Geographical Information Systems and Dynamic Modeling via Agent Based Systems

Cláudio Antônio da Silva
Pontifical Catholic University of Rio de Janeiro – PUC-Rio, Computer Graphics Technology Group - TECGRAF-
Rua Marquês de São Vicente 225, 22453-900 Rio de Janeiro, RJ, Brazil
cads@tecgraf.puc-rio.br

Roberto de Beauclair Seixas
Institute of Pure and Applied Mathematics – IMPA, Visualization and Computer Graphics Laboratory - VISGRAF -
Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, RJ, Brazil
tron@impa.br

Oscar Luiz Monteiro de Farias
Post-Graduation Program in Computer Engineering,
Rio de Janeiro State University,
Rua São Francisco Xavier 524, bloco D, sala 5028, Maracanã
20550-900 Rio de Janeiro, Brasil
fariasol@eng.uerj.br

ABSTRACT
A full integration among Geographical Information Systems and Agent Based Simulation Tools (software where entities of the real world can be simulated through the use of computational entities called agents), established around open source software and interoperability standards, would give to dozen of scientists, not necessarily computer experts, access to powerful simulation tools. Unfortunately, till this moment, there are few applications that effectively implement this kind of integration [12], [10], [22], not to speak in a generic tool. In this paper we try to explore this subject, presenting two agent-based simulation models, implemented in C programming language, that are somehow integrated with Geographical Information Systems (GIS). The first one is the movement of pedestrians in an urban environment, whose foundations are based mostly in empirical results of the literature [14], [23], [6], [29]. The second one is the maneuver (steering behavior) of soldiers in a terrain, guided by some simplified rules of combat [8][27].

In both models, the movement of the actors, soldiers and pedestrians, occur in real environments, that is, in regions of a geographical space. So, it is very natural that we seek to have at hand of our agent based models all the data commonly stored and organized in GIS's, because GIS's usually have a set of specific tools to build and manage geographical space representations in a more precise manner. In the two models we used a common architecture to implement the agents. This architecture can be considered a rudimentary nucleus of an agent based simulation tool. Our main goal was to gain experience, theoretical and practical, in problems that arise when integrating GIS's and Agent Based Models (ABM's). This experience could, in the future, guide us in developing a true integrated software tool among GIS's and ABM's.

The experience with these two prototypes leads us to some conclusions. The first one, already mentioned in the literature [5], that is possible to simulate complex systems having as a starting point only simple elements (agents), embedded in some type of environment. These agents can interact which each other in only specific and very simple ways. As the result of the interaction among the agents and the environment, the behavior of the complex system then emerges. According this line of thinking, there are a lot of social and ecological processes that can be simulated by agent-based models. Another important point is that the computer code needed to a particular application should not be developed from scratch. It is very feasible the gradual building of a library of generic perceptions, actions, and communications for agents. From that library, we could select only the particular behaviors we wish to incorporate to a given model; Third, and undoubtedly, most important, there is an urgent need of integration among GIS's and generic agent-based simulation tools. Currently we are working in this line of research, for it opens new perspectives for dynamic modeling and study of spatial-temporal phenomena using data stored in current GIS's data pools.

ACM-GIS'05, 13th ACM International Symposium on Advances in Geographical Information Systems, November 4–5, 2005, Bremen, Germany.

Categories and Subject Descriptors
I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Intelligent agents
I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence - Multi agent systems
I.6 [**Simulation and Modeling**]: Applications

General Terms
Design, Standardization, Experimentation

Keywords
Autonomous agents, agent based models, integration, GIS, simulation tools, urban modeling, military modeling

1. INTRODUCTION
This paper shows the possibility of integration among agent based models and Geographical Information Systems (GIS), with the objective of simulating dynamic systems directly from data organized and stored in GIS data pools [12], [10]. Agent based models could, in this approach, be thought as a tool for spatial analysis, or a way of incorporating the analysis of temporal phenomena through the use of traditional GIS.

In this paper, an agent based model consists in a collection of autonomous agents (or simply agents) which interact between themselves ruled by some simple characteristics that are modeled from the observation/analysis of real world entities in order to simulate their more complex characteristics. Agents can be considered as computational entities that can conceptually incorporate in a natural way some human mechanisms such as: perception, action selection, autonomy, etc. This kind of simulation, based on agent models, and a formal definition of agents are presented in section 2.

