Using BDI Agents to Simulate Evacuation Procedures

Vágner de Oliveira Gabriel[1], Carlos Eduardo Pereira de Quadros[1], Diana Francisca Adamatti [1], and Cleo Zanella Billa [1]

[1]Center of Computational Science - Federal University of Rio Grande - FURG
Av. Itália, Km 8 - Campus Carreiros CEP 96203-900 - Rio Grande/RS. Brasil.

vdeoliveiragabriel@gmail.com,

{carlos.quadros, dianaadamatti, cleobilla}@furg.br

Abstract. *In closed places is essential the existence of emergency doors, because if an accident happens then people need to leave quickly. The existence of an emergency exit is as important as its location, because it has to be located on an easy access point. Computer simulations are very useful tools to emulate incidents and the behavior of people in emergency situations. In addition, cognitive multiagents systems aim to mimic human behavior. This paper proposes the validation of a cognitive multiagent based tool to simulate evacuation through a proof of concept, that two emergency doors are better than one.*

1. Introduction

As global population increases, urban centers are increasingly inhabited, and closed places need special evacuation plans to avoid major disasters. In case of accidents, evacuation time should be as fast as possible [Pereira et al. 2011].

Computers can be very useful tools to simulate accidents and how to minimize their consequences. Evacuation simulation can analyze different factors, such as: number of emergency exits, location of emergency exits, size of emergency exits, etc.

In a computer simulation, we can have an overview of the location, allowing the execution of several tests, with different risk situations, with low cost and in short time. Computer simulation can be used to develop better strategies to public administration and improve public security [Zampronio et al. 2015].

Multiagent Systems (MAS) are very useful tools in the simulation field. MAS can be cognitive or reactive. Reactive agents behave like animals, that respond to a situation spontaneously, and cognitive agents behave like humans, reacting to situations in a planned way [Shendarkar et al. 2008]. Simulation through cognitive agents can be closer to real environments when human behavior is desired, because they aim to simulate logical reasoning and emotions.

This paper presents a tool to simulate people evacuation of closed places. Our tool was built using a multiagent approach. The agents were modeled according to the BDI architecture, where BDI means Beliefs, Desires and Intentions. To validate our tool, we proved the concept that two emergency doors are better than one. Using two different scenarios, our experiments show that, despite the room layout, two doors are better than one.

This paper is organized as follows. Section 2 introduces the multiagent area, and has one subsection about BDI architecture. Section 3 describes the proposed tool and the

materials we used to develop it. Section 4 presents our case study to validate our tool. In section 5, the related work is discussed. Finally, section 6 shows our conclusions.

2. Multiagent Systems

Agents can be defined as computational characters that act according to a script, directly or indirectly defined by a user [Rezende 2003]. These agents are located in an environment and are capable of autonomous actions, intending to achieve their goals [Wooldridge and Jennings 1994]. Agents can act alone or work in communities forming Multiagent Systems (MAS).

The comprehension of agent is vital to understand MAS. According to [Wooldridge and Jennings 1994], intelligent agents have the following properties:

- Autonomy: agents execute without a direct connection with humans or other agents, they have total control of their actions.
- Social abilities: agents should be able to communicate with other agents, including humans.
- Reactivity: agents should perceive changes in their environment and react, in a acceptable time, to these changes.
- Proactivity: agents should pursued their goals.

MAS can be defined as a group of autonomous agents collaborating with each other, in order to solve a problem [Amandi 1997]. MAS is a subarea of distributed artificial intelligence, that studies the behavior of set of autonomous agents. These autonomous agents have different characteristics, socialize in the same environment, interact, cooperate, exchange information and avoid conflicts in order to achieve objectives [Wooldridge 2009].

2.1. BDI Architecture

In 1987, Bratman [Bratman 1987] proposes the BDI model as philosophic theory about practical reasoning, where human behavior is model as three concepts: beliefs, representing the knowledge that the agent has about the environment, and the agents on the environment, including itself; desires, representing the goals or states that the agent wishes to achieve; and intentions, representing the action plans that the agent follows to achieve its desires.

