

GSPACES: A META-MODEL FOR SIMULATING AGENT MOBILITY IN URBAN OR ARCHITECTONIC DESIGNS

Klaudia Laffaille, Kay Tucci, Mayerlin Uzcátegui and Jacinto Dávila
Universidad de Los Andes
Mérida, 5101. Venezuela
email: {klaudia,kay,maye,jacinto}@ula.ve

ABSTRACT

gSpaces is a meta-model based in a DEVs formalism combining a full description of space, time and states with a discrete treatment of mobile agent to mobile agent and mobile agent to environment interactions. We called a meta-model a sort of template which can be instantiated into different, particular models of urban or architectonic designs. Mobile agents dynamics are modeled on a discrete level dividing the space into cells and the updating mobile agents speed by a movement function particular to each space. The movement function can be designed in order to distinguish different kinds of mobile agents in the system. The initial spatial characteristics of the designed space can be imported from a Drawing Interchange Format (DXF) ASCII file using a toolkit developed for gSpaces. During the execution of a model, the gSpaces meta-model allows for these spatial features to change, at run-time, to simulate the consequences of a catastrophic event.

KEY WORDS

Agent Mobile Simulation, Urban Spaces.

1 Introduction

Architects and city planners know how difficult it is to judge the adequacy of an urbanism to be located in a dangerous area. Nevertheless it is very necessary to be able to do so since, a great part of cities at world-wide level are located in dangerous areas. In this direction, architects and city planners are starting to manage concepts like preventive urbanism[4] and seismic vulnerability [8] in order to adjust their designs in case of events that would require an evacuation [6].

Now days, it is possible to develop computational models, using modeling and simulation techniques, that allow proving and comparing the feasibility and effectiveness of designs for the evacuation of mobile agents in case of catastrophes. There are two classic ways to define a model for this kind of systems. One way is using cellular automata [1, 7, 11], where space, states, and movement rules are discrete. The second one is modeling the system using continuous space, states, and movement rules [5]

to obtain more realistic models. In both cases the system evolves in constant discrete time steps. This article presents gSpaces, an hybrid and flexible platform to develop and simulate models for this systems, that combine characteristics of both classical paradigms.

The gSpaces platform is based on the DEVs modeling formalism and combines a full description of space and states with a discrete treatment of mobile agents to mobile agents and mobile agents to environment interactions.

gSpaces components are developed as module of the GALATEA simulation platform [10, 3, 9, 2] and can be used as a template that has to be completed with specific linguistic structures to produce a certain model.

2 The meta-model

In gSpaces the mobile agents are modeled using GALATEA messages, therefore each mobile agent is a distinguishable entity with its particular characteristics that can change during the simulation. The behavior of a mobile agent can be associated to its characteristics, allowing in this way different types of mobile agents behaviors in the same model. In addition to the usual fields of a message, the mobile agent has the following fields: x and y , that represent its position; the instantaneous speed V_x and V_y ; the visibility that represents the distance from where mobile agents can see a door; and a the next target to reach, that can be used to select one of many doors.

In gSpaces models the architectonic or urban space is divided into Spaces. Each Space contains: a list of mobile agents that are inside of it, a list of segments which represent its walls, a list of segments that represent its doors and an internal point of the space, which allows to define if any other point is inside or outside of the space. A Space is modeled using a continuous plane divided into spatial units called cells. The cells are square with user selected length side. Cells have a set of attributes that can be useful in order to move the mobile agents, for example the list of walls, ordered by distance, from smaller to bigger; the list of doors, ordered by the probability of being selected the amount of mobile agents in the cell, the obstacles index, and the preferential direction to follow in the cell.

One of the main assumptions to make when modeling with gSpaces is that all mobile agents inside a cell are equally affected by the cell's surroundings. Therefore, the

surroundings do not make any difference when treating mobile agents inside the same cell, even though their positions are different. Nevertheless, a message-agent may display a peculiar, different behavior, depending on the particular values of its attributes. This simplification allows to save great amount of calculations necessary in the molecular dynamic models like the model proposed by Helbing et al. [5].

