SmartSim: Improving visualization on social simulation for an emergency evacuation scenario

Antonio M. Diaz and **Ganggao Zhu** and **Alvaro Carrera** and **Carlos A. Iglesias** and **Oscar Araque** [1]

Abstract.
The simulation of indoor evacuation is important for rescue and safety management, while a better visualization of simulation could help users to understand the evacuation plan better and to design the evacuation activities more effectively. The purpose of this paper is to show an indoor evacuation simulator with more realistic graphical user interface for both interacting and visualizing the simulation of evacuation plans. The proposed evacuation simulator combines a social simulation framework UbikSim and a character animation platform SmartBody. UbikSim is used as a back-end social simulation engine for evacuation scene management and evacuation simulation calculation such as computing agent positions and evacuation path. SmartBody is focused on various behaviours and capabilities of agents with digital 3D character in real time, which is used to visualize the locomotion, emotion and facial expressions of agents with more realistic animations in simulation. We develop a connector for SmartBody to control and visualize the simulation by communicating with UbikSim. The proposed evacuation simulator is validated in a real world university evacuation scenario with multiple simulation settings.

1 Introduction

Social simulation is a research field that applies computational methods to study issues in the social sciences. In social simulation, computers aim to imitate human reasoning activities by executing processes, mechanisms and behaviours that build the reality. This approach enables to investigate some complex models that cannot be investigated through mathematical models. Social simulation is considered as a third way of doing science, differing from both the deductive and inductive approach [1], in which simulating a phenomenon is akin to constructing artificial societies. Agent-Based Simulation (ABS) is a kind of social simulation that represents a simulation system as a society of agents that are designed to describe the behaviour of observed social entities such as individuals or groups [6]. Agent based social simulation is very useful to predict the behaviour of individual agents or crowds in complex environments, especially for simulating a dangerous environment and experimenting the possible results of some actions based on simple rules.

Various emergence cases can happen in a building such as fire, earthquake, water leak, and gas leak, to name a few. Crowd evacuations, such as disasters at massive parties, sport events and terrorist attacks can also lead to tragedies when performed without careful planning. In both type of emergencies, effective emergency evacuation is a key component of emergency response. Emergency evacuation preparation activities are required to be developed in advance because they ensure that people can get to safety in case of emergency. However, the evacuation demonstration in case of emergency is not always feasible because of ethical, practical and financial issues [8]. In order to define effective evacuation plans, understanding disasters and crowd emergency evacuation behaviours [3] conveniently with low cost, the ABS can be used to simulate the crowd behaviour and to analyze the effectiveness of evacuation plan. For instance, in a evacuation simulator, the building is modeled and populated by different numbers of agents representing various types of persons (e.g., handicapped persons, etc.) and common emergence situations such as blocked doors. Different agents behaviour according to predefined rules and the results of their actions are measured, hence the best of evacuation model can be selected according to simulation, without risking any real assets and situating human in dangerous situations.

UbikSim 2.0 [11] is such kind of agent-based social simulator to recreate the human behaviour inside a building. Specifically, UbikSim is used to model the map of the building where the emergency simulation takes place. Then, it simulates the virtual users (agents) under emergency and calculates the evacuation path for agents based on various criteria such as least crowd or nearest exit. However, UbikSim has limited features of graphical user interface in controlling and visualizing agents with abundant behaviours and various characters, where agents are represented as simple as equivalent figures in the map and there is no way to inspect visually the agents types (e.g. man or woman) or their emotions (e.g. fear or happiness). In order to enhance the visualization of UbikSim framework, we propose to incorporate SmartBody [12] to provide visualization of agents in an animation approach. More specifically, the agents in the map are represented as human-like 3D animations. The movements of agents can be demonstrated in a more realistic way and with more options such as walk, run or jump. Furthermore, agents are able to express emotions in their animated face and to response to events in an interactive life-like manner such as speak with gestures and face expressions. Moreover, the description of such behaviour is simplified by using Behavior Markup Language (BML) [9] because SmartBody is also a BML realization engine that transforms BML behaviour descriptions into real time animations. Consequently, the proposed system can provide a complete graphic rendering platform to bring various characters with predefined movement animations and behaviour sets together with a social simulation engine. In this way, we could add many different type of agents by simply adding their behaviour descriptions through BML settings. In addition, the system is also designed to be easily extended for future development.

