be notified only if there is something associated to the movement that is relevant to the logic of the simulation.

A RHYMAS Backend is implemented as a directed graph where the nodes are the agents (elements) of the simulation (fires, firefighters, firetrucks, victims, etc.) and the edges the relations that exist among them. These relations represent any kind of influence that an agent exerts over another agent at a specific time step. A relation stores all the details that are relevant about that influence. Agents and relations are created and removed on demand following the evolution of the simulation. The Backend simulator engine is implemented as a traditional synchronous simulation with a main loop that each step asks sequentially the agents and relations to update their internal state. The state of an agent is the result of its internal model, the state of the incoming relations and the messages received from its counterparts in the Frontends. As a consequence of this update, apart from changing its internal state, the agent can create new relations and modify the details of the outgoing existing relations. Each simulation step the process is repeated.

4. Current state and future work

The framework is still under development. At the moment of writing these lines, we are working on a first prototype that implements the ideas described in this paper in a single simulation scenario like the ones used in the "Escola de bombers de Catalunya". The prototype is being implemented using Python for the Backend and the Unity game engine and different web frameworks for the Frontends.

The next step will be extending the framework to allow scenarios with multiple simulations at the same scale level. This will imply solving the problem of the communication between simulations and their synchronisation. Finally, we will extend the functionalities of the framework to include scenarios with multiple simulations at different scales like the one described in section 2.

Acknowledgements

This research has been funded by the Ministerio de ciencia e innovación "Programa Estatal de I+D+i Orientada a los Retos de la Sociedad" through the project "RHYMAS-Real-time Hybrid Multiscale Agent-based Simulations for emergency training" (PID2020-113594RB-100). Ignasi Camps is financed by a IIIA-CSIC JAE Intro Scholarship. We thank the people at the "Escola de bombers i protecció civil de Catalunya" for their advice in the scenarios definition.