# A Model for Fire Spreading by Multi-Agent Systems: A RoboCup Rescue Simulation and Swarm Platform Approach

Terry L. Ruas, Maria das Graças B. Marietto, Robson dos S. França, André Filipe de M. Batista
Centre of Mathematics, Computation and Cognition
Federal University of ABC
Campus Universitário Santo André
CEP. 09.210-170 - Brazil
{terry.ruas, andre.batista, maria.marietto, robson.frança}@ufabc.edu.br

***ABSTRACT:*** *This paper presents an analysis of the RoboCup Rescue Environment tools for simulations based on the fire-spread phenomenon. Its main purpose is highlighting the features of two simulation tools: the Agent Simulation and the Swarm platform, simulating a fire spread in a particular city. Two simulations had been run using the Agent Simulation unit: one with a model that gives a fire spread that cannot be controlled by the agents and a second one using a model which the agents can control that fire spread, providing an environment and a framework for future studies about the connection between fire spread and collective behavior. Also, this paper shows a theoretical analysis of the Swarm platform and it suggests integration among the tools presented herein.*

**Keywords:** Fire, Spread, RoboCup, Rescue Environment Tools, Multiagent, Simulation

**Received**: 11 May 2011, Revised 14 July 2011, Accepted 18 July 2011



## 1. Introduction

The area of Multiagent-based simulations is formed by the intersection of the Distributed artificial Intelligence (DAI) and the Computational Simulation areas. This field of research provides a proper infrastructure for modeling and understanding the processes related to social interactions such as coordination, cooperation, training and coalition of groups, resolutions of conflicts, among others. Multiagent-based simulation models are based on the concept of the individual-program relationship allows the simulation of an artificial world formed by interactive computing entities, which each agent represents an entity of the target system, or a group of them. Since the infrastructure of technological and theoretical areas of Simulation allows grabbing the essential elements of a target system without working directly with it, it is reasonable to deal with a phenomenon such as the spread of fire without putting the integrity of living beings and the environment at risk [4]. Specifically, the fire spreading phenomenon is fully related to the panic in crowds' phenomenon, which is a kind of collective behavior that usually involves a group of people exposed to a hazardous situation, and there is always a risk and an imminent sense of urgency to act in part of the individuals. This phenomenon may be triggered by various factors such as natural hazards (floods, earthquakes) and threats caused by humans, such as terrorist attacks.

Researches about panic in crowds are important because they assist in obtaining a clear and detailed understanding of the social theories, and they can be used as basis for the development of algorithms that can show new ways of computing solutions. Also, the results of these studies may help in the definition of measures to predict the emergence of collective panic and/or actions that may point to minimize material losses and especially human losses. However, even with the importance of studies on the phenomenon panic in crowds, there are a limited number of simulations that address this type of behavior [2]. In order to aid in filling this blank, this paper presents a multi-agent environment and tools for future simulations about the panic in crowds. More specifically it analyzes the integration between the Robocup Rescue's Fire Spread Simulator (FSS) [1, 13] and the Swarm multi-agent simulation platform [6].

The RoboCup Rescue project aims to simulate disaster situations by the usage of digital resources. The project promotes a study on physical robots and virtual environments for simulated searching and rescuing, coordination of teamwork and strategies in an interconnected way. Swarm is a multiagent simulation software platform for the study of complex adaptive systems. In the Swarm system the basic unit of simulation is the "swarm", a collection of agents executing a schedule of events. Swarm accommodates multilevel modeling approaches in which agents can be composed of swarms of other agents in nested hierarchies.

This paper studies the RoboCup Rescue environment in a practical sense; it also studies the Swarm platform, but through a theoretical approach.

The integration between the FSS - also known as ResQFire simulator - and the Swarm platform is important because it might lead to a model and a representation which are closer to real situations. The remaining of this paper is structured as follows. Section 2 presents the main elements of the Robocup Rescue project, specially the Agent Simulation Unit. In Section 3, there is an overview of the Swarm platform and the model's architecture. Section 4 presents two scenarios about fire situations and a discussion about the simulations' results. Finally, Section 5 presents this paper's conclusions and some suggestions for future works.

## 2. RoboCup Rescue Leagues

The RoboCup Rescue project has three kinds of search and rescue (S&R) simulations: (i) S&R in virtual environments with multiple agents. Each agent has its own distinct features (Agent Simulation Unit); (ii) S&R using physical robots to rescue victims (Rescue Robots League); (iii) S&R robots using virtual robots (Virtual Robots Unit). This paper is focused on the Agent Simulation unit; the hierarchical structure that composes the RoboCup Rescue project is displayed in Figure 1. A detailed view of the other units can be found in [1].