Walls are unpenetrable straight segments defined by the coordinates of the segment extremes. Each Space has a list of Walls that restrict the movement of mobile agents inside of it. Walls are added to or deleted from a Space during the execution of the model allowing the simulation of events that can change its architectonic or urban spatial characteristics like earthquakes, tsumamis, fires or floods.

The Spaces also have a list of Doors. A Door is a straight segment that only allows mobile agents to exit from the Space where the Door is located and to go to another Space or to an exit. Like Walls, Doors are added to or deleted from a Space during the execution of the model. The Doors behavior is characterized by the amount of mobile agents that can pass through it at the same time; the time that a mobile agent needs to cross the Door; and a flag that indicates if the Door is bidirectional or not. A bidirectional door must have two entries in the model, one in each Space that connects through it. Doors are extensions of the GALATEA resource node, therefore Doors can use all the resource node facilities.

The mobile agents movement is modeled using a method called `move`. `gSpaces` has already some simple movement methods built-in into the implementation, but it is possible to implement new ones. The method calculates the new speed for each mobile agent inside the cell. In order to do the calculation, the `move` method can use the mobile agent attributes like its previous position and speed as well as all the information provided by the cell where the mobile agent is.

In order to move mobile agents the Space where they are has to be activated by a GALATEA event. When the Space is active its cells are scanned in a random way, and for each mobile agent inside the cell the `move` method is invoked. If the movement is normal, that is, the mobile agent does not pass through Doors or Walls, the new position of the mobile agent is given by

$$\begin{cases} x' = x + V_x \Delta t, \\ y' = y + V_y \Delta t, \end{cases} \quad (1)$$

where, (x', y') is the mobile agents final position; (x, y) is its current position; Δt is the time elapsed from the last activation of the Space, and (V_x, V_y) is the linear velocity calculated using the `move` method.

In order to check if a mobile agent displacement is normal or not, `gSpaces` verifies the existence of any intersection between the $((x, y); (x', y'))$ segment and each Wall and Door. If there are one or more intersections, the mobile agent is moved to the closest one. If the intersection is with a Door and the Door has enough free capacity, the

mobile agent is removed from the Space, is sent to Door, and an activation event is programmed in GALATEA's future events list according to the delay of the Door. When a Door activation occurs, all mobile agents in the Door that already spent their delay time in it are sent to another Space or to an exit according to the model.

As a module of GALATEA, `gSpaces` components can be used with other GALATEA components in order to develop more complex and sophisticated models.

3 Example

In this section the main `gSpaces` components used in order to make models are shown and explained. Figure 1 shows the architectonic space used as an example in this section. The architectonic space is divided in three Spaces: Outside, LeftRoom, and RightRoom. In the model they are declared as follows:

```
public static Space Outside =
    new Space("Outside", 2,
              1.0, 1.0, 2.0, 0.1);
public static Space LeftRoom =
    new Space("LeftRoom", 6,
              4.0, 4.0, 1.0, 0.1);
public static Space RightRoom =
    new Space("RightRoom", 6,
              8.0, 6.0, 1.0, 0.1,
              "MyRule");
```

where, the Space method parameters are: name of the space, number of mobile agents at starting time, an internal point coordinates, spatial resolution, and time step. In the RightRoom declaration there is one more argument, "MyRule", used to select the movement function in this Space.

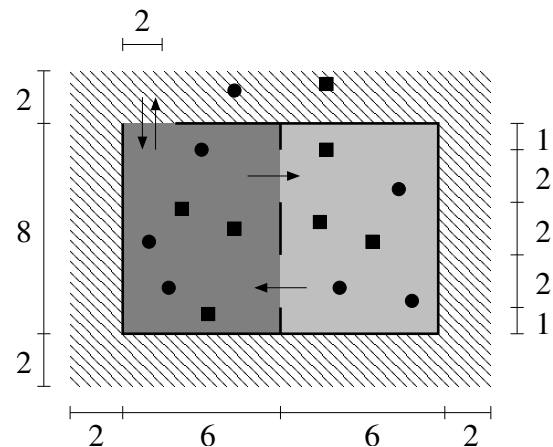


Figure 1. Architectonic space with 2 rooms and 1 external space. Between the rooms there are 2 doors, the arrows indicate their direction. In the left room there is a bidirectional door to go to or to come from outside. Circles and squares represent two different types of mobile agents.

