A SOFTWARE ENVIRONMENT FOR MICROSCOPIC PEDESTRIAN SIMULATION

Edgar F. Esteves, Rosaldo J. F. Rossetti, Eugénio C. Oliveira
Department of Informatics Engineering (DEI)
Artificial Intelligence and Computer Science Lab (LIACC)
Faculty of Engineering, University of Porto (FEUP)
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
E-mail: {edgar.esteves, rossetti, eco}@fe.up.pt

KEYWORDS
Pedestrian simulation, microscopic pedestrian modelling, multimodal transport exchange interfaces, multi-agent systems, OpenSteer.

ABSTRACT
The growth of both research on and the relevance of pedestrian simulation in the definition of strategies for urban planning increasingly reinforce its adoption and ultimate utility among practitioners and engineers. This paper reports on first steps and preliminarily results of the implementation of a platform for pedestrian simulation in multimodal exchange interfaces. Whereas most approaches in pedestrian simulation are aimed at studying intra and interactions among people using a common environment with different purposes and under different mobility restrictions, our research is focused on how those interactions will ultimately affect transportation operation at multimodal stations. This paper begins with a general overview of pedestrian simulation and focus on the microscopic representation of individuals and their interoperability as the basis for the implementation of our model.

INTRODUCTION
Nowadays the prediction of the influence of urban planning to people comfort and mobility inside public buildings is becoming a very important process.

Several studies were carried out in the last years as an attempt at predicting these influences. In this perspective modern pedestrian simulators become one of the most important tools to help one in achieving it. These tools have the objective of simulating and predicting pedestrian behaviours, as well as people or crowd flow in normal or in emergency conditions. Their main focus is on people movement inside public buildings such as train stations, shopping malls, airports and many others. The rising number of people attending certain public buildings and building restructuring makes the investigation of pedestrian movement very important in order to insure measures that will improve building capacity and better comfort to people using public building services by improving their mobility. For example, in a train station it can be very useful to know what kind of changes can be made in order to improve efficiency in interchange operations or even how many ticket boxes or ATM machines are necessary to reduce client queues. Another situation where pedestrian simulation can be very useful is to the design of emergency evacuation plans.

With the intention of modelling and simulating pedestrian crowd behaviours, it is often used two distinct levels of pedestrian analysis: macroscopic level, which examines the average characteristics of pedestrian flows; and microscopic level, which analyses the movement and behaviour on an individual basis. This paper aims to clarify the reader about the types of microscopic simulation often used and its place in the overall picture of pedestrian studies. It also aims to provide a basic idea of how to conceptualise such applications.

Throughout this paper we will analyse the literature related to this topic and some of the most recent platforms and tools for this kind of simulation. Finally, we will present a proposal for the architecture of a microscopic simulation platform, the preliminary results obtained, as well as some final remarks and conclusions.

PEDESTRIAN MODELLING AND SIMULATION
Pedestrian modelling studies started to gain relevance since a few decades ago. The main knowledge of pedestrian traffic systems comes from empirical and observation studies that can be consulted in greater detail in (Helbing et al. 2001) and (Helbing et al. 2002). According to (Teknomo 2002) the pedestrian studies can be divided into pedestrian data collection and pedestrian analysis as illustrated in Figure 1. The data collection focuses its attention in the observation and compilation of pedestrian movement while pedestrian analysis is concerned with interpretation of the collected data, both to understand an observed situation and planning (urban planning, for example).

As illustrated in the Figure 1, these studies have an analogy to vehicular traffic studies (Ferreira 2008) since they can also be divided into two categories, namely the macroscopic and microscopic abstractions (Klüpfel et al. 2000) (Keßel et al. 2002) (Shao 2006). The former compares the pedestrian flows with gases or liquids and studies the flow on global basis by examining the average speed and density whereas the latter studies focus on the individual speed and individual interaction comparing the pedestrian crowds with gas kinematics and fluid dynamics.

In this work, we focus our attention on the Microscopic Simulation Analysis Model (MSAM). It can be categorised
into four distinct types: Cellular Based Models, Physical Force Based Models, Queuing Network and AI-Based Models (Helbing et al. 2002).