There are several advantages on this approach integrating agent based models and GIS: i) the simulation of complex models, e.g. traffic systems, movement of herds, wildfire propagation, different kinds of social and ecological processes, via the so-called micro-simulation, where we model only the behavior and interaction of the simple components of the complex system on a given environment; the behavior of the complex system at macroscopic level then emerges as the result of the interaction of its single components [1], [5]. Sometimes an analytical model is very difficult, if not impossible, to attain, or its mathematical complexity turns it very difficult to use. So, the use of micro-simulation models is a very practical and useful way of simulating complex systems and to obtain results that would be impracticable to achieve via the use of other methods; ii) very often the impediment of simulating some phenomenon lies in the difficulties (including costs) associated to gather "real" data about the phenomenon under interest; so, the use of data already stored and organized in GIS data pools would circumvent all these obstacles; iii) GIS, in spite of their widespread use and acceptance, is basically a static and bi-dimensional system. Adding tools capable of simulating dynamic phenomena would be a form of incorporating the analysis of temporal phenomena by GIS [1], [2].

In the integration among GIS and agent based models (ABM) we need, firstly, to build an environment, that is, the place where the agents interacts which each other. Secondly, the agents must be built with some mechanisms of perception and action similar to the real objects that we want to represent. It must be emphasized that the agents and the environment are strongly connected, that is to say, for each kind of phenomena, the agents are interested only in particular objects of the real world. For example, in a traffic simulation system the agents would be interested in information about streets, routes, and semaphores. Their movements (agents representing vehicles) will be allowed only in the limits of the streets, avenues, etc. If we want to simulate soldiers in a terrain, the movements of agents are not limited to streets, avenues, etc. In this last case, the agents have a greater degree of freedom in their movements. But certainly they are interested in a new kind of information, related to the elevation of the terrain. So, the information concerned to the digital elevation model of the terrain would be essential to build the environment, in which the agents will simulate the real objects, the soldiers.

Generally speaking, all the phenomena we know in our tri-dimensional world occur in a spatial temporal continuum. For each specific phenomenon we are interested only in particular information about the environment. So, considering the geographical space where the phenomenon develops, geographical space probably already represented on a GIS, we need to filter only the aspects of the environment we are interested in. We need only to select the features that will compose the environment as seen by the agents, and where all the process will be simulated. Here we can think of a coordinate system, added with the features we imported from GIS. In this coordinate system, via an ABM, the temporal phenomenon would be, then, simulated.

At first, we could imagine that each kind of problem to be simulate is completely different from the others problems, having its special idiosyncrasies. Aside from using the GIS's data pools for representing the environments of the system to be modeled, we could think that in the proposed GIS-ABM system integration, all the computer code needed to a particular application should be developed from scratch. Fortunately, this is not true, as several tools for building agent based models evidence (Swarm [18], Starlogo [27]). Therefore, we could have, in the long run, the gradual building of a library of generic perceptions, actions, and communications for agents. From that library, we could select only the particular behaviors we wish to incorporate to our agents. In this way, a lot of code could be reused.

To demonstrate the viability of the above proposal we developed two agent based model systems integrated with GIS. The first one simulates soldiers moving around a terrain, based on some simplified warfare rules; the second, pedestrians walking in an urban region, based on some empirical results from literature. With the accurate study of these two applications we can generalize the GIS-ABM system integration proposed in this paper.

2. THE AGENTS AND THE SIMULATION ENVIRONMENT

Using micro-simulation it is possible to simulate the movement of objects in a geographical space, starting with a model of the movement of a single object. In this way we can model complex systems, such that: movement of pedestrians in an urban region, evacuation of people in case of sinister, traffic of vehicles in urban regions, movement of soldiers in a terrain, movement of navies on the sea, etc. Therefore, is very important to understand the movement of a single object or element, in order to understand the behavior of the complex system under consideration. This behavior is expected to naturally emerge from the interaction of dozen, hundreds, or thousand of elements. It´s not an easy task but results as presented in [21] shows that micro-simulation is a good alternative to simulate complex systems. The concept of autonomous agents (or simply agents) can be used to model the mentioned single object or element in micro-simulations.

A formal definition for autonomous agent is a very controversial topic [30][7]. An acceptable definition of agent for the purpose of this work is given by Franklin and Graesser [4]. They defined agents as: "a system situated within and part of a environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to affect what it senses in the future". As one can see, this definition mentions common human characteristics what allow us to conclude that agents can be used, among other applications, to simulate human behavior in the many complex systems he actuates. In order to make this task well defined, it's important to define a model.