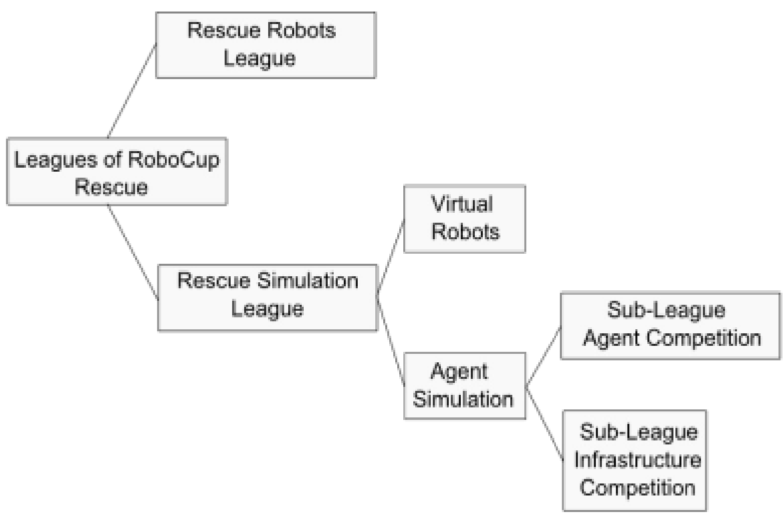



Figure 1. RoboCup Rescue Leagues

The Agent Simulation Unit has two goals. The first one is the development of simulators to prepare the framework for systems' simulation and to emulate the prevailing phenomena found in disasters. The second goal is to design multiple intelligent agents which are able to interact with each other in situations of disaster. Thus, the Agent Simulation Unit can be divided in two competition sub-leagues: Infrastructure Competition and Agent Competition.

The sub-league Agent Competition deals with algorithms for coordination and competition of agents in the different maps of the Rescue Simulation League platform. In this case, the challenge towards to the development of algorithms which allow the coordination of the agent teams Ambulance, Police Force and Fire Brigade in order to save as many civilians as possible and to extinguish fire outbreaks in the city where an accident just occurred. The sub-league Infrastructure Competition involves the evaluation of tools and simulators to simulate issues in the disaster management [11].

### 2.1 Architecture of Agent Simulation Unit

The design of the Agent Simulation Unit starts with the definition of its architecture, their blocks (or modules) and the main structural interaction among these blocks. Initially, the proposed architecture takes the following main blocks into consideration [8]:

- A number of agents with different capabilities called Agents;
- A set of simulators for specific aspects of the field called SimAED;
- A process that connects the Agents and SimAED called Kernel;
- A process for maintenance of geographic information called GIS (Geographic Information System);
- A predefined set of messages for inter-agent communication and perception of each agent on the state of the world.

In each cycle of the simulation, the kernel sends to each agent all the information that is perceived by the agents' auditory and visual sensors. The visual information includes the value of the properties of all agents and objects in the geographical area, which are in the range of sight of the agent. The radius of sight is a parameter of the simulation, usually defined as 10 meters. The auditory information contains all messages received by telecommunications channels, or only through natural voice [8].

The kernel process also generates the evolution of simulation time. According to its goal of realism, the RoboCup Rescue intends to simulate the five hours following the occurrence of an earthquake. The simulation considers a discrete time evolution, which the execution of each cycle represents a time lapse of one minute in real time. Thus, the standard simulation takes 300 cycles and the kernel imposes the restrictions of real time to all other processes (agents and simulators) [8].

In the architecture of the Agent Simulation unit, the GIS process ensures the maintenance of the properties values of each geographical and road building. For those, the GIS editor is used, which is a graphical tool to edit the interactive GIS files. Enabling operations, such as the creation of maps, adding and removing objects, checking inconsistencies, adjusting officers' positions, among others are made without any sort of high-level programming [9].

### 2.2 Components of the Agent Simulation's Architecture

In addition to the blocks described in Section 2.1 the Agent Simulation Unit can also be divided into smaller sub blocks. These sub-blocks are made of roads, buildings, Agents (Civilian and Search and Rescue, shown in Figure 2) and simulators (traffic, blocking/collapse, fire, visualization, among others). In order to have a better understanding of the performed simulations, a brief explanation of these components is required.

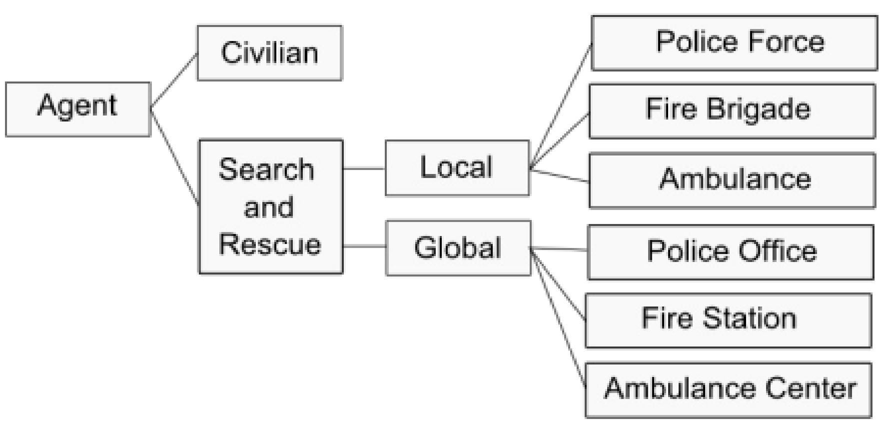



Figure 2. RoboCup Rescue Agents

The roads have properties such as the geographical coordinates of each end, length, width, number of lanes for each direction, width of each track and the type of street (viaduct, bridge, tunnel or highway emergency). Also, there are two properties related to the streets' blockage situation: one situation is caused by debris or cracks; the other situation is related to the number of cycles needed to clear the street (task carried out by Police Force agents) [8].