The BDI model is based on the idea that actions come from the process of practical reasoning, and it is compound by two steps: The first step is called deliberation, where a set of desires are chosen; and the second step defines how the selected desires can be achieved.

In 1995, [Rao et al. 1995] proposed a BDI architecture to software agents, introducing a formal theory and a BDI interpreter. Figure 1 shows the BDI architecture.

In figure 1, the `Belief Revision Function` receives information from the sensors, and along with the current beliefs, updates the beliefs. The `Option Generation Function` checks alternative states that can be achieved by the agent in order to achieve its goals. This procedure is conducted based on the current state of the world, or in other words, the information available in the beliefs and intentions of the

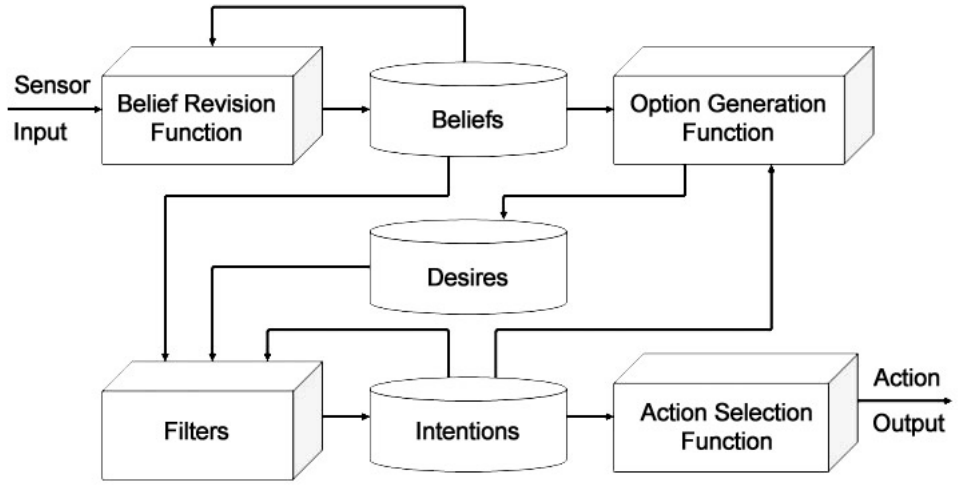

Figure 1. BDI Architecture Adapted from [Hübner et al. 2004]

agent. After the goals update, the agent has to decide which actions it must execute. However, the agent has already committed with other actions, so new actions can not conflict with previous actions. The `Filter` function updates the intentions checking new beliefs, new goals, and previous intentions. At the end, the `Action Selection Function` selects which action should be executed.

3. Materials and Methods

This section describes the tools we used in this work, and how we modeled the environment and the agents using a BDI approach.

3.1. AgentSpeak(L)

AgentSpeak(L) is an agent oriented programming language with support to events and actions. It was presented in [Rao 1996] and was inspired in the BDI architecture [Rao et al. 1995]. In the AgentSpeak(L), beliefs, desires and intentions are not explicitly declared but they are supported through the language features.

The knowledge that an agent has about himself, the environment, and the other agents can be seen as beliefs. The state that an agent intends to achieve can be seen as desires, and the choice of the plans to be followed, based on internal beliefs and external stimuli, can be seen as intentions.

- Beliefs Base: as the name says, it is the set of initial beliefs.
- Goal: it is the state that the agent wants to achieve.
- Plan Library: it is a set of plans that the agent has. Each plan is a sequence of actions to achieve a goal.

3.2. Jason Platform

The simulator presented in this paper was developed in the Jason Platform. Jason is a tool that integrates Java and AgentSpeak(L) in order to facilitated BDI-based MAS [Bordini et al. 2007].