[1] Intelligent Systems Group, Universidad Politécnica de Madrid, Spain, Spain, email: antoniom.diaz.dom@gmail.com, gzhu@dit.upm.es, a.carrera@dit.upm.es, cif@dit.upm.es, oscar.aiborra@alumnos.upm.es

To summarize, we would like to show the main contributions of this work:

- We propose and implement a novel agent-based evacuation simulator, named SmartSim[2], that combines an agent-based social simulator UbikSim with a character animation platform SmartBody.
- The proposed evacuation simulator has been validated and demonstrated in a realistic school building with different simulation scenarios.

The rest of this paper is organized as follows. In section 2, we present the background of this work. Section 3 gives the overview of the proposed evacuation simulator and discusses the implementation details. Finally, we describe the evaluation of the proposed simulator in a school evacuation scenario with different settings in Section 4 and close the paper in Section 5 by showing the main conclusions of this work and providing a possible view on future work.

2 Background

This section introduces some background knowledge of required components to develop the proposed evacuation simulator. We first review the idea and functionality of agent-based social simulator UbikSim in Section 2.1. Then, we introduce the character animation platform SmartBody and the behaviour description language BML in Section 2.2.

2.1 Agent-based Social Simulator UbikSim

Agent-based social simulation [5] is good at predicting the behaviour of agents in complex environments. Ubiksim 2.0 [11] is an implementation of an agent-based social simulator which has been developed by Universidad de Murcia and Universidad Politécnica de Madrid [3]. It is a framework that can be used to develop social simulation which emphasizes the construction of realistic indoor environments, the modeling of realistic human behaviours and the evaluation of Ubiquitous Computing and Ambient Intelligence systems. UbikSim is implemented in Java and employs a number of third-party libraries such as SweetHome3D and MASON. It consists of a console used to launch the simulation as well as a map in 3D or 2D where the position of all the agents involved in the simulation can be visualized.

Moreover, UbikSim tries to be a tool for using Multi-Agent Based Simulation (MABS) [4] in Ambient Intelligence (AmI) [10] which is a computerized environment that is sensitive to human and objects actions. MABS consists of modeling the environments with many artificial agents in order to observe the behaviours of agents, while it is possible to learn about their reactions. In case of evacuation simulation, effective activities can be derived from observing the behaviours of artificial agents and the outcomes of some simulated phenomena in evacuation. These behaviours cannot be observed in non-evacuation conditions. In contrast, other kinds of simulations model the entire environment as mathematical models where the individuals are viewed as a structure that can be characterized by a number of variables. Conventionally, it is not feasible to test a large number of users in AmI, whereas UbikSim enables the simulation of social behaviours from large group of users by applying the MABS approach to AmI environments.

As an example, Figure 1 illustrates a map used for evaluation based on UbikSim, including a demostration of an agent-based simulation.

[2] SmartSim Repository: https://github.com/gsi-upm/SmartSim
[3] Ubiksim Public Repository: https://github.com/emilioserra/UbikSim