The buildings have properties such as the geographical location, the number of entries (that connects the building to a street), the number of floors, the area of deployment in the field, total area of construction (sum of the floors' areas), building materials (wood, steel and cement). Each material has a fire transmission rate and intensity level. Finally, there is a special building called refuge which is immune to fire and it is the place where the local agents are safe from disasters [8].

The local agents are able to act directly on the environment and they can be any one of the following: Police Force, Fire Brigade and Ambulance. Any local agent can communicate and move on the simulation's geographic area and it has its specific skills. The Civilian agent (individual or family) is a victim of the disaster, which might need help (by the Ambulance agent). The Police Force agent is able to restore traffic conditions on roads, removing blockages caused by debris or cracks. The Fire Brigade agent can put out fire in buildings and handle the hose to set up the output angle and the amount of water. The Ambulance agent is able to rescue the local agents (including other ambulances). Only the Ambulance agent can help directly any local agent.

A global agent represents an organization of local agents. These organizations are the Police Office, the Fire Station and the Ambulance Center. The non-humanoid agents are represented by constructions and they do not interact directly to the environment; their actions are performed by the local agents that compose them. The Police Office allocates resources to assign tasks to staff members and collects all the information coming from the Police Force agent that explores the environment. The Fire Station collects and integrates all the information sent by the Fire Brigade agents and allocates it according to a simple policy as the Police Force does. Likewise, the Ambulance Center does the same as the Fire Station. However it collects the information sent by Ambulance Center agents.

According to [8], the main role of the global agent is to prepare a disaster overview that aids in the supporting and coordinating activities, which are developed on the ground for each individual (local agent). Building a global perspective by composition of different perspectives is supported by the local characteristics of the communication model, defined in RoboCup Rescue. This model is outlined in Figure 3, which each message sent by a local agent is distributed to all agents of its kind and his global agent. A message sent by a global agent is distributed to its own local agents (Police Office to Police Force) and to all non-humanoid agents (Police Office to Fire Station).

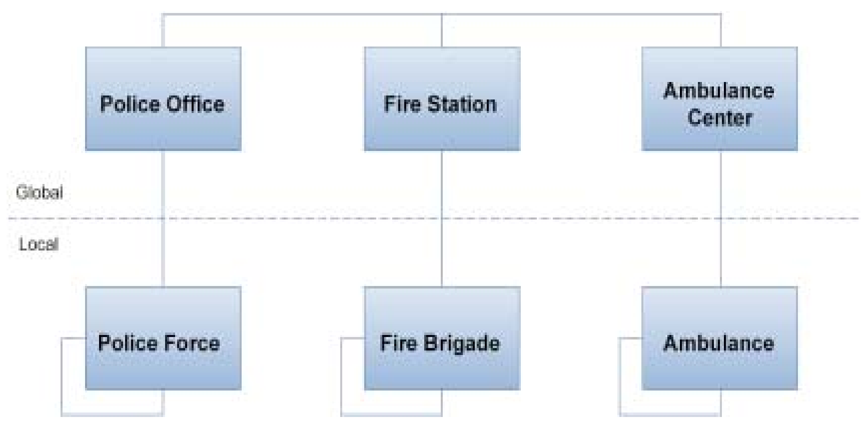



Figure 3. A Model of Communication between Global and Local Agents

Each simulator of a specific aspect of the domain updates the properties of geographic objects and calculates the effect of actions performed by each agent. For practical reasons, the computational power in the RoboCup Rescue is limited both in the server and the client. Each simulation is bound to the kernel to finish all calculations and the communication on the network within a discrete time of 500 milliseconds (or 0.5 seconds). Therefore, the usage of efficient algorithms is an important requirement [7].

The Traffic Simulator (TFS) covers the aspects of movement (handling) of the agents on the geographic area. The agents movement is interrupted by blockades on roads (Blockage Simulator) and the conflicting movement (e.g. in opposite directions on the same track) of agents can trigger a gridlock traffic [7].

The “Building Collapse and Road-Blockage Simulator” (BRS) deals with all aspects of the destruction caused by the collapse of buildings. The simulator controls the earthquake’s level of damage in each building. In turn, it follows that the collapse can cause the burial of humanoid agents who remain trapped inside the building or its vicinity. The most relevant aspect of this simulator is how it can restrict the humanoid agent survival [7].

The Fire Spread Simulator (FSS), also known as ResQ Fire, addresses the issues of fire in buildings, as well as its evolution and spread to near buildings. The evolution depends on factors such as the material of construction, the building area and the amount of water already deployed (by Fire Brigade) to extinguish the fire. The spread depends on the distance of the buildings and the number of neighbors. The fire is the only destructive effect that keeps on spreading after the end of the earthquake. The FSS was not designed to be applied only in cases of earthquakes, but also in cases of conventional fire [10].