The common characteristic of the objects or elements in the complex systems mentioned before is their movement in a geographical space. So, it is possible to define a generic agent model, in which the elementary agents have some basic behavior related to the movement in a geographical space. Later on, depending of the system under study, this generic model will be expanded, for incorporating specific behaviors associated with the problem under consideration. Our generic agent model has three components: i) its internal state; ii) a perception module; iii) a behavior module.

In the internal state are stored the agent properties related to its goal, motivations and current situation. Of course, these properties are defined in function of the characteristics we want to incorporate to the agents. In our study we were interested only in the aspects related to the movement of objects. The properties defined were: i) orientation; ii) maximum angle of turn that the agent can support between different cycles of the simulation; iii) maximum speed; iv) instantaneous speed; v) position, that stores the current coordinates of the agent; vi)goal; that defines a vector in the direction of the agent goal. The angles are defined relative an imaginary vertical axis that passes by the current position of the agent. These *internal properties* can be modified, depending on what the agent perceives and also in its interaction with other agents.

The perception module gives the agent the capability of navigating in their environment avoiding collisions and deciding what action(s) to adopt. For doing this, the agent must be capable of analyzing its surrounds through its mechanism of perception. The basic one, we included in our model, was the ability to *see*. Our agents can see the objects and perceive the environment situated inside a circle of a radius *r*, whose center coincides with its current position. If desired, it is possible to add more *senses* to our agents.

The autonomous action mechanism of our agents is implemented through specific behaviors. In our two case studies, we want the agents be capable of moving themselves on a given environment, avoiding obstacles, avoiding collisions with other agents, and pursuing some goal, that is, they try to reach some position to satisfy their objectives. One class of behavior, the so called *steering behavior*, was employed here. It is very important to understand that the behavior modeling is somehow dependent of the phenomenon under consideration. So, the behaviors needed to simulate the movements of soldiers around a terrain will be different of the behaviors needed to simulate the pedestrian movement in downtown. However, it is possible to define a set of generic behaviors, which can be used independently of the phenomenon to be simulated. These behaviors define primitive actions and can be associated to build more complex and specific behaviors to be applied to a certain phenomenon. Reynolds [20] presents several generic steering behaviors. In our implementations we need only, some of them: *seek, flee, pursuit, path follow*, and *obstacle avoidance.* The use of these primitive behaviors made easier the implementation of our two case studies.

The simulation environment gives us the *locus* where our agents move, interact, and pursue their goals. These agents *mimic* the behavior of people in a particular geographical region. On that ground, the environment definition consists in selecting the spatial objects (rivers, streets, buildings, etc.) relevant to reproduce the geographical region and, to study the phenomenon we are interested in. Objects that are not of interest must be discharged, for an environment full of superfluous objects would unnecessary complicate the implementation and reduce the simulation performance. We should always be in mind, that our agents must be capable of interacting with the representation of the objects selected. Inadequate representations could lead to complex data structures to store the environment information, and complex algorithms to implement the agents´ behaviors. In Reese [19] we can find some observations about the representation of objects and their implications.

Though there are several antrophic studies using agent based models, there are yet only few of then [2], [10], [12] that use *real data* to define the simulation environment. Here, we understand for *real data*, data that preserves the position and the form, of the objects of the geographical region under analysis. *Real environments* are those simulation environments defined on the basis of *real data.* Obtaining *real data* can be done via a field survey or via data pool repositories already built. Field survey is a highly cost activity, requires procedures and specialized equipment and, most important, demands time. Meanwhile, the use of data repositories requires only the decodification of the format in which the data were originally stored. The cost and time savings are evident in this later alternative. We have chosen this last option and we will use GIS's to obtain the data needed by our ABM's.

There are some data formats with widespread use in the market, standing out the *Arcview Shapefile*, for vectorial data, and the *ArcView Binary Grid*, for matrix data. We chose to gather data from these two formats basically for two reasons: i) the lack of an interoperability specification for GIS's data of widespread acceptance; ii) The amount of data already at our disposal in GIS's repositories that share these formats.