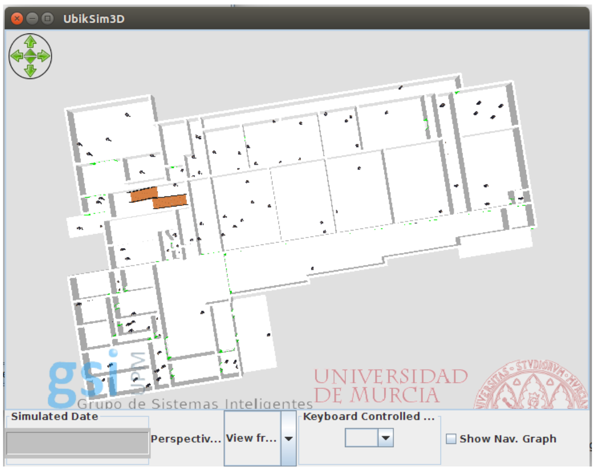

Figure 1. Example of Ubiksim framework for Agent-based Simulation

2.2 SmartBody and BML

SmartBody framework[4] is an open source character animation platform for animating Embodied Conversational Agents (ECAs) [12], which provides capabilities for digital 3D characters in real time such as the animations of locomotion, steering, object manipulation, lip syncing, gazing, non-verbal behaviour or re-targeting. SmartBody contains its own viewer and 3D renderer so that it can be run as a standalone system or incorporated into game or simulation engines. SmartBody is focused on proving various behaviours and interactive characters of artificial agents so we use it as graphical user interface of evacuation simulator, while UbikSim takes charge of scene management and simulation computation. In addition, the life-like behaviour requires the synchronized movement of multiple parts of the agents simulated body. For example, to realize the gaze behaviour requires coordination of eye, head, neck movements. Moreover, to support coherent interpretation of behaviour, the animation of gestures, eye flashes and speech audio must be synchronized in time with each other. SmartBody implements the behaviour realization engine that transforms BML behaviour descriptions into real time animations. As a consequence, we are able to have various predefined animations of agents with different types by describing their different behaviours with BML.

Listing 1. A BML Example

```
<bml id="bml1" characterId="Rachel">
  <required>
     <gaze id="gaze1" target="PERSON1"/>
     <speech id="speech1">
        <text>Welcome</text>
     </speech>
   </required>
   <head id="nod1" type="NOD"/>
</bml>
```

BML is an XML based description language for controlling the verbal and non verbal behaviour of ECAs [2]. BML is used to describe the physical realization of behaviours (e.g. speech and gesture) of the agents expressing them with movements that need to be realized by an agent. Those movements are single elements (e.g. gaze,

[4] SmartBody Web Site: http://smartbody.ict.usc.edu/

speech, head) and listed one after another, as exemplified in Listing 1.

3 SmartSim Simulator

The main goal of the proposed evacuation simulator is to use UbikSim as social simulation engine and to use SmartBody as graphical interface of the simulator. This section presents the implementation details of the proposed simulator system in Section 3.1 and also offers an overview of the simulation gateway in Section 3.2.

3.1 Architecture Overview

The SmartSim simulator system consists of a social simulation module (UbikSim), a simulation gateway module and a graphical visualization module (SmartBody). The overview of the proposed system has been illustrated in Figure 2. The idea is to connect the social simulation engine with the animation engine through the simulation gateway, in order to provide an integrated evacuation simulation system. We rely on the existing simulation engine, while we develop the simulation gateway and incorporate the animation engine into a complete graphical user interface for controlling, managing and visualizing the simulation.

The social simulation module is based on UbikSim and is used for managing agents, describing the emergency scenario, modeling the indoor evacuation environment and creating evacuation plan. The graphical visualization module is based on SmartBody and is used for visualizing the agent behaviour in life-like animation in simulation. To combine UbikSim and SmartBody, we implement a simulation gateway that helps to manage the social simulation configuration and to provide communication between UbikSim and SmartBody in real time while running the simulation. Moreover, a user friendly graphical user interface based on SmartBody has been implemented to utilized the simulation gateway so that end users can manage and visualize the simulation conveniently.

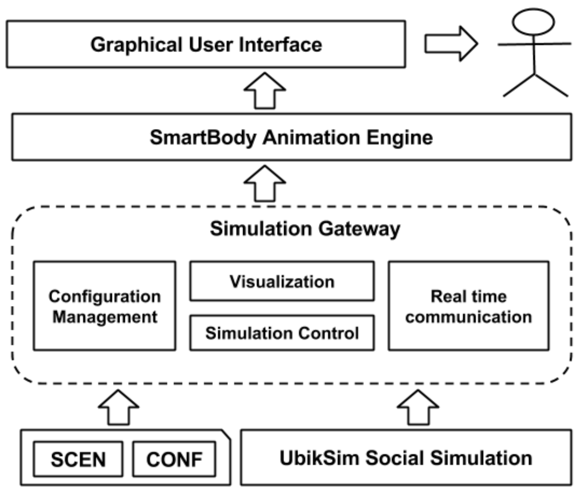

Figure 2. General Architecture of SmartSim simulator

In addition, UbikSim provides a scene editor that can pass the environment map to SmartBody. As SmartBody is not able to perform the simulation and calculate the paths that agents have to follow in order to evacuate the building. It relies on UbikSim to perform simulation computation and retrieves position data from UbikSim in real time. Within the map, and position data in run time, SmartBody presents a realistic 3D evacuation environment and enables users to control the simulation such as pausing or advancing.