The Visualization Simulator (VS) creates a virtual reality view and displays it for spectators (human beings). This simulator does not simulate a particular aspect of the field (like other simulators), but it helps to understand the flow of the simulation. The other simulators take care of many aspects not treated by the description above, such as the health evolution of the buried or injured humanoid agents. The Figure 4 illustrates how the existing simulators cooperate with each other to generate a full simulation [7].

**3. Swarm Platform’s Conceptual Model and Architecture**

The Swarm project started in 1994 by Chris Langton at Santa Fe Institute, New Mexico - USA. The Swarm’s conceptual model deals with a set of agents which interact with each other by discrete events. Generally speaking, an agent is any actor in a system or entity that triggers events that can affect it and other agents as well [6]. In the Swarm model, the main component that manages the agents is called SWARM. A SWARM is a set of agents with an activity time scheduling for them [3].

The Swarm model employs hierarchical levels - since agents can also be made of SWARM’s - thus creating nested structures [3]. Swarm model’s architecture keeps the observation actions and the simulation actions apart. In the architecture first level, called OBSERVER SWARM, the simulation can be watched. Therefore, what the user sees or perceives is controlled by this level [3, 6]. Among the actions performed in the OBSERVER SWARM, the main actions are:

- User interface management by creating observation screens for example;
- Second hierarchical level instantiation, called MODEL SWARM;
- File and/or chart generation based on MODEL SWARM resulting data.

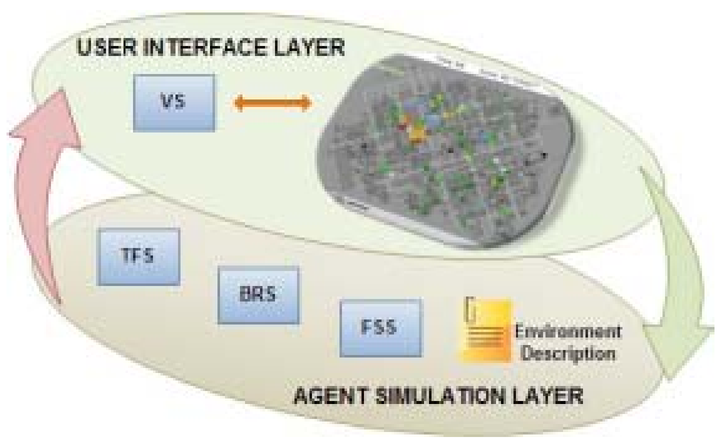



Figure 4. A Representative Model of Communication between the Simulation layers on the Agent Simulation Unit

The time scheduling at the OBSERVER SWARM leads the data gathering (getting the information and showing it to the user in an interactive manner, or storing that information in files when in batch mode). In the second hierarchical level, the MODEL SWARM runs the simulation. These are some of the actions performed at this level [3]:

- Simulation agents’ instantiation along with other possible hierarchical levels;

- MODEL SWARM agents' time-scheduling;
- Controlling of the actions performed by the simulation's agents;
- Gathering of information to be sent to the OBSERVER SWARM level.

### 3.1 The Architecture of Swarm Platform

What makes Swarms scientifically interesting, and often mathematically intractable, is the coupling between the individual and the group behaviors. Although the individuals might be relatively simple, their collective behavior can be quite complex. Swarms allow us to focus directly on the fundamental roots of complexity: they capture the point at which simplicity becomes complexity.

The swarm behavior as a whole emerges in a highly nonlinear manner from the individuals' behaviors. This emergence involves a critical feedback loop between the behavior of the individuals and the behavior of the whole collection. In a swarm, the combination of individual behaviors determines the collective behavior of the whole group. In other words, the behavior of the whole group determines the conditions (spatial and temporal patterns of information) which each individual makes its behavioral choices. Once more, these individual choices collectively establish the overall group behavior, continuously, in a never-ending loop.

Agents that receive messages are free to perform whatever computation they wish in response to the message. Typically, the response to a message will be the execution of some algorithm that captures the model agent behavior. Agents can also insert other actions into the schedule. That way, Swarm allows a discrete event style of simulation. Also, Swarm programmers are free to use simple, fixed schedules that continuously repeat the same actions.

## 4. Simulation Experiments and Data Analysis I

From the study of the RoboCup Rescue environment some simulations were performed in order to have a better understanding of the fire spread in cities and the cooperative actions among agents to achieve a common goal. For this study two scenarios were created, both using the city of Kobe (Japan) as the main map.

### 4.1 Scenario 1: "Fire out of Control"

In this first scenario were chosen three initial fire outbreaks, a point of refuge, three Fire Brigade agents, twenty (20) Civilian agents, two Ambulance agents, non-humanoids Fire Station and Ambulance Center agents. In this scenario, the Fire Brigade Agents had a low capacity to hold the fire. For that, the variables responsible for the water skills allowed for each agent, the initial amount of water each agent, the radius of sight related to notification of fire, the distance needed to fight the fire, the radius of communication allowed, the number of messages allowed, the amount of water available in shelters (water replenishment), the amount of water supplied in the refuge, among others were assigned in order to undermine the fight against the fire.