Figure 1: Pedestrian Studies (Based on Teknomo 2002)

**Cellular Based Model:** In this model the pedestrians are simulated as a particle in a cell. The environment is represented by a grid of cells (e.g. 0.5m by 0.5m cells) and the pedestrian movement is modelled by jumping from one cell to another. The next cell is calculated according to two scores (Shao 2006) (Teknomo 2002) (Teknomo et al. 2000). Basically, a cost score is calculated on the basis of the proximity to pedestrians (similar to repulsive forces) in all the neighbourhood cells and the self standing cell, and a gain score based on the distance to target (expressing attraction forces). The disadvantage of this model is the discrete and static positioning possibilities for pedestrians. Nonetheless, it reduces the calculation processing time for positioning updates.

**Physical Force Based Model:** In contrast to the previous model, this one can predict the precise positions of the pedestrians (Quinn et al. 2003). There are two approaches to this model: Magnetic Forces and Social Forces. The former approach, developed by Okazaki and Matsushita, applies the magnetic models and equation of motion in magnetic field to the pedestrian movement. Both pedestrian and obstacles have a positive pole whereas the negative pole is assigned to pedestrian goals. Two forces affect each pedestrian movement allowing them to move to their goals and to avoid collisions as explained in (Teknomo et al. 2000). The Social Forces approach combines the principles of both the Benefit Cost Cellular Model and the Magnetic Force Model, and applies them to each pedestrian social forces (repulsive forces assigned to the other pedestrians and obstacles and attraction forces assigned to the goal) that combine actions such as motivation and creation of desire direction and acceleration. At each time instant, those values can be dynamically changed with the concern that the maximum and minimum values of velocity are never crossed. This approach assumes that every pedestrian has a desired goal or destination.

**Queuing Network Model:** This model is often used in pedestrian crowd simulations in evacuation situations (Helbing et al. 2002), for example in buildings. It is similar to a discrete space in the way that each room is designated as a node and the doors between them as links. Each pedestrian departs from one node, queues in a link and arrives to another node. The objective of pedestrians is to achieve the exit point as quickly and safely as possible. The choice of the next node/room is made based on a weight calculated by the density of pedestrians in the other room. If a pedestrian cannot pass to the chosen room, it waits or chooses another room as destination.

**AI-Based Model:** Such models complement all the others and bring new features to pedestrian simulation. The main contribution given to other models is the improvement of space searching queries and the pedestrian decision-making, which are implemented through algorithms such as A*, D*, Dijkstra, among other search algorithms. An often used approach to AI-Based models is the Agent-Based Model. In this model the basic unit of activity is the agent (Schelhorn et al. 1999). Generally such approaches use a large number of agents that represents a crowd of pedestrians allowing both interactions between agents and between them and the world to be modelled. The goal of this model is to build more complex human behaviours since realistic simulation requires a set of parameters such as psychological, physiological and social attributes (Sarmady et al. 2008).

The definition of the kinematics formulation is of vital importance to obtain accurate results with good performance measures in real-time simulations. Reynolds (1999) and Buckland (2005) divide motion behaviours for autonomous entities into three main layers:

1. **Action Selection** – Responsible for goal choosing and for deciding a plan to follow.
2. **Steering** – Responsible for trajectories calculation in order to accomplish goals and plans set by the previous layer.
3. **Locomotion** – Responsible for activating the necessary mechanisms that allow an entity to move in order to execute a certain action.

In (Reynolds 1999) the author justifies the abstraction division between the steering and locomotion layers as to anticipate the extensibility of new locomotion modules to be “plugged into” an entity. So, imagine an entity that was responsible for controlling a horse is now asked to control a motorbike. Its actions and steering behaviours will remain the same but the mechanisms to execute movement will change. So, the only change that must be performed is in the locomotion layer maintaining the action selection and steering layers untouched.

As already referred an autonomous entity is able to control moveable objects. Any moveable object is defined by the position of its point of mass and the velocity that can be changed by applying forces, as well as the maximum force (the maximum force that can be applied to a moveable object based on its own motor capacities), the maximum velocity and an orientation. An autonomous entity is able to steer a moveable object by applying behaviourally determined steering forces to the moveable object point of mass. Depending on the orientation and value of the resulting force, the acceleration is calculated and used to update the object old velocity in order to produce a new velocity that is truncated by the maximum velocity property value. Based on
the new velocity vector the object position is then updated as well.