UbikSim has many kinds of options, such as editing and creating artificial environment with an easy to use interface, configuring various number of agents. The communication between UbikSim and SmartBody is based on Representational State Transfer (RESTful) [7] architecture through web requests.

3.2 Simulation Gateway Implementation

The simulation gateway is composed by four different modules: simulation configuration module, graphical visualization module, real time communication module and simulation control module. The **simulation configuration module** parses user defined configuration of simulation such as agent numbers, emergency scenarios, initial positions and evacuation plans. Some relevant configuration options are illustrated in Table 1. Then, the configuration data are passed to UbikSim social simulator through web request API and to SmartBody through its Python API. According to the configuration data, social simulator initializes the simulation, creating the agents and setting their positions. The scenario resources are loaded to set the mark for emergency such as emergency position. The character resources such as skeleton and polygonal model are loaded for further usage of SmartBody.

Option	Description
amountAgents	The number of agents in our simulation.
amountLeaders	The number of leaders in our simulation.
ubikSimServer	The address of UbikSim server.
meshScenario	The scenario file for simulation.
modeSimulation	The possible simulation modes.

Table 1. Summary of SmartSim configuration options.

Based on configuration data, SmartBody creates the simulation scene (e.g. maps and agents) and starts the **graphical visualization module** that calls the graphic interface of SmartBody and a default camera to display the simulation. Moreover, the configuration module also loads the description resources for different character of agents from BML description resources, so the different type of animation can be rendered in simulation.

Option	Effect
output=web	Displays the web graphic interface.
control=pause	Executes the pause control.
control=play	Executes the play control.
control=stop	Executes the stop control.
control=frames	Starts the displayers in the server side.
position=people	Returns the agents positions.
position=map	Returns the map coordinates and obstacles.
position=emergency	Returns the emergency position and room.
position=(id,x,y)	Adds the agent to the position.

Table 2. Summary of UbikSim API actions.

The **real time communication module** retrieves agents' positions and paths from UbikSim and converts them into specific form of position and path for SmartBody. Consequently, the SmartBody can present the animation of agents that execute the evacuation plan. The UbikSim simulation run time Web API is illustrated in Table 2.

Furthermore, a **simulation control module** is implemented in SmartBody to control the simulation and make agents follow their path. It can be used to control every step of simulation and make agents *Pause*, *Advance*, and *Stop*. The actions of those commands are passed to UbikSim through the real-time communication module, so the gateway is able to coordinate the simulation in real time between UbikSim and SmartBody. After simulation finished, the simulation control module will record the simulation results containing the time that an agent spent to exit the building from its initial position and more relevant data for further analysis.

4 Use case scenarios

The implemented evacuation simulator system has been validated in a real use case scenario which is simulating evacuation activities. The indoor environment is selected as the building B of the School of Telecommunication Engineers (ETSIT) of the Universidad Politécnica de Madrid. A demonstration video of all the validation tasks can be found in YouTube[5]. The implementation of the simulator as well as validation case studies are published and available in a public Github repository[6]. We will first introduce how we create the validation map in Section 4.1 and then present three evacuation scenarios. Section 4.2 illustrates a single agent scenario where an agent escapes, leaving the building from any initial position of the building. Also, this section presents a more complicated case where multiple agents evacuate the building following a agent leader. Finally, we present a more realistic social simulation that different type of agents escape the building from different initial position and adopt different evacuation path.

4.1 Map Creation

The map of the building has been modeled in UbikSim and is illustrated in Figure 1. The generated map file has been exported to SmartBody in configuration. Note that any polygonal model generated with 3D modeling program such as Blender could fit the requirement of SmartBody. UbikSim editor is based on SweetHome3D which is a free indoor design application. We can draw the map of our scenario, arrange furniture on it and visualize the results in 3D. It is also easy to create a scenario as drawing the walls and rooms. Several objects libraries have been added and can be imported to the editor, which can add completion and detail of our scenario. We implemented an extension in UbikSim, so the created scene can be exported to SmartBody automatically. Figure 3 shows the 3D school map in SmartBody GUI that have been passed from UbikSim, using the map shown in Figure 1.