Then, in this simulation, the Fire Brigade Agents did not have a good amount of water available, the amount of water returned to the agents at the shelter was not fast enough, it was required to be very close to the target during the fight and their messages beyond a small scope were quantity restricted. These variables were set in order to provide a scenario which the fire could not be controlled by the agents (Fire Brigade). Thus, furnish a scenario which agents, modeled to escape from a city/place on fire for example, would have problems to escape. The Ambulance agents' configuration was not changed during the experiment.

Figure 5 shows the development of simulation in the cycle 94. According this figure, it is evident the low efficiency in fighting the fire spread. In this experiment, even with a reasonable amount of cycles, the Fire Brigade Agents did not manage to hold the fire which spreads quickly, thus impairing the action of the Ambulance agents to rescue the victims (Civilian Agent) from the fire.

### 4.2 Scenario 2: "Fire under Control"

In this scenario were chosen three initial outbreaks of fire, a point of refuge, three Fire Brigade agents, twenty (20) Civilian agents, two Ambulance agents, non-humanoids Fire Station and Ambulance Center agents. In this scenario, the Fire Brigade agents had a high capacity to hold the fire. In order to achieve this feature, the variables responsible for the water skills allowed for each agent, the initial amount of water each agent has, the radius of sight related to notification of fire, the distance required to fight the fire, the radius of communication allowed, the number of messages allowed, the amount of water available in shelters (water replenishment), the amount of water supplied in the refuge, among others were assigned in order to maximize the fight against the fire.


Figure 5. Uncontrolled Fire Scenario

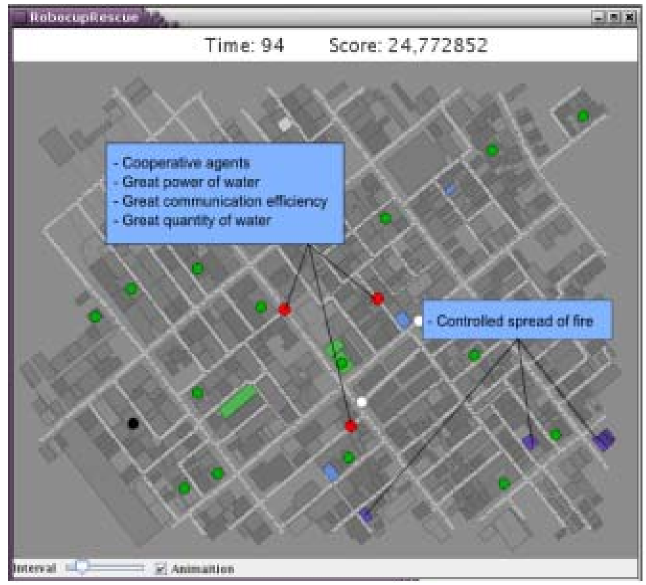



Figure 6. Controlled Fire Scenario

According to such setup, in this simulation the Fire Brigade Agents had a good amount of water available, the amount of water returned to the agents at the shelter was fast enough, it was not required to be very close to the target during the fight and their messages beyond a large scope were not limited in quantity. These variables were chosen to provide a scenario where the fire could be controlled by the agents (Fire Brigade). Thus, furnish a scenario where agents, modeled to escape from a city/place on fire for example, would not have problems to escape the fire. The Ambulance agents' configuration was not changed during the experiment.

Figure 6 shows the development of the simulation in the cycle 94. Despite the same time of simulation as the previous one (94 cycles) at this time it is possible to notice that few buildings and civilians were affected by the spread of fire. The results show that even with the same number of cycles, a good cooperation between humanoid Agents (Fire Brigade and Ambulance) and non-humanoid Agents (Fire Station and Central Ambulance Center) can provide satisfactory results.

## 5. Simulation Experiments and Data Analysis II

Based on studies of the Fire Spread Simulator, two scenarios were modeled and simulations where performed in order to get a better understanding of the fire spreading in cities. Both scenarios used the Foligno city (Italy) as a primary map and they were performed in the RoboCup Rescue Environment's Fire Spread Simulator.

### 5.1 Scenario 1: Fire wall

According to the proposed model in [5], the simulator settings were defined and a "fire wall" was created. In this model – applied in panic in crowds and evacuation procedure studies – a dangerous situation (fire) was designed, and that situation steadily grew. When the fire expanded it took a single direction in the map, in order to push certain agents to move in just one direction. Thanks to that unidirectional aspect, the fire forces the agents to take only one route: to the exit.

Figure 7 shows how the scenario proposed by Dirk Helbing was modeled in the fire simulator ResQFire. At Figure 7(a) the fire starts in a specific point and grows progressively, forming a unidirectional fire wall. In Figure 7(b) the fire, now with a strong and fast spreading, takes a single direction in the map, forcing certain agents to move in the direction that it is the opposite of the taken by the fire. Figure 7(c) shows the fire's whole path during the simulation, which the gray shaded buildings represent the places destroyed by the fire.