Reynolds (1999) and Buckland (2005) suggest several basic steering behaviours that if combined can produce more advanced steering behaviours. The suggested behaviours (compiled and implemented into an open source library named OpenSteer that can be found in (Reynolds 2004) are summarized in Table 1. Readers are referred to (Reynolds 1999) for more details.

Table 1: Suggested Steering Behaviours

<table>
<thead>
<tr>
<th>Steering Behaviours</th>
<th>Seek</th>
<th>Flee</th>
<th>Pursuit</th>
<th>Evasion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Offset Pursuit</td>
<td>Arrival</td>
<td>Obstacle Avoidance</td>
<td>Wander</td>
</tr>
<tr>
<td></td>
<td>Path Following</td>
<td>Flee Field Following</td>
<td>Unaligned Collision Avoidance</td>
<td>Separation</td>
</tr>
<tr>
<td></td>
<td>Flocking</td>
<td>Cohesion</td>
<td>Alignment</td>
<td>Leader Following</td>
</tr>
</tbody>
</table>

Another concern inherent in pedestrian modelling studies is the representation of the environment in the virtual world upon which the pedestrians/autonomous agents/actors will act. It can be assumed two modelling approaches (Shao 2006) for the environment: Map Representations in Robotics and Models in Animation. Whereas the former is concern with the computational efficiency, the environment is represented in a simpler yet rich way. The most predominant representations are Occupancy Grid Mapping implementing grids with fixed resolution, Feature Mapping that represents features, such as points, lines, cylinders, corners and plans in a parametric way and Topological Mapping that uses graphs to represent the environment. Animation models are more complex and the environment information is more detailed. In these models, we can find accurate hierarchical topological structure from virtual geometric databases that makes the development of visibility computation possible, as well as neighbours detection, collision avoidance and optimised path planning algorithms. Synthetic vision is also used to make the actors capable of perceiving the environment. However, the best improvement that these models bring to the environment representation is to provide, beyond the geometric information, semantic concepts that help actors to better understand the world and act upon it in a more heterogeneous way.

SYSTEM DEVELOPMENT

Microscopic Simulation Platforms and Tools

Analysing Table 2, it is possible to distinguish between two types of applications: black-box applications (in this specific case they are all commercial applications with the exception of Micro-PedSim) and open-source applications. The former set of applications was only analysed in a higher level of detail so as to gather some of the most relevant functional requirements for this type of applications. On the other hand, the latter set of platforms was analysed in more details (e.g. architecture, modularity and performance), which was easily accomplished by analysing their source code.

Table 2: Microscopic Simulation Platforms & Tools Analysis

<table>
<thead>
<tr>
<th>Software</th>
<th>Open Source</th>
<th>Model</th>
<th>Runtime</th>
<th>Emergence Conditions</th>
<th>3D Visualization</th>
<th>Scenario Edition</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPS</td>
<td>no</td>
<td>Agent Based</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>VISSIM</td>
<td>no</td>
<td>Social Forces</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SimWalk</td>
<td>no</td>
<td>***</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Legion</td>
<td>no</td>
<td>***</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Micro-PedSim</td>
<td>no</td>
<td>Agent Based</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>AnyLogic</td>
<td>no</td>
<td>Agent Based</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SimPed</td>
<td>no</td>
<td>Agent Based</td>
<td>yes</td>
<td>***</td>
<td>***</td>
<td>yes</td>
</tr>
<tr>
<td>U&amp;F</td>
<td>no</td>
<td>Agent Based</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>PedSim</td>
<td>yes</td>
<td>***</td>
<td>yes</td>
<td>yes</td>
<td>***</td>
<td>no</td>
</tr>
<tr>
<td>OpenSteer</td>
<td>yes</td>
<td>Social Forces</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

In a higher level observation, it is possible to gather some important conclusions that help to distinguish obvious aspects of the most used applications. The edition and construction of simulation scenarios (through the use of a graphical user interface, GUI) are a strong point in all commercial applications that is translated into a higher efficiency and dynamics of the simulation. The 3D simulation in real time also enhances the application value to the simulation environment providing a better and more profound visualization of several key points of the simulation scenario. Focusing on two different simulation conditions (normal and emergency conditions) provided by the different platforms, it is possible to state that the integration of both, in the same application, features simulations with a higher degree of analysis.