4.2 Single Agent Escaping

As mentioned previously, the simulation of agents is based on UbikSim, while SmartBody retrieves the paths and positions from it. By configuring scene in the UbikSim editor, we set the positions and numbers of agents, scenarios and location of the emergency. All these data is passed to SmartBody via the simulation gateway. UbikSim also retrieves the initial configuration from simulation gateway. The

[5] SmartSim Video: https://www.youtube.com/watch?v=8kGKD8Ofxuw
[6] SmartSim Repository: https://github.com/gsi-upm/SmartSim

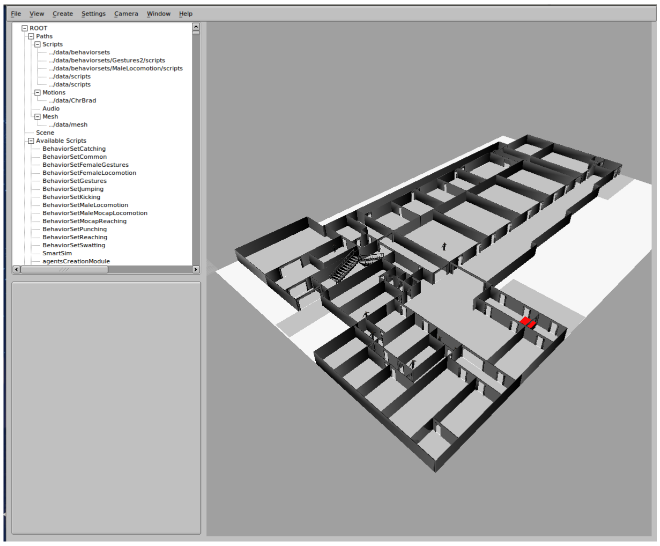

Figure 3. The map model loaded in SmartBody GUI automatically imported from UbikSim.

simulation result data are also generated by simulation gateway containing the exit time of each agent. We first validate the system in a scenario of evacuation of single agent from the building. Agent will escape the building following the path given by UbikSim. We demonstrate the emergency and the character escaping the building. The configuration of simulation is set as single agent without any character. The agent needs to exit the building from his initial position based on the predefined path. SmartBody is set up to show the evacuation of agent with animations, while simulation gateway will record the time the agent used to escape the building. This scenario is used to validate the system correctness.

The second scenario is the extension from the previous one by adding the number of agents and a simple evacuation model. The escaping in a crowd is a common phenomena in evacuation and is the main place that dangerous situation may appear. In the crowd simulation, we design a number of agents and one of these agents become the leader, while the other agents will follow the leader from their initial point to the exit. This scenario helps to extend the previous scenario with consideration of multiple agents.

It is a common phenomena in evacuation plans some crowds are leaded by a leader. The setting is similar to the previous case, while we also define the numbers of leaders and their following groups of agents. The non-leader agents will follow the path as their assigned leader. After simulation, the exit time of all the agents will be recorded. Figure 4 shows the animation of crowd escaping with leaders in SmartBody GUI. This scenario can help validate the performance of simulator with multiple agents and validate the correctness of evacuation plan execution.

4.3 Social Simulation with Characters and Emotions

Finally, we set a more realistic simulation scenario, where multiple agents with different type of characters escaping the building from different initial position following different paths. Figure 5 shows the screenshot of animation of agents starting from different location and execute different evacuation plans. It has been shown in Figure 5 that