Jason has the following features [Hübner et al. 2004]:

- Fail handlers in plans;
- Communication is based on talking actions, including source information as belief;
- Identification of plans. This can be used to develop specific function to plan selection;
- Support for environment development, in Java;
- Execution of MAS in a distributed environment, such as a network;
- Use Java to personalize the functions of plan selection, belief revision, or other agent specialty;
- It has a internal actions library, and it can be extended.

3.3. Environment Model

To perform the simulations, we create an environment using the Jason Platform, where it is possible to show how an evacuation would occur in case of an emergency situation on a closed environment. This environment is represented by a matrix 25×25 and it has walls, exit doors and a safe place for the agents.

3.4. Agents Model

The agents were modeled according to the BDI architecture:

- Beliefs: Agents know where they are, where is a safe place, the location of other agents, and the path that was already been traveled.
- Desires: The agents want to achieve a safe place.
- Intentions: The agents have plans to move and others that helps them to achieve a safe place.

4. Case Study

This paper presents a case study where we simulate evacuation plans in two different rooms. For each room, we show two scenarios, one without a emergency exit and another with an emergency exit. So, we show the simulation in 4 different scenarios: room 1 with and without emergency exit, and room 2 with and without emergency exit. We tested each scenario with different number of agents: 20, 50 and 100. For each configuration we execute the simulator 10 times. Table 1 resumes our configuration parameters.

4.1. Agents Model

We model the agents using BDI architecture, defining beliefs about the environment, the desire to evacuate the room, and intentions to move around. Below we described the model implementation in details. The code is in AgentSpeak(L).

4.1.1. Beliefs

Each agents knows its initial position, and the location of an exit door. Agents also knows how to identify walls (figure 2), and they have memory. In other words, they record where they already have been, so at each step the beliefs of the agents are updated, recording where it has passed.

Table 1. Case Study Scenarios

Scenario	Emergency Exit	Number of Agents	Number of Repetitions
Room 1	No	20	10
Room 1	No	50	10
Room 1	No	100	10
Room 1	Yes	20	10
Room 1	Yes	50	10
Room 1	Yes	100	10
Room 2	No	20	10
Room 2	No	50	10
Room 2	No	100	10
Room 2	Yes	20	10
Room 2	Yes	50	10
Room 2	Yes	100	10

```
01    initialPosition(10,10).
02    exitDoor(5,10).
03    wall(3,4).
04    wall(3,5).
...   ...
...   wall(3,15).
```

Figure 2. Agents Beliefs

4.1.2. Desires

Agents have the desire to reach the exit door (figure 2: line 2). AgentSpeak(L) does not have an explicit way to specify desires, they are implicit defined inside the plans.

4.1.3. Intentions/Plans

Agents have plans to achieve its goals, in this case, reach the exit door.

The first lines defines the plan `start`. The agent first checks if its position is the same of exit door. If it is true, the agent stops. If not, the agent executes the plan `walk` (figure 3: lines 01-4).

In line 06 of figure 3, there is a plan `+Occupiedposition(D,C)[source(R)]`, which receives the position of others agents and it add to the beliefs base.

The walking plan decides the next step and checks if it is not a wall and if it did not passed before (not in memory). If it is true, it walks (change position) and add the new position to the memory of the agent (figure 3: lines 06-12).

The agent repeats all steps until it reaches an exit door.

```
01  +!start : true
02    <- ?position(X,Y);
03       ?exitDoor(A,B);
04       !walk(X,Y,A,B).
05
06  +Occupiedposition(D,C)[source(R)].
07
08  +!walk(X,Y,A,B): X>A & Y>B
                       & not wall(X-1,Y-1)
                       & not memory(X-1,Y-1)
                       & not Occupiedposition(X-1,Y-1)
09    <- P1 = X-1;
10       P2 = Y-1;
11       -position(X,Y);
12       +position(P1,P2);
13       +memory(P1,P2);
14       .send(agent2, tell, Occupiedposition(P1,P2));
15       .print(P1,P2);
16       - Occupiedposition.
17
18        !start.
```

Figure 3. Agents Plans

To avoid that two agents occupy the same place, we used message exchange. Each agent send a message to all other agents informing its position, so when a agent tries to move, it verifies if there is not another agent in its desired destination.