An open-source application provides the possibility of the formation of a community that will experiment, analyse and contribute with new features to the application and so they become the primary focus of this analysis. As it can be seen, none of those applications integrates all the analysis features. So, focusing on the two last applications the evaluation of low level parameters may help us in deciding whether to use one of the adaptations or to adapt one specific tool by aggregating the best of both. The applications evaluation is summarised in Table 3.

Table 3: PedSim and OpenSteer Analysis

<table>
<thead>
<tr>
<th>Software</th>
<th>Modularity</th>
<th>Programming Language</th>
<th>Performance</th>
<th>Plug-in System</th>
<th>3D Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>PedSim</td>
<td>reasonable</td>
<td>C++</td>
<td>reasonable</td>
<td>no</td>
<td>***</td>
</tr>
<tr>
<td>OpenSteer</td>
<td>reasonable</td>
<td>C++</td>
<td>reasonable</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Analysing Table 3, it is possible to conclude that both platforms require complementary features in order to compete with the features provided by commercial tools, especially considering a graphical module for the construction and editing of simulation scenarios. With the objective of differentiating between the two platforms the analysis of the parameters described in Table 1 is a critical step to make a decision between both of them. Two parameters have a distinct evaluation, the Plug-in System and the 3D Engine.
The fact that OpenSteer incorporates a 3D engine for visualisation gives it some extra points when comparing it to PedSim, since this one requires an external 3D engine. Nevertheless the most important parameter that elected OpenSteer as the chosen platform to adapt was the fact that it provides a plug-in system that makes the integration of extra features to the application to be very easy (application extensibility).

Nevertheless, the OpenSteer platform is not a pedestrian simulator. This platform is a kinematic library that provides an example plug-in that contains a basic pedestrian simulator using this library. This fact is quite important to the creation of heterogeneous behaviours. An approach that uses the most significant features in the OpenSteer platform seems to be a very reasonable solution then.

System Architecture

According to the knowledge obtained through this research, the framework was not only conceptualised as a system capable of performing microscopic pedestrian simulation under normal conditions, but also with such a modularity that permits it to be used in further and different kinds of simulations, such as pedestrians simulation under emergency conditions or even vehicular traffic simulation. Figure 2 illustrates the conceptualised system architecture.

As it can be noted in previous the figure, the platform has six distinct modules: Simulation Editor, Model, Simulation Engine Controller, Autonomous Agents, 3D Visualization and Simulation Analysis modules.

The Simulation Editor Module (SEM) can be classified as information and manipulation software (Bret 2006) and features the interaction environment that interfaces the application and the end-user. The SEM must allow the model to be easily handled by the end-user then, from model construction to parameters configuration. It must also offer the appropriate visualisation of operations performed. Some challenges to the implementation of this module are to favour an easy integration and to implement a WYSIWYG-based (What You See Is What You Get) editor. This model is also composed of a set of features that contribute to these characteristics to be implemented. One imperative component that must be focused, however, is the scenario builder component (SBC). The SBC integrates capabilities such as to build scenarios (from very basic ones to more complex ones) in a structured way, from scratch. Through WYSIWYG, structuring scenarios follows a very intuitive approach, with all elements resembling their counterparts in real world (e.g. double lines for walls, single lines for sidewalks, icons for Points of Interest (PoI), and other appropriate representations for flow controllers (FC)) in scale with regard to the real scenario. The SBC is a vector-based environment, which facilitates the edition and handling of lines and other similar graphical representations. Also, positioning other elements, such as PoI, is just a matter of “dragging and dropping” it. Zones definition is also an important feature of this component, as they basically define areas through which pedestrians can actually walk. Apart from pedestrians, all objects placed in zones will belong to the specific zone to which the object has been attached.

The Model Module is an essential module that serves the purpose of integrating in an associative way both the simulation model and the real-world model. Basically, parameters are manipulated in such a way a reference model is always kept for further consultation. This module must support an easy creation and definition of new scenarios, and also allow one to navigate among different scenarios, in order to identify appropriate comparison points. Its structure contains the simulation scenarios, as well as the rules for some other components to work properly, according to the type of simulation one intends to carry out.