3. THE ARCHITECTURE FOR PROTOTYPE BUILDING

It is very desirable that the researchers (they are not necessarily computer experts) have access to some computational tool to create their agents on a given environment and to simulate specific phenomena. Some tools, like Swarm [18], and StarLogo [28] give the user some of these facilities. But, till this date, there isn't a generic agent tool that operates in an integrated manner with known GIS's systems. For this reason, in our exploratory study we decided to design an architecture for agent based systems and implement our prototypes based on it. This is a simplified architecture, because it doesn't tackle some questions already discussed in the literature, such that: communication among agents, memory mechanisms, adaptability of learning, etc. However, this architecture can be easily expanded to attend some of these questions. Our architecture comprises three distinct layers: i) the environment layer; ii) the goal and motivation layer; iii) the action layer. These layers formalize how the components of our agent model (internal state, perception and behavior) should act during the simulation.

The environment layer represents the simulation environment, and embodies all the spatial objects (in their matrix or vectorial form) and agents. This layer is referred, at the beginning of each simulation cycle, by the perception module, because the agents need to get information about the regions on their surrounds. This information will be used later, to select the actions the agents should adopt. This layer could be expanded to incorporate a spatial index to accelerate spatial searches, or a communication bus to allow communication among agents.

The goal and motivation layer implements the intelligent of our agents. The purpose of this layer is to analyze the information gathered from the environment layer and, based on the motivations and internal state of an agent, select the most appropriate action. This layer could incorporate memory mechanisms and other adaptive learning.

The action layer is responsible for executing the behavior selected in the goal and motivation layer. Exactly how the behavior will be executed depend on the individual characteristics of the agents (maximum speed, maximum angle of turn, position, etc.). In this way, different agents, though adopting the same behavior, can present different effects.

4. THE CASE STUDIES

4.1. Pedestrians Walking in an Urban Region

Generally, the studies about pedestrian movements analyze the circulation of people in urban areas and try to explain this circulation [14], [23], [6], [29]. It is known that pedestrian movement is influenced by personal motivations, the spatial configuration (streets, squares, walls, etc.), and attraction areas (movie theaters, supermarkets, shopping centers, bookstores, etc.).

The paradigms adopted for the pedestrian movements were based in the work of Helbing [6] and Feurtey [3]. They include the following: i) pedestrians try to maintain a free space around them (this includes other pedestrian and the walls, and is called the comfort area); ii) pedestrians have a resistance in adopt a direction opposed to their original planned direction (detour); iii) places, where pedestrians move in opposite directions, show the formation of queues in both directions. In our case study we added to the internal state of our agents a numeric property, whose value indicates the radius of a circle delimiting the comfort area. The center of this circle is the pedestrian position. The characteristic of formation of queues were not explicitly furnished to our model, but emerged as the result of the interaction among agents. It's important to mention that we consider that streets are exclusive for the movement of pedestrians, a situation that is very common in some urban centers.

Although it may seem complicated to make a population of agents—representing the pedestrians—reproduce these paradigms, we showed that the implementation of only two behaviors is required for the agents, namely: following one's path and avoiding collision. The behavior to stay on one's path allows the agents to move along a predefined path, while the behavior to avoid collision allows them to move in an environment with other agents and obstacles without colliding with them. These behaviors were originally proposed by Reynolds [20] and underwent minimal adaptations to be used in our case study simulation environment described by the geographic information of GIS's. The sections 4.1.1 and 4.1.2 detail these adaptations.

In general, people don't walk aimlessly. We can imagine a tuple <need, activity> defining the destination of pedestrian in an urban area, because people plan their paths in function of the places where their needs and activities can be satisfied. In our model we have agent demands, corresponding to people activities and needs, and supply points (attractions), where these demands can be satisfied (companies, shopping centers, restaurants, movie theaters, etc). We added to our agents some properties (satisfaction level, need threshold and satisfaction decay tax), in order to simulate their needs/activities. The attraction points were simulated through a well determined position in the environment and also, some additional properties (number of agents been attending, maximum capacity, time needed to satisfy agent's demands, number of agents in the waiting line, maximum waiting capacity).

In real life some attractions attract more pedestrian than others, based on product quality, trade mark, price, services, localization, etc. This is a consequence of the free competition between economic agents. Therefore, there is a differentiation among attractions. In our model, however, for simplifying reasons, we adopted attractions of the same type that differ only in the values of its properties (that is common to all). We used a criterion in that the agents prefer the nearest attraction. This is very reasonable, because here the attractions are equivalent and, in Geomarketing, we can find the gravitational model for the interaction between pedestrians and attractions. Gravitational model rules that the influence of one attraction over one agent is directly proportional to its offers and inversely proportional to the square of distance between the attraction and the agent. To avoid calculating the distance between agents and attractions all the time, we precompute an approximation of the Voronoi diagram through the position of the attractions. So, with the aid of Voronoi diagrams we can quickly know what attraction is closer to the agents. With this simplified approach is possible to create many experiments to simulate the influence of attractions in the movements of pedestrians.