### 5.2 Scenario 2: Fire Ring

Based on the proposed model found in [12], the simulator settings were defined so a "Fire Ring" was created. In this model, the fire spreads according to a series of successive ignitions from the fuel bed. Starting from a specific point and increasing in a round shape while everything next to that point burns. Figure 8 shows how the spreading of this kind of fire occurs.

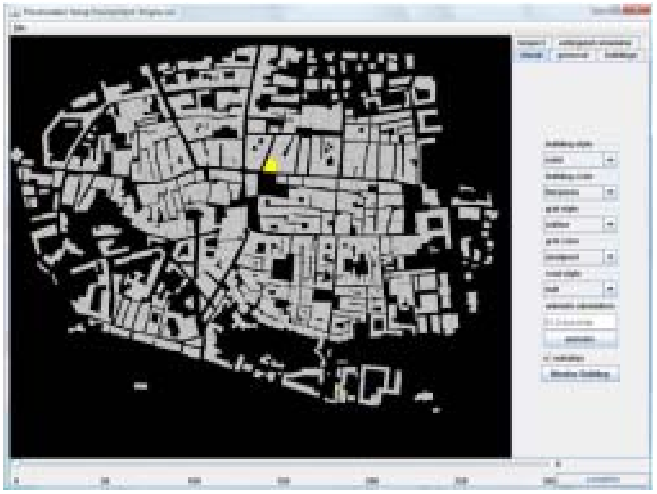
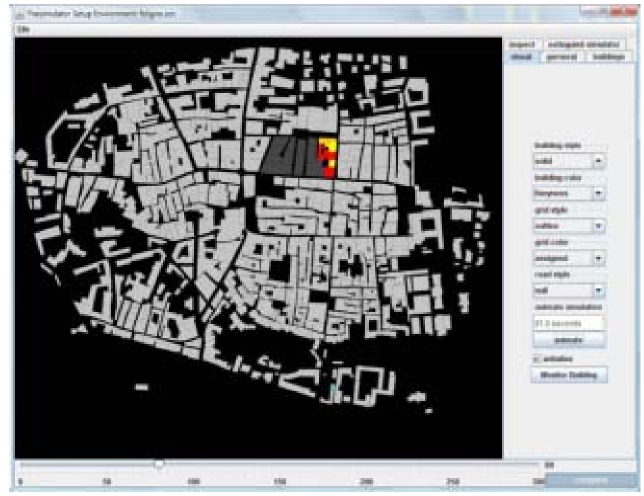


Figure 7(a). Fire Wall - $0^{th}$ cycle Figure



7(b). Fire Wall - $80^{th}$ cycle

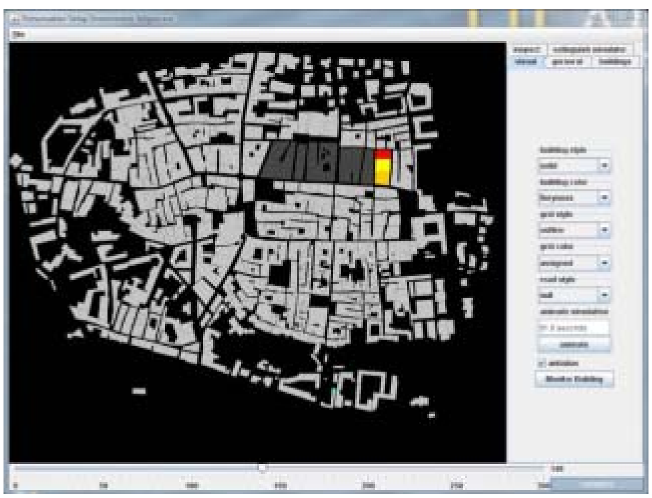


Figure 7(c). Fire Wall - $140^{th}$ cycle

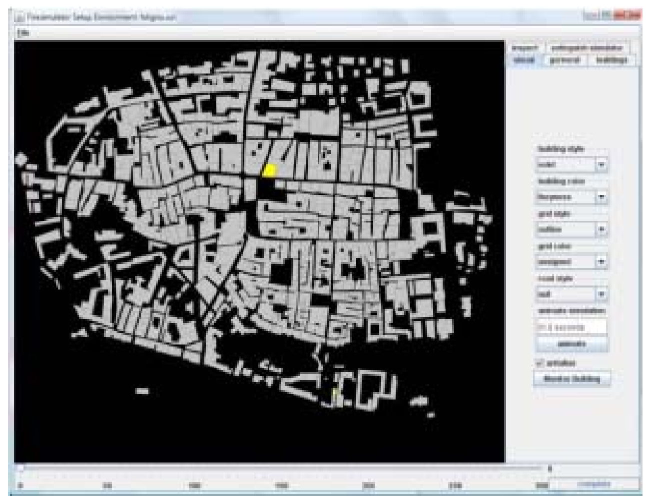


Figure 8(a). Fire Ring - $0^{th}$ cycle

In Figure 8(a) the fire starts from a certain point. As proposed by [12], the fire gradually expands to make a “Fire Node”. Figure 8(b) shows when the fire starting point expands, although its nodal characteristic is not yet clear. However, in Figure 8(c) the fire spreads in multiple directions because its propagation has become faster and stronger which, in turn, leads to a rounded-shaped expansion. In that configuration, the larger the diameter formed by the ring of fire, larger is the number of buildings affected.