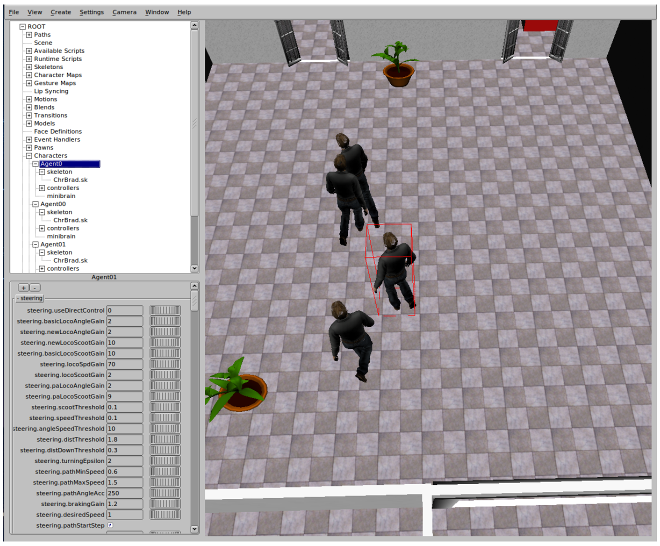

Figure 4. The Crowd Escaping

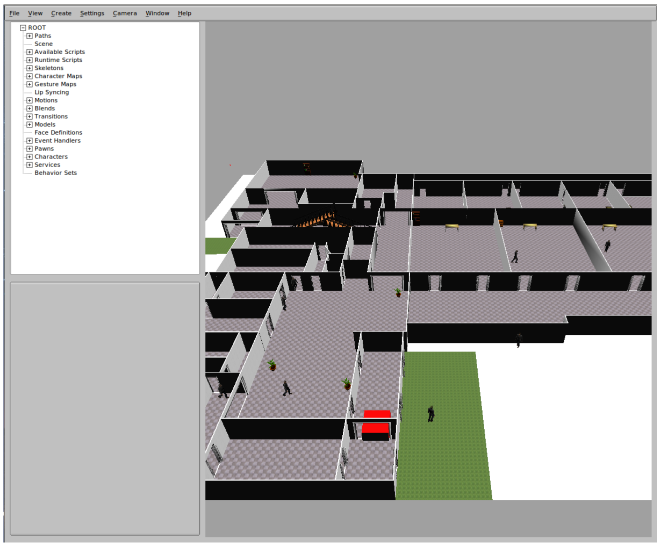

Figure 5. The agents escaping from different places

the simulator is able to present the social simulation of emergency evacuation correctly. Moreover, the visualization of the simulation become more realistic because there more kind of agents with different emotions. The previous scenario has leader and follower characters, while the agents can have different gender or ages. For example, Figure 6 illustrates a female agent named Rachel which is different from the previous male agents. The SmartBody and BML enable the animation of agents in a life-like way. By defining the behaviours in BML files, agents can have different motions and face expression to represent more human-like behaviours. For example, Figure 7 illustrates the agent expressing his happiness. This is achieved by configuring the face element in BML and realized by SmartBody. We believe that enabling the agents to express their feelings in face such as fear in facing an emergency and happiness after evacuation can make the visualization of simulation more realistic and help to make the evacuation plans better.

5 Conclusions and Future Works

This paper presents an agent-based simulation system, named SmartSim, for evacuations based on Ubiksim, where the graphical interface has been enhanced with realistic animations and emotions in agents using SmartBody.

The interaction between UbikSim and SmartBody, which allowed end-users to interact with simulation systems conveniently and visualize the simulation more powerfully, is implemented in different modules written in Python. The system is designed as modular components that can ease the future implementation of various simulation purposes. The system has provided facilities for creating simulation scenarios easily based on simple configuration file and those scenarios can be exported to UbikSim and SmartBody automatically. The visualization of simulation is achieved by very detailed artificial agents in animations. Furthermore, agents are able to express emotions and various behaviours which make our simulator more realistic. End users are allowed to select the numbers of agents as well as their types with particular animation and behaviours.