The Simulation Engine Controller Module (SEMC), on the other hand, is responsible for guaranteeing synchronisation, coordination and rule compliance throughout the simulation process, among all modules of the proposed architecture. The SEMC also integrates two important components, namely the 3D Visualisation and the Simulation Analysis modules. As for the SEMC operation, four essential elements must be present, namely the Model Data Manager Component (MDMC), Plug-in Manager Component (PMC), Agent Registry Manager Component (ARMC) and Simulation Manager Component (SMC).

The MDMC is responsible for reading, in a structured way, all information related to the model being simulated. To improve efficiency, we apply to the structure of the model the special queries structure proposed in (Reynolds 2000). The MDMC also interacts with the ARMC whenever a new agent must be registered in the simulation scenario. The PMC, on the other hand, allows plug-ins to be incorporated in the simulation environment. Its basic function is to manage the plug-ins and guarantee their proper use according to the type of simulation being carried out. The ARMC is one of the most important features of the SEMC, as it is responsible to manage all agents being simulated in a certain scenario. Whenever agents are transferred from zone to zone, the ARMC feature guarantees agent parameters will be updated accordingly. Boundaries conflicts that may arise from this interactions between agents and zones are also dealt with by the ARMC feature. Finally, the SMC feature is responsible for the simulation whole process, including synchronisation and coordination among the various other components that compose the system architecture. Synchronisation is basically achieved in accordance with the virtual time unity being adopted (i.e. time step), which forces all agents to keep their actions schedule appropriately ordered throughout the
simulation. After agents have defined the next actions to perform, they are actually executed in due time. This process accounts for the effects of each action performed, i.e. the transformation actions caused over the environment and their objects, which are updated accordingly. Thus, at the end of each time-step, the whole simulation scenario is updated and the time scale evolves to the next time step, with the whole process being repeated all over again until an end condition is reached.

As for the agent concepts used in this work, we based our approach on the work presented in (Reynolds 1999) and (Buckland 2005), in which agents are regarded as entities embodied within a moveable structure which they control according to their reasoning capabilities. Therefore, agents are autonomous to decide where to go and which path to follow in order to reach their destinations. To implement the reasoning kernel of an agent, we adopted the BDI (beliefs, desires and intentions) architecture, initially proposed by Rao and further discussed in (Kinny et al. 1996). This is one of the most popular architectures for cognitive agents, which has been applied to the domain of traffic and transports and other applications elsewhere (Rossetti et al. 2002) (Ronald and Sterling 2005).

Environment Representation Model

In this section we discuss on the approach adopted for the description of a general scenario meta-model, which is instantiated accordingly whenever a new scenario is built within the simulation environment. The meta-model was developed with special account of requirements for pedestrian simulation in multimodal transport interfaces. Thus, the elements involved in any public transport exchanging area were considered in the conceptualisation of this model. Nonetheless, the model structure is flexible enough to support the definition of other different scenarios, such as outdoor simulation of highly populated environments. In Figure 3, the meta-model structure is presented through a UML class diagram.

As the diagram suggests, a model can be built up from very elementary components. Despite its simplicity, such a structure allows great flexibility and eases the modelling task, even for large simulation scenarios (e.g. a whole building).

The movement delimiters are physical and rigid structures basically used to build the contour of the areas through which agents can walk. Examples of such elements are the walls and sidewalks, for instance. Walls are obstacles that avoid agents to transpose the area in which they are currently situated (for instance, it will avoid an agent to go from one room to another). On the other hand, sidewalks can be used to lay out desired paths we want agents to follow (as in open areas, such as streets, gardens, etc.).

The components/obstacles are virtually all the other objects one can find situated in diverse environments (e.g. banks and ATM, ticket offices and machines, bars/restaurants, and so on), excluding of course movement delimiters (as previously discussed). Obstacles can be understood as physical barriers, imposed by the intrinsic structure of certain components that oblige pedestrians to go around.

![Figure 3: Buildings Case Scenario Architecture](image)

Any area that can be delimited by movement delimiters are regarded as zones. Such areas can be labelled (e.g. restaurant zones) and can contain in their structure any other components. The connectors are quite related to zones as they are used to connect them, allowing pedestrians to move across several zones. The levels, in turn, aggregate zones in virtually independent environments, but yet can such an abstraction be connected to other levels. This is an important element of the model, especially when larger areas are being represented. For instance, different floor of a building could be easily represented by levels, whereas stairs or elevators could be used to connect them.