**6. The Technological Puzzle: The Swarm Platform and the Fire Spread Simulator Working Together**

This section presents the integration of the SWARM platform [6] and the Robocup Rescue [10] fire simulator. These two technologies are combined to use their best features, such as the freedom in the agents’ architecture (SWARM) and the domain specification found in the fire simulator (ResQ Fire). These features can work in a collaborative way.

Figure 9 shows an example of the integration of these two platforms. Inside SWARM there are two worlds apart: the Agents’ world and the world where these agents live in. The agents, in this model, are designed according to SWARM platform architecture. On the other hand, the simulation environment (World) uses an interface able to bring ResQ Fire features to

SWARM environment. The fire simulator (composed of IO blocks, Util, World, Kernel and Simulator) is bound to SWARM platform's world to provide the required framework for a fire spreading simulation in a distinct domain. In order to aid such integration, the Java programming language is used and, thanks to its portability and ease of development, it provides the means required so the communication between the distinct computational environments can be established.

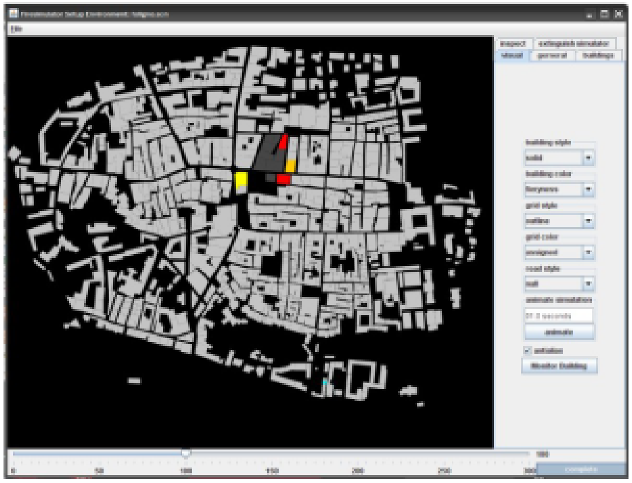


Figure 8(b). Fire Ring - 100$^{th}$ cycle

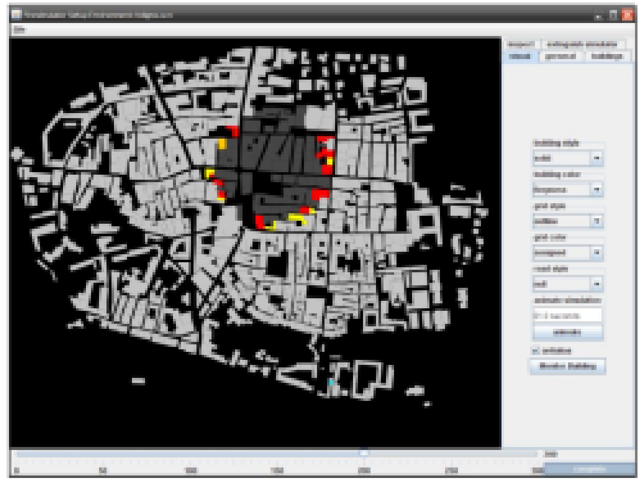


Figure 8(c). Fire Ring - 200$^{th}$ cycle

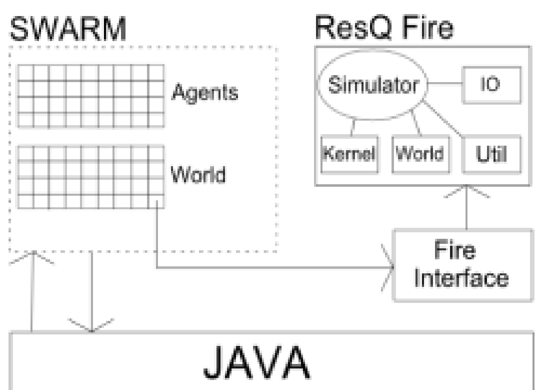



Figure 9. Platform Integration Model

### 7. Conclusions and Future Works

As it was previously seen, the Swarm platform is domain-independent and it can be applied to model in areas such as Economy, Ecology, Anthropology, Political Sciences, among others [3]. This all-purpose feature might pose as a disadvantage because the programmer does not have a full set of simulators and specific architectures at his disposal, having to build them from scratch. On the other hand, the RoboCup Rescue environment (RCR) is domain dependent, with a focus on simulations of disaster situations. For this task it offers a set of simulators and specific agents' architectures. The simulators model fire incidents, earthquakes, traffic, blockage, and so on. Usually, the agents are tightly bound to RCR's architecture which makes their manipulation difficult in certain scopes [9].

Finally, in future works, a hybrid architecture based on the Swarm platform and the RoboCup Rescue environment fire simulator (The Fire Spread Simulator) will be proposed. Such integration is significant because it allows the usage of the best aspects of each environment, optimizing the modeling and implementation process.