The simulation environment for the urban case was defined based on the geographic information of street axes and block boundaries. The street axes consist in straight-line segments (vector representation) that correspond to the central lines of the streets, while the block boundaries consist in closed polygons (vector representation) that delimitate the circulation space for pedestrians. This information was available in Arcview shapefiles.

Finally, for using our urban prototype we must first define the simulation environment, what is accomplished first, through files defining the mesh of streets and the polygons corresponding to blocks. Then we add the attractions (setting their positions and properties). We can add a single agent or a group of agents. In this later case the agents are randomly distributed in the environment. Agents and attractions can also be removed. Finally the simulation can be started. The Figure 1 illustrates our urban simulation.

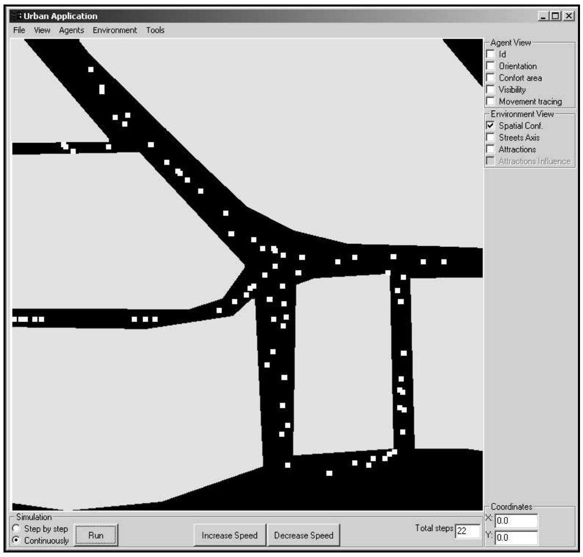

Figure 1. Prototype's interface for the urban simulation.

4.1.1. Path Follow Behavior

According to Reynolds [20], the definition of the path the agents must follow can be done by means of a polygonal representation—a sequence of connected straight-line segments. Based on this, the behavior ensures that the agents will move always close to the line that represents the path, not moving away from it more than a given predetermined distance. Initially, we implemented this behavior as described by Reynolds. To represent the possible paths in the simulation environment, the information from a mesh (network) of street axes was imported in the system. However, we verified in the simulation that the agents often collided with block boundaries in some places or failed to use the available space for circulation (presenting a tendency to move in concentrations along the street axes).

After analyzing these effects, we found out that both the collisions with the block boundaries and the tendency to move in concentrations along the street axes occurred because the *path follow* behavior originally suggested that the agents must move to the center of the street whenever there is a distance larger than a predefined value. Therefore, in narrow streets—where the width was smaller than the predefined distance—, they tended to collide with the block boundaries, while in wide streets—where the width was larger than the predefined distance—, their movement tended to be aligned in the center of the street. This deviation condition—to the center of the street—cannot be considered when geographic information is used, because the streets in a real environment hardly ever have the same width. After we removed this condition, the agents presented a more adequate behavior, that is, they made better use of the available space and did not collide.

4.1.2. Obstacle Avoidance Behavior

The original proposal by Reynolds [20] was adopted to avoid collision among agents. This proposal uses a rectangle to detect possible collisions, located in front of each agent and aligned with their orientation. This makes the collision test efficient because it ensures that only agents with collision potential, that is, those inside the detection rectangle, will be considered.

In order to prevent the treatment of the agents' collision with block boundaries implicating a pertinence test between polygons—the detection rectangle and the block boundary polygons—, we decided to apply a heuristic that generated good results**.**

The heuristic works as follows: whenever each new position of the agent is computed, a point-in-polygon test is applied to see if this new position is inside one of the polygons that mark the blocks in the simulation environment. If affirmative, the agent's new position is computed considering the possibility of collision with the detected block. If negative, the agent will move to the computed position.

4.2. Military Case Study

Simulations of battlefield behavior play a variety of important roles in modern army. Such simulations can be used to evaluate courses of action, by illustrating how opposing forces are likely to respond, to evaluate the impact of new equipment or strategies, or to plan for contingencies (e.g. "what-if" analyses). In addition, battlefield simulations are playing an increasing role in training, especially for staff officers who need to learn how to make decisions under various circumstances and to work with new digital equipment [9].