From the simulation point of view, there are some interesting abstractions made on certain components that are accounted for as a means to simplify the process of handling the simulation of a model. These abstractions, of course, will depend on the nature of components and their semantic relevance to the simulation study and to the agents (pedestrians) that take part in the simulation process. In general terms, there are two main categories of such abstractions, namely the Points of Interest (PoI) and the Flow Controllers (FC). The former generalise all components that are necessary as an aid, or sometimes even imperative, for an agent to achieve its final goal. Distinguishing between being essential and only being an aid will certainly result from the agents’ own goals. For instance, ticket machines are used mainly because passengers need to hold tickets whenever travelling in public transports. Thus, ticket machines would be essential in this perspective. On the other hand, a newsagent booth can be regarded as a place where travellers can potentially kill time before a train departs or just to get some newspapers and magazines for the journey. Not all people will aim the newsagent spot, for sure. In this
meta-category, we can include ATM and ticket machines, bars and restaurants, newsagents and other similar elements.

The FC are the generalisation of all components that propitiate people coming in and going out of the simulated environment. Examples of such components would include, for instance, main entrance doors, escalators, stairs to/from different levels, as well as buses, trains and taxis). These components implement dispatching and draining functions that must be calibrated according to the purpose of the simulation study being carried out.

As a means of improving realism in the simulation models and of adjusting them to different study purposes, it was necessary to design different set of rules in certain components. Rules are a simple and quite intuitive way of defining the behaviour of components, which are very accessible to the majority of end-users, even with no knowledge in programming. Their syntax can be as simple as “if <condition> then <action>”, which is reasonably intuitive to the majority of practitioners that work in simulation. We intend to further enhance this feature in our simulation environment as a means to allow user to define their own components and event extend the behaviour of the existing components. Even though such is already possible, as people can extend the application programme as it is open-source, we are working on the implementation of an intuitive interface and a specific-purpose rule syntax that will foster the modelling task in our framework.

PRELIMINARY RESULTS

A prototype was developed in C++ using the Qt framework that follows the main concepts of the OpenSteer architecture, with regard to the pedestrian behaviours and Plug-ins system. As a result of the implementation, the integration and the interaction of the several application modules were possible in the functional prototype proposed. This contains a very intuitive and easy to use editor that allows different simulation scenarios to be built from scratch. The environment then is well suited to run pedestrian microscopic simulations in a continuous space environment. A screenshot of the editor application is depicted in Figure 4.

As we can see in Figure 4, there are four zones exemplifying the most important components of this application, which are explained below.

1. **Zone 1 (ToolBarMenu):** This zone contains the scenario drawing tools to create walls, sidewalks, as well as moving and resizing objects capabilities.
2. **Zone 2 (ComponentDockWidget):** This zone contains the components that can be positioned and inserted into a simulation scenario. In the provided example, the user might place components as ATM machines, Coffee shops or Restaurants, ticket machines, pedestrian entry and exit points, as well as a connector that represents a door.
3. **Zone 3 (EditionWidget):** This zone contains the scenario drawing pane.
4. **Zone 4 (PropertiesWidget):** This zone provides a object’s properties edition dialog interface, which is activated according to the context of the object being edited.

The simulation application provides a 3D view of the environment, using the OpenGl technology integrated into the Qt framework. In Figure 5, we present a screenshot of a simulation being executed. The scenario was built through the editor module mentioned above and is the same as the model depicted in Figure 4.

The pedestrian behaviour model used is based on a stochastic plan allocated on a stack that is dynamically adjusted according to the agents’ basic needs. A probability of choosing each of the attraction components of the scenario is assigned to each of the agents in the synthetic population. During the simulation and after the achievement of each task, a new one is assigned to the agent, according to the pre-defined task plan.

The prototype is currently in a stable version and has a very intuitive interface. Nevertheless it still does not allow setting up the objectives/rules of pedestrian behaviour in runtime. Since this feature is not yet implemented in this prototype and for us to obtain some preliminary experimental results, the following simulation parameters were defined in the

![Figure 4: Simulation Editor Application](image)

![Figure 5: 3D Simulation Visualization](image)
generation of the agent population used in the simulation runs of this experiment:

- 20% of the created pedestrians will search and go directly toward an ATM machine.
- 50% of the created pedestrians will search and go directly toward a ticket machine.
- 30% of the created pedestrians will search and go directly toward an exit.