Both models based on [12] and [5] computational simulations were presented with their propositions. The first proposition deals with a "Fire wall" that goes in a single direction, from a starting point to another (marked by the end of simulation). The second proposition is represented by the “Fire Ring” (or “Fire Node”), which the fire spreads in a round shape, from a starting point up to the city limits (and the end of the simulation).

This paper proposed integration between the Swarm platform and the RoboCup Rescue’s Fire Spread Simulator in order to bind distinct technologies and systems in the same project. This hybrid approach highlights the advantages of the Swarm (freedom for the agents’ architecture) and Robocup Rescue (different simulators from specific domains) platforms. Since the 21st century major issues are related, their solutions must be related as well. In order to achieve this condition, the studies and the technological systems used to solve these issues must cooperate for a greater goal. Through themes and environments analyzed, this paper suggests that, in order to get a better understanding of the collective behavior and some of its surrounding elements (e.g. fire spreading), distinct technological systems (SWARM, FSS and Java) should be integrated. That leads to an environment where the best features of each platform can be combined and a new studying environment is born.

**References**

[1] Building rescue systems of the future. (2006). http://www.robocuprescue.orghtml

[2] Berends, P., G. (1999). Simulation as a research tool in management studies, *European Management Journal* 17 (6) 576–583.

[3] Burkhart, R. (1994). The swarm multiagent simulation system. OOPSLA’94 Workshop on The Object Engine.

[4] Deborah, Vakas Duong, J. G. (2004). Sister: a symbolic interactionist simulation of trade and emergent roles, *Artificial Societies and Social Simulation,* 8.

[5] Helbing, Dirk., Farkas, Illés J., Molnár, Péter., Vicsek, Tamás. (2002). Simulation of pedestrian crowds in normal and evacuation situations.

[6] Swarm, S. (2008). http://www.swarm.org, Accessed December.

[7] Kaneda, T., Matsuno, F., Takahashi, H., Matsui, T., Atsumi, M., Hatayama, M., Tayama, K., Chiba, R., Takeuchi, K. (2000). Simulator complex for robocup rescue simulation project as testbed for multiagent organization behavior in emergency case of large scale disaster. http://simons.intlab.soka.ac.jp/ matsumi/

[8] Morimoto, T. (2000). How to develop a robocuprescue agent. http://ne.cs.uec.ac.jp/ morimoto/rescue/manual/index.html

[9] Nardi, D., D’Agostino, F., Farinelli, A., Grisetti, G.., Iocchi, L. (2002). Adaptability of the robocup rescue simulator. Technical Report, University of Roma, Via Salaria 113, Roma I00198, Italy, August.

[10] Nüssle, T. A., Kleiner, A., Brenner, M. (2004). Approaching urban disaster reality: The resq firesimulator.

[11] U. of Bath. Bai (2007). Robocuprescue agent and infrastructure world championships. https://lists.bath.ac.uk/sympa/arc/bai/200711/msg00013.html.

[12] Almeida, Rodolfo Maduro, (2007). Modelo de propagação do fogo em incêndios de superfície baseado no método nodal. Master’s thesis, Instituto Nacional de Pesquisas Espaciais INPE, São José dos Campos SP Brasil, Fevereiro.

[13] Science and E. at The University of Edinburgh. (2005). Firegrid cluster for next generation emergency response systems. http://www.firegrid.org/.

**Author’s biography**

**Terry Lima Ruas.** Has Electrician Maintenance degree for SENAI in the Course of Industrial Education. Graduated in the course of Bachelor of Science and Technology at Federal University of ABC. An undergraduate course of Computer Science at Federal University of ABC. Fellow by PIBIC - CNPq, project in the area of Intelligent Systems. Co founder of the Group of Students of Computer Science. Serves as an intern at International Business Machines (IBM) on supporting web server applications.

**Maria das Graças Bruno Marietto.** Associate Professor at the Federal University of ABC (UFABC) Degree in mathematics from Catholic University Dom Bosco with a Masters in Systems Engineering and Computer Science Federal University of Rio de Janeiro (UFRJ / COPPE) and PhD from the Technological Institute of Aeronautics (ITA, Sao Jose dos Campos, SP). Acts in Distributed Artificial Intelligence, Multiagent Simulation, Collective Behavior in Crowds and the Semantic Web.

**Robson dos Santos França.** Master of Science (Information Engineering) and Social Simulations researcher at Federal University of ABC (UFABC). He is part of a multiagent simulation study group at UFABC that has published several papers about multiagent systems, social simulations, theoretical and practical usages of multiagent frameworks. Currently, Robson is analyzing new ways to deal with multiagent systems and social simulations for collective behavior phenomena, new techniques for validation and the usage of games along these technologies and concepts. He was born in Santo André – São Paulo, he is married and currently lives in São Paulo city.

**André Filipe de Moraes Batista.** Bachelor of Science and Technology at Federal University of ABC (UFABC). He is part of a multiagent simulation study group at UFABC that has published several papers about multiagent systems, social simulations, theoretical and practical usages of multiagent frameworks. Has experience in computer science, with emphasis on Artificial Intelligence, acts in Artificial Intelligence, Multiagent Simulation and Complex Systems. He works in the Kyatera (FAPESP) project and in the Working Groups of the Network National Education and Research Network (RNP).