We implemented a model for military confrontations in which autonomous agents are used to simulate the tactical actions of two opposing groups of soldiers disputing strategic positions in a terrain. Although this implementation places more emphasis on the modeling of the agents' actions (determining what they could do and how to do it), the use of geographic information was required to adequately perform the computations of the mobility and visual accuracy of the agents.

The mobility computation determines the movement—time, displacement and speed—of the agents in the simulation environment, while the visual accuracy computation determines what the agents are able to see—other agents and objects—in the simulation environment. For these computations to be adequately made in the simulation, they must consider the restrictions imposed by the real environment both on the movement and on the visual accuracy of the soldiers. The sections 4.2.1 and 4.2.2 detail how geographical information was used in these computations.

Our case study is concerned about offering alternative to allow users experiment formations of defense agents against attack agents (and vice-versa), to train officers in one of the opposing forces with the other force being simulated (typically automatic defense) and to aid in the patrols´ path planning for good area coverage. We also implemented a military prototype that works similarly to the urban one. For using it, we must first define the environment—which is done by selecting the files for the DEM, trafficability and, optionally, an image. Then we must add agents and interactively define theirs paths on the terrain. Finally simulation can be started. During simulation, it is possible to inspect the properties (speed, position, events perceived, etc.) of all agents. The Figure 2 illustrates our military simulation.

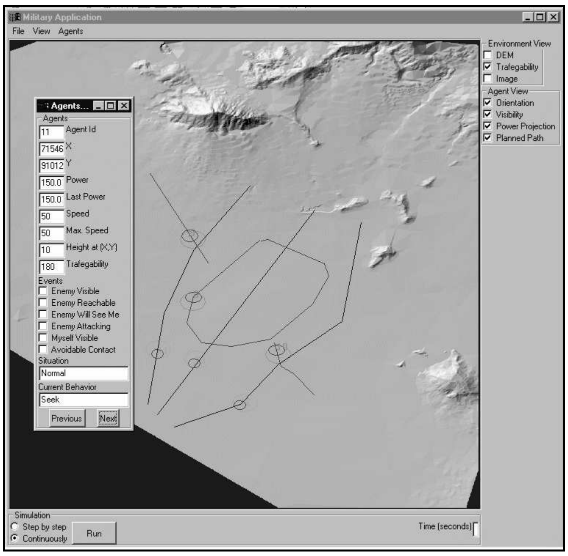

Figure 2. Prototype's interface for the military simulation.

4.2.1. Mobility

According to Lauro et al [15], in order to compute the mobility of soldiers in a simulation, a *trafficability* map of the terrain is used. The *trafficability* map is a thematic map that indicates the speed—thematic class of the map—the soldiers can reach in each position in the terrain. This map is produced in GIS's by combining various pieces of information, among them the digital elevation model (DEM), hydrography, vegetation and roads.

The *trafficability* information must be known in all points of the terrain, so a grid of points (matrix or raster) is used to represent it in GIS's. Through a partnership with the coordination team of Sistemas de Jogos Didáticos (Didactic Games System—SJD [8]), from the Brazilian Navy, we obtained a *trafficability* map in Arcview Binary Grid format to use in the simulation tests.

The agents' movement computation is made by means of the classical kinetic formulation. The agents are considered to move uniformly, that is, at constant speed (as defined by the *trafficability* map). Thus, when an agent decides to move from a point A to a point B on the terrain, the *trafficability* map informs the speed the agent must reach along the AB trajectory—due to the terrain's restrictions—and the interval, usually in time units, defined to update the agents' situation on the computer screen. Knowing the agents' speed and the time interval to update the situation on the screen, one can easily compute the next position of the agents.

The use of the *trafficability* map in the simulation avoids the need of, at every movement of the agents, determining the speed—by means of the combination of the various pieces of information already mentioned—along the planned trajectory, which would be computationally very expensive, especially considering that usually several agents are simulated at the same time.

4.2.2. Visual Accuracy

The computation of visual accuracy is the denomination adopted in military models for the perception mechanism—according to the terminology used in the field of autonomous agents. For the sake of simplification, a modeling for this mechanism consists in considering a detection area around the agents whose reach is defined by the range of the visual field and, in order to determine what they can see, one would simply check what agents/objects are inside this area.