The next step to make the model is adding Walls and Doors to the Spaces. For example in the Outside Space there are four Walls:

```
Outside.addWall(2.0,2.0,14.0,2.0);
Outside.addWall(14.0,2.0,14.0,10.0);
Outside.addWall(14.0,10.0,4.0,10.0);
Outside.addWall(2.0,10.0,2.0,2.0);
```

and in the LeftRoom there are five Walls:

```
LeftRoom.addWall(2.0,2.0,8.0,2.0);
LeftRoom.addWall(8.0,2.0,8.0,7.0);
LeftRoom.addWall(8.0,9.0,8.0,10.0);
LeftRoom.addWall(8.0,10.0,4.0,10.0);
LeftRoom.addWall(2.0,10.0,2.0,2.0);
```

Note that Walls shared by two Spaces have to be added in both of them. It is also important to mention that the unidirectional Door that allows mobile agents move from RightRoom to LeftRoom is a Wall from the LeftRoom movement rules point of view.

There are five Doors in Outside Space, one is the bidirectional door shared with LeftRoom and the other four doors are surrounding Outside Space that remove mobile agents from the system through a GALATEA exit node. These Doors are added as follows:

```
Outside.addDoor(2.0, 10.0, 4.0, 10.0,
  LeftRoom,'B',1,0.0, 1);
Outside.addDoor(0.0, 0.0, 16.0, 0.0,
  exit,'U',10,0.0, 1);
Outside.addDoor(16.0, 0.0, 16.0, 12.0,
  exit,'U',10,0.0, 1);
Outside.addDoor(16.0, 12.0, 0.0, 12.0,
  exit,'U',10,0.0, 1);
Outside.addDoor(0.0, 12.0, 0.0, 0.0,
  exit,'U',10,0.0, 1);
```

where, the first four addDoor method parameters are the door extremes coordinates; followed by the Space or node where mobile agents that use the Door are sent, the door capacity, the time needed to cross the door, and a coefficient that can be used in the movement function to select between several Doors. Doors in the other two Spaces are added as follows:

```
LeftRoom.addDoor(8.0, 7.0, 8.0, 9.0,
  RightRoom,'U',1,0.0, 1);
RightRoom.addDoor(8.0, 3.0, 8.0, 5.0,
  LeftRoom,'U',1,0.0, 1);
```

Note that a gSpaces bidirectional Door is defined only in one of the Spaces that share it but it is added in the Doors list of both Spaces, while an unidirectional Door is only defined and added in the Space where the mobile agents go out through it.

Once Walls and Doors are added, the cells in the three Spaces are created by

```
Outside.build();
LeftRoom.build();
RightRoom.build();
```

Finally, standard GALATEA code is used to complete the model

```
Glider.setTsim(100);
Glider.act(creator,0);
Glider.act(Outside.getMove(),0);
Glider.act(LeftRoom.getMove(),0);
Glider.act(RightRoom.getMove(),0);
```

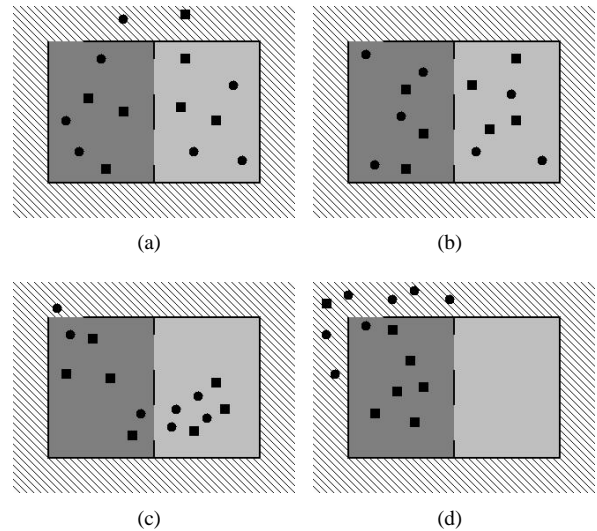


Figure 2. Snapshots of the simulation of the example model. a) represents the initial state of the simulation, b) indicate a previous times of the events that would require an evacuation, c) shows the next instants on evacuation process, d) The fastest mobile agents (circles) already are outside whereas the other agents (squares) remain in the architectonic space.