4.2. Results

In this sub section, we present the results obtained in the simulations carried out as a case study of this work.

4.2.1. Room 1

Room 1 is presented in figure 4. It is a room without internal walls and has only one door. In this scenario we perform 30 simulations, 10 with 20 agents, 10 with 50 agents and 10 with 100 agents. Then, we add a second door (emergency exit) on the room, see figure 5, and repeat the tests described above. Table 2 shows the results of the simulations. The last column shows the standard deviation of the evacuation time of all agents.

Table 2. Room 1 - Simulation Results

Scenario	Number of Agents	Evacuation Time Without Emergency Exit (s)	Standard Deviation	Evacuation Time With Emergency Exit (s)	Standard Deviation
Room 1	20	23.0	1.268	18.6	1.280
Room 1	50	40.3	0.921	30.6	1.624
Room 1	100	58.2	1.019	49.3	2.002

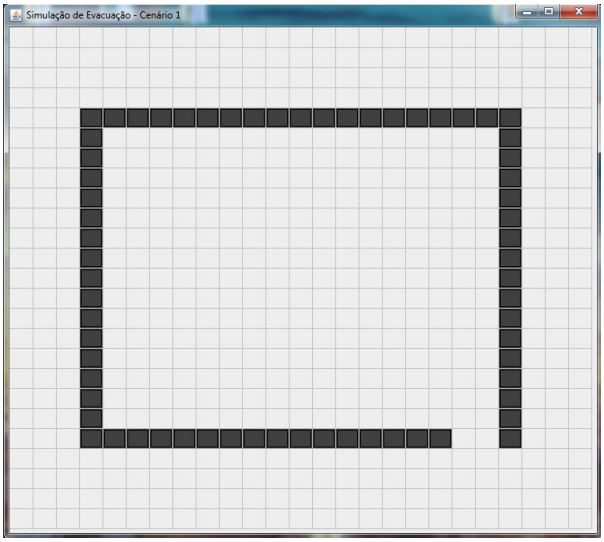

Figure 4. Room 1 Without Emergency Exit

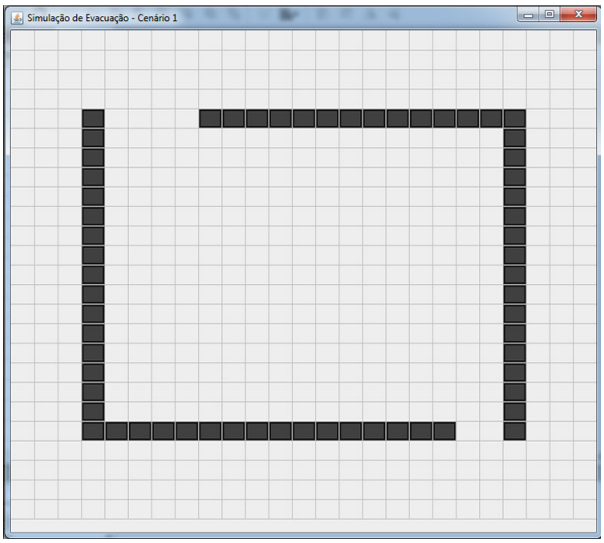

Figure 5. Room 1 With Emergency Exit

4.2.2. Room 2

In room 2, figures 6 and 7, we add an internal wall in the environment. As the previous experiment, we test two scenarios. One without emergency exit and the other with an emergency exit. For both cases, we simulate 10 times the evacuation plan using 20, 50 and 100 agents.Table 3 resumes the results. The last column shows the standard deviation of the evacuation time of all agents.