Nevertheless, this approach is not enough, because the terrain's topography can interfere in this detection when there is an elevation or a depression in the detection area. Therefore, in order to consider such restrictions of the terrain's topography, a digital elevation model (DEM), or height map, of the terrain was used. The height map of a terrain consists of a grid of points (matrix or raster) in which each point stores the height value (elevation) of a given position in the terrain. Similarly to the *trafficability* map, the height map is also a product of GISs, usually generated from contour maps [26]; [24]. In the partnership with SJD, it was also possible to obtain a DEM in Arcview Binary Grid.

This way, the visual accuracy computation was made in two stages: a filtering stage, in which only the agents/objects inside the detection area are selected, and a classification stage, in which we verify whether there is any terrain obstruction between the agent and the other agents/objects selected in the filtering stage. To verify terrain interference we used the algorithm described by Seixas et al [25] and the information from the terrain's height map. The algorithm determines that two agents can see one another, when, along the imaginary line that connects them, there is no height in the terrain that intercepts this line.

5. CONCLUSIONS

In our two experiments (the implementation of the two prototypes) we arrived at some specific results already presented along the text. However, what we wish to emphasize here, are the aspects associated with GIS's and ABM's integration. Our work demonstrated the great usefulness of having Agent Based Models (the two prototypes) using real data about the environment, gathered from already existing GIS's data pools. Aside avoiding the economic costs involved in getting real data by our own efforts, the use of *real data* already stored in GIS's bring to the simulation process more accuracy. Another important point is that the computer code needed to a particular application should not be developed from scratch. It is very feasible the gradual building of a library of generic perceptions, actions, and communications for agents. From that library, we could select only the particular behaviors we wish to incorporate to a given model. Having a full integration among Agent Based Simulation Tools and Geographical Information Systems, a lot of code could be reused, and dozen of scientists, not necessarily computer experts, would get access to powerful simulation tools. We think that, in a couple of years, this kind of integration will be very common. What is need is to tackle the interoperability problems, build libraries of open source code and disseminate similar experiences.

6. ACKNOWLEDGMENTS

The authors would like to thank: i) the Marines Corps of the Brazilian Navy for the information provided for the military case study; ii) Professor Bin Jiang, from the Gävle University in Sweden, for the valuable help in modeling the urban case study; iii) Marcelo Tílio Monteiro de Carvalho, from TecGraf/PUC-Rio, for the geographic information provided for the urban case study.

7. REFERENCES

[1] Anderson, J. Providing a Broad Spectrum of Agents in Spatially Explicit Simulation Models: The Gensim Approach. In *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes,* Edited by H. Randy Gimblett, Oxford University Press, 2002, 21-58.Ding, W., and Marchionini, G. *A Study on Video Browsing Strategies.* Technical Report UMIACS-TR-97-40, University of Maryland, College Park, MD, 1997.

[2] Box, P. Spatial Units as Agents: Making the Landscape an Equal Player in Agent-Based Simulations, In *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes,* Edited by H. Randy Gimblett, Oxford University Press, 2002, 59-82.

[3] Feurtey, F. *Simulating the Collision Avoidance Behavior of Pedestrians.* Master Thesis, 2000.

[4] Franklin, S., and Graesser, A. Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In *Intelligents Agents III: Agent Theories, Architectures, and Languages*, edited byMuller, J. P. , Wooldridge, M. J., and JenningsN. R. . Berlin: Springer, 1997, 21-35.

[5] Gimblett, H. R. *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes,* Edited by H. Randy Gimblett, Oxford University Press, 2002.

[6] Helbing, D., Farkas, I.J. ,and Molnár, P. Simulation of Pedestrian Crowds in Normal and Evacuation Situations. In *Pedestrian and Evacuation Dynamics*, Springer-Verlag, Berlin, 2002, 21-58.

[7] Ingham, J. What is an Agent ? Technical Report #6/99, Centre for Software Maintenance, University of Durhan, Durhan, London.

[8] Instituto de Matemática Pura e Aplicada. *Webpage of Sistema de Jogos Didáticos,* available at: <http://www.impa.br/~tron/sjd>, accessed in: Aug 13, 2004.

[9] Ioerger, T. R., Volz, R. A., and Yen, J.; Modeling Cooperative, Reactive Behaviors on the Battlefield with Intelligent Agents. In *Proceedings of the nineth Conference on Computer Generated Forces (9th CGF)*, Orlando, May, 2000, 13-23.