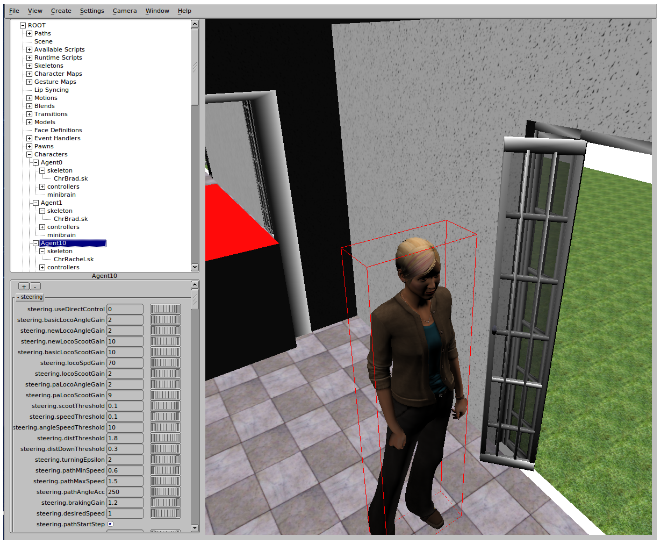

Figure 6. A 'Rachel' type character

Several research lines that can be considered as following work to continue and extend the features of this work. Firstly, a graphical interface for scene control might be useful to help users in avoiding mistakes in defining agent commands. Secondly, although SmartBody offers very good performance of visualizing agent's animations, it can be integrated with a graphical engine such as Unity to improve the quality of animation. Finally, apart from the current desktop version, we are planning to implement a mobile version or web-base version.

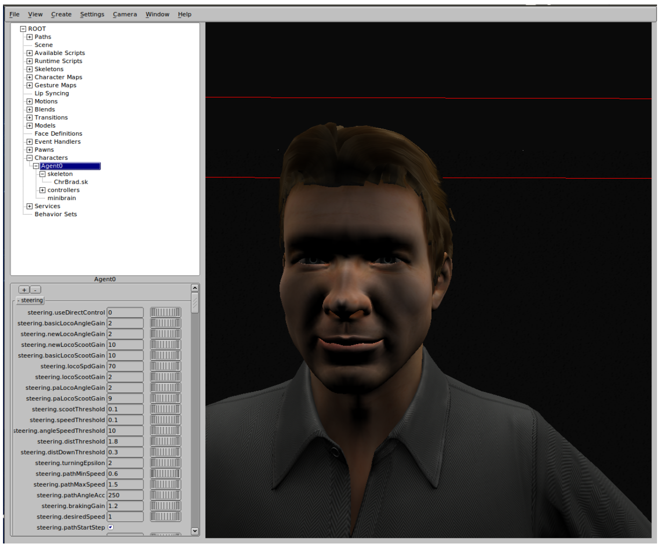

Figure 7. An agent expressing happiness

ACKNOWLEDGEMENTS

This work is supported by the Spanish Ministry of Economy and Competitiveness under the R&D projects SEMOLA (TEC2015-68284-R) and MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER).

REFERENCES

[1] Robert Axelrod, 'Advancing the art of simulation in the social sciences', *Japanese Journal for Management Information System*, **12**(3), (2003).
[2] Justine Cassell, *Embodied conversational agents*, MIT press, 2000.
[3] W Challenger, WC Clegg, and AM Robinson, 'Understanding crowd behaviours: Guidance and lessons identified', *UK Cabinet Office*, (2009).
[4] Paul Davidsson, 'Multi agent based simulation: beyond social simulation', in *Multi-Agent-Based Simulation*, 97–107, Springer, (2000).
[5] Paul Davidsson, 'Multi agent based simulation: Beyond socialsimulation', (2000).
[6] Paul Davidsson, 'Agent based social simulation: A computer science view', *Journal of artificial societies and social simulation*, **5**(1), (2002).
[7] Roy Thomas Fielding, *Architectural styles and the design of network-based software architectures*, Ph.D. dissertation, University of California, Irvine, 2000.
[8] Steve Gwynne, ER Galea, M Owen, Peter J Lawrence, and L Filippidis, 'A review of the methodologies used in the computer simulation of evacuation from the built environment', *Building and environment*, **34**(6), 741–749, (1999).
[9] Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R. Thrisson, and Hannes Vilhjlmsson4, 'Towards a common framework for multimodal generation: The behavior markup language', (2006).
[10] Emilio Serrano and Juan Botia, 'Validating ambient intelligence based ubiquitous computing systems by means of artificial societies', *Information Sciences*, **222**(0), 3 – 24, (2013). Including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems.
[11] Emilio Serrano, Geovanny Poveda, and Mercedes Garijo, 'Towards a holistic framework for the evaluation of emergency plans in indoor environments', *Sensors*, **14**(3), 4513–4535, (2014).
[12] Marcus Thiebaux, Stacy Marsella, Andrew N. Marshall, and Marcelo Kallmann, 'Smartbody: Behavior realization for embodied conversational agents', in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pp. 151–158, Richland, SC, (2008). International Foundation for Autonomous Agents and Multiagent Systems.