Table 3. Room 2 - Simulation Results

Scenario	Number of Agents	Evacuation Time Without Emergency Exit (s)	Standard Deviation	Evacuation Time With Emergency Exit (s)	Standard Deviation
Room 2	20	39.6	0.916	35.8	0.917
Room 2	50	72.7	0.871	68.5	0.663
Room 2	100	105.5	1.673	99.9	0.943

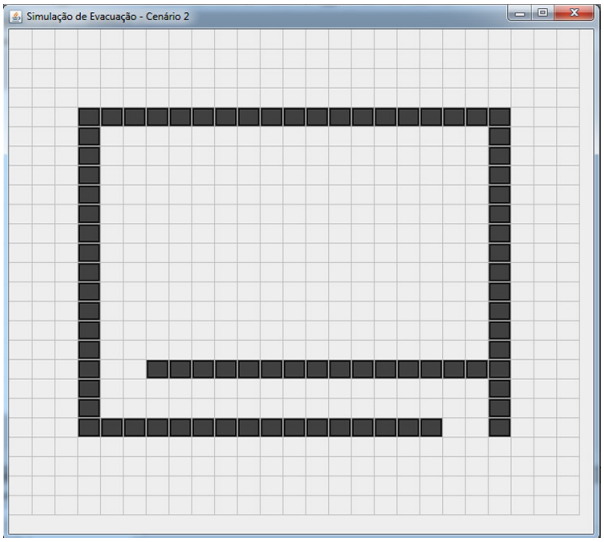

Figure 6. Room 2 Without Emergency Room

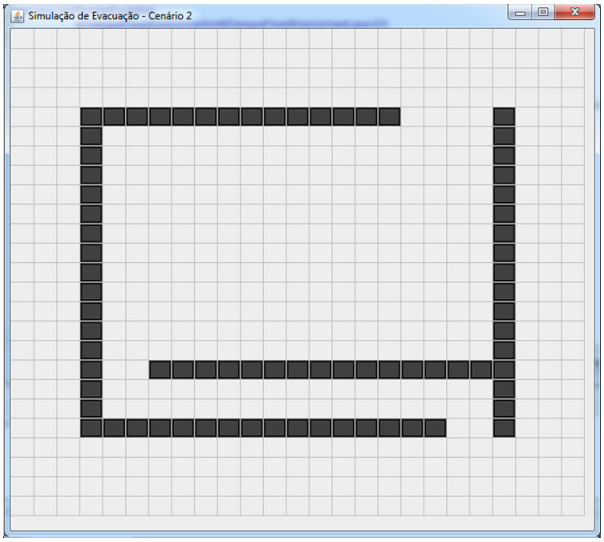

Figure 7. Room 2 With Emergency Room

In figure 8, we can observe that the use of an emergency exit port in the simulations reduced the time of agents evacuation. However, the evacuation time (in seconds) was not decreased in the same proportion (in half of the time), because the agents position and their movement in the environment are also relevant factors to define their exit.