[10] Itami, R. B.. Mobile Agents with Spatial Intelligence, In *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes,* Edited by H. Randy Gimblett, Oxford University Press, 2002, 191-210.

[11] Itami, B.,. Gimblett, R , and Raulings R. RBSIM: Using GIS-Agent Simulations of Recreation Behavior to Evaluate Management Scenarios, In *Proceedings of AURISA 99 Conference,* 1999.

[12] Jiang, B. , Gimblett, H. R. An Agent Based Approach to Environmental and Urban Systems within Geographic Information Systems, In *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes,* Edited by H. Randy Gimblett, Oxford University Press, 2002, 171-189.

[13] Jiang, B. Agent-based Approach to Modeling Environmental and Urban Systems within GIS, In *Proceedings of 9th International Symposium on Spatial Data Handling*, Beijing, 2000.

[14] Jiang, B. SimPed: Simulating Pedestrian Flows in a Virtual Urban Environment. In *Journal f Geographic Informationand Decision Analysis,* v. 3, n.1, 1999, 21-30.

[15] Lauro, A., Seixas, R.B., and Torres, L., Caminho Mínimo em Matrizes de Custo Dinâmico. In *IV Simpósio de Pesquisa Operacional e V Simpósio de Logística da Marinha – SPOLM 2001*, Rio de Janeiro, December, 2001.

[16] Lyrio, G.H.S.O., and Seixas, R. B. Riverine Operations: Modeling and Simulating, In *Proceedings of Advanced Simulation Technologies Conference - ASTC, in Military, Government, and Aerospace Simulation Symposium* , 2004.

[17] Lyrio, G.H.S.O. , Andrade, D. V., Seixas, R. B. Modelagem de Ambientes Ribeirinhos para Jogos Militares, In *Proceedings of Brazilian Workshop on Games and Digital Entertainment - WJOGOS*, 2003.

[18] Minar, N., R. Burkhard, C. Langton, and M. Askenazi. The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations, 1996, *Working paper 96-06-042,* Santa Fe Institute, Santa Fe, NM, available at: http://www.santafe.edu/projects/swarm/overview.ps

[19] Reese, B., and Stout, B. Finding a Pathfinder. In *Proceedings of AAAI99 Spring Symposium on Artificial Intelligence and Computer Games,* 1999, 69-72.

[20] Reynolds, C. Steering Behaviors for Autonomous Characters, *Proceedings of Game Developers Conference*, 1999, 763-782.

[21] Reynolds, C. Flocks, Herds, and Schools: A Distributed Behavioral Model. In *Proceedings of SIGGRAPH'87*, 1987, 25-34.

[22] Rodrigues, A. , Grueau, C. , and Raper, J**.** Environmental planning using spatial agents, In *Innovations in GIS* 5 ,1998, 108—118.

[23] Schelhorn, T., and O´Sullivan, D., STREETS: An agent-based pedestrian model, In *Computers in Urban Planning and Urban Management*, Venice, September, 1999.

[24] Seixas, R.B., Figueiredo, L.H., and Silva, C.A. Interpolação de Curvas de Nível por Difusão de Calor, In *Proceedings of the III Geoinfo*, 2001.

[25] Seixas, R.B., Mediano, M.R., Gatass, M. Efficient Line-of-sight Algorithms for Rea Terrain Data. In *III Simpósio de Pesquisa Operacional e IV Simpósio de Logística da Marinha – SPOLM 1999,* Rio de Janeiro, December, 1999.

[26] Silva, C.A. , Seixas, R.B., and Figueiredo, L.H. Uma Metodologia para Geração de Modelos de Elevação a partir de Curvas de Nível. In Proceedings of the II Geoinfo, 2000, 82-87.

[27] Silva, C. A. *Modelagem Comportamental para Agentes Autônomos em Ambientes Reais*, Master Thesis, University of the State of Rio de Janeiro, Brazil, 2003.

[28] *Starlogo on the Web*, available at: <http://education.mit.edu/starlogo/>, accessed in: May, 31, 2005.

[29] Therakomen, P., *Mouse.Class – The Experiment for Exploring Dynamic Behaviors in Urban Places*. Master Thesis, University of Washington, Washington, USA, 2001.

[30] Wooldridge, M. J., Jennings, N. R. Agents Theories, Architectures, and Languages: A Survey. In *Proceedings of ECAI'94 Workshop on Agent Theories, Architectures and Languages*, 1995, 1-22.