Figure 2, shows the system status for four different simulation times in the same realization. Two different types of mobile agents are represented, the faster ones by circles and the slower ones by squares. Before the evacuation start (Fig. 2(a)-2(b)) both mobile agents types have the similar behavior. After events that would require an evacuation (Fig. 2(c)), all mobile agents go to the exits doors. Figure 2(d) shows that the faster agents (circle) reach the exits before the slower ones (squares).

The skeleton of the model can be generated, by means of the dxf2g tool, from a Drawing Interchange Format (DXF) ASCII file, commonly used by the architectonic design environments. In this way, modelist would only have to establish the resolution of each Space and set some dynamic parameters like the amount and type of mobile agents in each Space at initial time of the simulation; the capacity and delay time that the Doors have; the movement rules of the different mobile agents types; and the model simulation time.

4 Conclusions

We have presented gSpaces, a hybrid and flexible meta-model for simulating agent mobility in urban or architectonic designs. The spatial characteristics of the system are added to or deleted from the model during its execution allowing the simulation of events that can change them. The gSpaces components are implemented as a module of the GALATEA platform, allowing development of complex models of architectonic or urban spaces with several types of mobile agents in them. For users who are not familiarized with GALATEA is relatively easy to create models using tools like dxf2g, since they can be dedicated to modeling the system dynamic, thus avoiding its spatial aspects, as well as programming details of the model.

We have tested the meta-model, by generating several actual models of simple urban designs. A more comprehensive model of an existing urban design is our next target.

Acknowledgements

This work was supported in part by a grant from the CDCHT-ULA projects I-666-99-02-E, I-667-99-02-B and C-1283-04-02-E.

References

- [1] O. Biham, A. A. Middleton, and D. Levine. Self-organization and dynamical transition in traffic-flow models. *Phys. Rev. A*, (46), 1992.
- [2] Jacinto Dávila and Mayerlin Uzcátegui. Gloria: An agent's executable specification. *Collegium Logicum. Kurt Gödel Society*, VIII:35–44, 2004. Vien, Austria.
- [3] Jacinto A. Dávila and Mayerlin Uzcátegui. Galatea: A multi-agent simulation platform. *AMSE Special Issue 2000. Association for the advancement of Modelling & Simulation techniques in Enterprises*, pages 52–67, 2002. Lion, France.
- [4] J. Figueroa. Urbanismo preventivo enfocado para santiago chile, 2003. Available from: <http://urbanismo.8m.com/>.
- [5] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, (407):487–490, 2000.
- [6] Klaudia Laffaille. Lineamientos espaciales urbanos para disminuir la vulnerabilidad sísmica en áreas destinadas para nuevos desarrollos, 2002. Tutor: Cova, Pablo.
- [7] K Nagel and J. H. Herrmann. Deterministic models for traffic jams. *Physica A*, (199), 1993.
- [8] M. Safina. *Vulnerabilidad sísmica de edificaciones esenciales. Análisis de su contribución al riesgo sísmico*, chapter 13. Departamento de Ingeniería del Terreno, Cartografía y Geofísica. UCP. Salvador., 2003.
- [9] Mayerlin Uzcátegui, Jacinto Dávila, and Kay Tucci. *Simulación Multi-Agente con GALATEA*. 2004.
- [10] Mayerlin Y. Uzcátegui. Diseño de la plataforma de simulación de sistemas multi-agentes galatea. Master's thesis, Maestría en Computación, Universidad de Los Andes. Mérida. Venezuela, 2002. Tutor: Dávila, Jacinto.
- [11] D. E. Wolf, M. Schreckenberg, and A. editors Bachem. *Traffic and Granular Flow*. World Scientific, 1996.