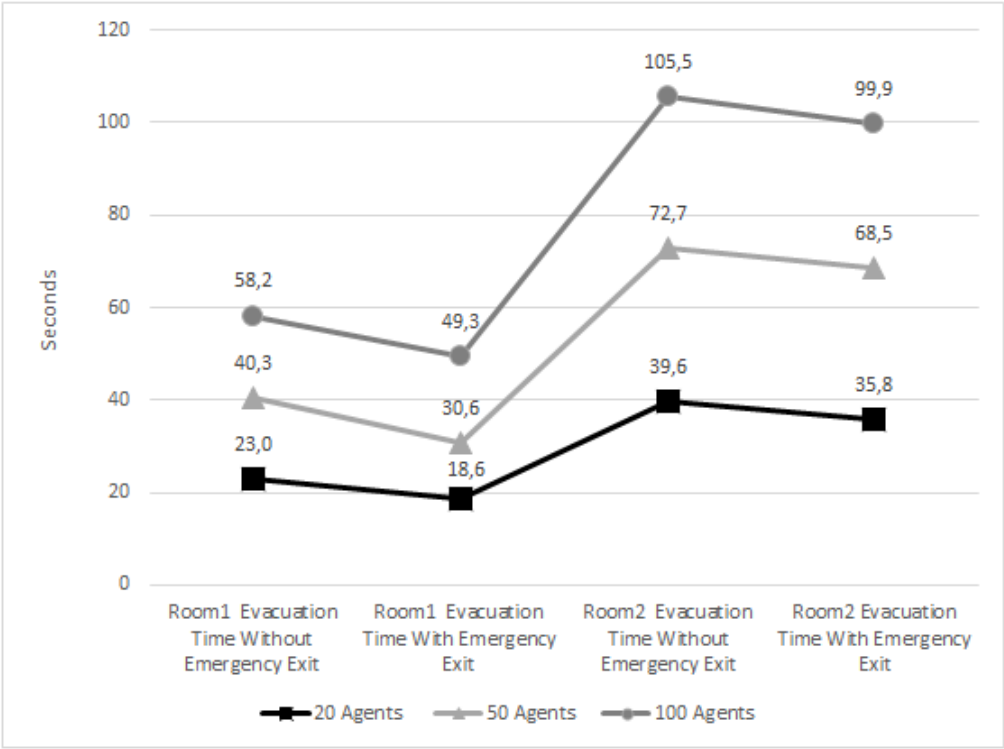

Figure 8. Results of simulations of each scenario with the evacuation time

Visualizing our simulation, we could note that all agents founded an exit door, none of them got stuck in the room. As expected the lowest evacuation time was in the room without internal walls and two exits. Actually, trough our simulation it is possible to

see that an extra door, may considerably reduce the evacuation time of a room. Although this was totally expected, showing these results, and the real gain of an extra door, we consider as an important contribution of this work.

The possibilities of increasing the model are enormous. Ones can be easily done by changing door locations, since its position can directly influence the evacuation time. More complex features can be add, especially regarding agents behavior. For example, agents can have different profiles, different emotions, different locomotion speeds, etc. Agents communication can also be an interesting feature to analyze. So the possibilities of future work are huge.

5. Related works

It is possible to find a considerable number of papers that discusses evacuation simulation or crowd control, each one using different techniques and/or tools.

The CrowdSim [Cassol et al. 2012] tool was developed to simulate crowd situations, in any previously mapped environment. This tool represents the environment in 3 dimensions, and the user defines people movement area. It is also possible to define a goal region, a place where people should go or stay during a simulation.

The work proposed in [Zampronio et al. 2015] presents a simulation system of 3D oil platforms, with results in real time. This system uses the Unity Game Engine as architecture. Unity was chosen because the authors wanted to focus in the logic and business rules of the simulation.

In [Pereira et al. 2011], the authors presents a model that has as main goal to facilitate the comprehension about the problem of crowd evacuation using simulation. The model to the crowd evacuation simulation is based on cell automata.

In [Braga 2006], the authors proposes a simulation of an escape route using augmented reality and multiagent. This simulation was created using the Blender application. Blender is a graphic tool that allowed the creation of three dimensional elements.

[Hamagami and Hirata 2003] proposes a crowd simulation using a model with two layers: a multiagent and cell automata. In this work, the crowd behavior emerges from the autonomous actions of the agents, and it distinguishes the autonomous action process from a physical interference constraint.

[Pan et al. 2007] propose a multiagent structure aiming to simulate social human behavior in an emergency evacuation. They build a prototype that it is able to simulate some behaviors, such as, competitive, queuing and grazing.