The pedestrians are also designed to go directly to a ticket office whenever they leave an ATM machine. After leaving the ticket office, they will go directly to an exit point. The ticket office and ATM machines were configured to allow a queue of five people. When this parameter reaches the defined limit, the pedestrians that wish to use the component will try to find one with a queue shorter than the imposed limit. If the pedestrian wishes to go to an ATM machine and all the ATM machines have a queue with five people, then it will wander the scenario until it finds one with a smaller queue too.

To better analyse the results of the simulation, a statistic component was implemented that draws two plots that represent the occupancy rates of the components and identifies the environment areas that have high occupancy rates. Two simulation runs were carried out and the obtained results are depicted in Figure 6 and Figure 7, respectively. Both are relative to one hour of simulation. The first scenario represents a room with an entrance, with three ticket offices, an ATM machine and two exit points and no walls or obstacles inside. The second scenario represents the same room but only with one ticket office, one ATM machine and two exit points.

![Figure 6: Example Scenario with no Obstacle](image1)

Through the analysis of Figure 6, is possible to obtain some basic conclusions. Analysing the plot that represents the occupancy rate of the environment spaces (represented by $a$ in the image), it is possible to observe that the zones with a higher occupancy rate are the areas where the ticket offices and the ATM machines are located. It is also possible to observe by analysing the component occupancy rates (represented by $b$ in the image) that the occupancy rate of the components (yellow for the ticket office and blue for ATM machines) stabilises by the eleventh minute of the simulated time and approaches the inputted parameter values, freeing considerably the exit points of the room.

This kind of analysis may help practitioners to understand if the scenario components are distributed well and if their numbers are enough to meet demand and the desired occupancy rates. The analysis of Figure 7 helps us to understand the effect that the number of components and their placement can have upon the occupancy rate inside the scenario.

![Figure 7: Example Scenario with Obstacles](image2)

From the occupancy zone plot and the occupancy rate of the components, it is possible to conclude that a single ticket office is not enough to deal with the amount of people trying to use it. This fact, associated with the inclusion of obstacles near the scenario exit points, creates a high occupancy rate in the entire scenario meaning that the exit rate after an hour of simulation is lower than the people entry rate (making the room completely crowded). Of course these are experimental results only but, despite the simplicity of the scenarios, they are good enough to demonstrate the potential of the proposed approach and framework prototype.

**CONCLUSIONS**

This kind of approach for the development of a microscopic pedestrian simulation platform has proved reliable, easy to implement and with some interesting results. Only part of the OpenSteer library was actually used in this work. To further improve and extend the presented framework, we intent to make full use and profit from all advantages of the entire kinematic library. As for the environment representation, it is important to bear in mind the relevance of the objects abstraction if the biggest concern is to ensure the reuse of the implementation of the steering behaviours and of the pedestrian decision-making processes, in both scenarios with similar and with different characteristics. Also, another useful scenario component that might play a key role to turn pedestrian behaviours into more heterogeneous characters is the “Connector”. That can happen if, parallel to a semantics graphical editor for each component, the pedestrian becomes

© EUROSIS-ETI
capable of reading, analysing and making decisions based on that information, which can be a possible analogy to the BDI architecture (Kinny et al. 1996). Finally, and really interesting is the particularity of the system to include a semantic editor. This can bring about the potential development of completely different simulations. That is, instead of defining semantics as referred before, which are almost exclusive to train stations, the user can choose to set up geographical information related to roads. From that point on, a vehicular traffic simulation looks more plausible. In addition to that, with similar scenarios and some visual changes, the convergence of such different kinds of simulations (namely traffic and pedestrians) becomes even stronger and more feasible.

ACKNOWLEDGEMENTS

Authors greatly acknowledge the kind support and contributions given by Prof. Álvaro Costa, from DEI/FEUP, as well as by Oana Santos and Vera Ferreira, from TRENMO Eng. Ltd. This project has been partially supported by RCM/Agência de Inovação.

REFERENCES


© EUROSIS-ETI