The work of [Shendarkar et al. 2008] is presented a proposal of crowd simulation under terrorist attack in public areas. The methodology includes using BDI based agent in an environment of virtual reality. The authors constructed the VR model to simulate the emergency scenario with CAVELib, which is a library of functions built for the CAVE system and used OpenGL Performer libraries to deploy the graphics. To program the hardware system graphics, the authors used visual C ++.

All previous cited work are good alternatives to simulate evacuation procedures, however, most of them focus on the graphical part. In this work, we propose to use BDI based agents because we understand that this simulation should be made based on the

mental attitudes of people, and the BDI architecture tries to simulate human behavior. As mentioned before, the work of [Shendarkar et al. 2008] uses BDI to model human behavior in a crowd simulation, however our work uses the Jason Platform as the simulation environment.

6. Conclusions

This work presented a simulator of evacuation plans using cognitive agents, or agents that act according to their beliefs and goals, to decide their actions. The potential of cognitive agents in this kind of simulation are enormous because they intend to represent human behavior.

As expected, the results of the simulation show that two exit doors are better than one, and that rooms without divisions are better than with divisions. Again, our experiments intend to validate our tool. The future work is to explore different scenarios and evacuation plans.

Another important contribution of this work is use a multiagent approach to model the simulation, especially a BDI-based MAS. As future work, the simulator has the ability to represent each person individually, with different features, add different obstacles, represent different scenarios. The possibilities of enhancing the model are enormous.

References

Amandi, A. (1997). Programação de agentes orientada a objetos. Tese de doutorado. Universidade Federal do Rio Grande do Sul.

Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons.

Braga, L. A. F. (2006). *Simulação de Rota de Fuga e Sinalização Utilizando Multi-Agentes e Realidade Virtual*. PhD thesis, D. Sc. thesis, Universidade Federal do Rio de Janeiro.

Bratman, M. (1987). Intention, plans, and practical reason.

Cassol, V. J., Rodrigues, R. A., Carneiro, L. C. C., Silva, A., and Musse, S. R. (2012). Crowdsim: Uma ferramenta desenvolvida para simulação de multidões. In *I Workshop de Simulação Militar-SBGames2012*.

Hamagami, T. and Hirata, H. (2003). Method of crowd simulation by using multiagent on cellular automata. In *Intelligent Agent Technology, 2003. IAT 2003. IEEE/WIC International Conference on*, pages 46–52. IEEE.

Hübner, J. F., Bordini, R. H., and Vieira, R. (2004). Introdução ao desenvolvimento de sistemas multiagentes com jason. *XII Escola de Informática da SBC*, 2:51–89.

Pan, X., Han, C. S., Dauber, K., and Law, K. H. (2007). A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *Ai & Society*, 22(2):113–132.

Pereira, L. A., Duczmal, L. H., and Cruz, F. R. (2011). Simulaçao de evacuaçao emergencial via autômatos celulares: Uma proposta de modificaçao do modelo de schadschneider. In *XXXII Congresso Nacional de Matemática Aplicada e Computacional*, pages 692–698.

Rao, A. S. (1996). Agentspeak (l): Bdi agents speak out in a logical computable language. In *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 42–55. Springer.

Rao, A. S., Georgeff, M. P., et al. (1995). Bdi agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319.

Rezende, S. O. (2003). Sistemas inteligentes: fundamentos e aplicações.

Shendarkar, A., Vasudevan, K., Lee, S., and Son, Y.-J. (2008). Crowd simulation for emergency response using bdi agents based on immersive virtual reality. *Simulation Modelling Practice and Theory*, 16(9):1415–1429.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John Wiley & Sons.

Wooldridge, M. and Jennings, N. R. (1994). Agent theories, architectures, and languages: a survey. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 1–39. Springer.

Zampronio, G. B., Raposo, A. B., and Gattass, M. (2015). A 3d simulation system for emergency evacuation in offshore platforms. In *Virtual and Augmented Reality (SVR), 2015 XVII Symposium on*, pages 99–106. IEEE.