ISSN 1870-4069

A BDI Agent System for the Collaboration of the Unmanned Aerial Vehicle

Mario Hernandez Dominguez, Jose-Isidro Hernández-Vega,
Dolores Gabriela Palomares Gorham, C. Hernández-Santos,
Jonam L. Sánchez Cuevas

Instituto Tecnológico de Nuevo León, División de Estudios de Posgrado e
Investigación, Nuevo León, Mexico

marioingmecatronica94@hotmail.com,jose.isidro.hernandez@itnl.edu.mx,
dolores.gabriela.palomares@itnl.edu.mx, carlos.hernandez@itnl.edu.mx,
jonam.leonel.sanchez@itnl.edu.mx
http://posgrado2.itnl.edu.mx

Abstract. This project describes the implementation of a BDI architecture for solving problems of collaboration and coordination among aerial mobile robots used through Unmanned Aerial Vehicle (UAV). A Multi-Agent System is implemented in people search and it is developed on Jason software.The benefits of developing this model is diverse such as optimizing the collaborative work among intelligent agents being able to use artificial intelligence in aerial mobile robots, improving the decision making process.

Keywords: Multi-agent system, distributed systems, collaborative agent.

1 Introduction

The purpose of this article is to present the simulation of a multi-agent model using the BDI architecture, where we can observe the way of communication to achieve a common goal of the agents. All this developed in a development environment in Jason 1.4.2.

This article contains an introduction,research related to multi-agent systems are exposed, then in the methodology we describe the theoretical resources that support the model of coordination and communication of multi-agent system, then we can locate the results, we expose images of the simulation model and we observe how agents are coordinated to achieve a common goal. And finally the conclusions of the authors of this investigation as well as discussions generated.

The contribution to the scientific community of this article is to disseminate the work being done on agents oriented programming with the help of Jason 1.4.2 software for the simulation and analysis of logic programming lines for multi-agent systems and to take this tool to improve the response of mobile robots.

pp. 113–124; rec. 2016-05-30; acc. 2016-09-07

Based on the "divide and conquer" principle, widely used in computer algorithms, it arises an existing problem in aerotronics area which consists in finding the way to work collaboratively between two unmanned aerial vehicles (UAV) in order to solve people searching jobs, using the Belief-Desire-Intention architecture coordination model (BDI).

Currently, UAVs are being implemented in various activities such as civilian areas (forest services, cartography, environment), agriculture, cinematography, military industry (security and border control), among others, and reviewing the history it shows that when the tasks are divided in an organized manner efficiency and breadth of the required objectives increases.

The collaboration and coordination among aerial mobile robot lies when an area is surrounded with more than one drone, there isn't collaboration between them to accomplish an assignment. On the other hand, the person to the assignment and the person controlling such equipment usually wastes time programming repeatedly the tasks to be performed, even when different drones do the same is the same work or part of the work, or while he is reviewing the information sent from both UAVs wasted in the searching that may be of vital importance to the victim or victims,an example would be in a mountain area where the forest guard service could have two drones that could generate the above described condition. As a result of the research, it is expected to build a communication bridge and coordination between drones, making it possible to collaborate between them. At the same time this will improve the autonomy of the unmanned vehicles insomuch as are based in BDI architecture.

After giving a short introduction to the investigation below is displayed in Figure 1 a very abstract model that may be taken to give a quick idea that we will see below in the simulation.

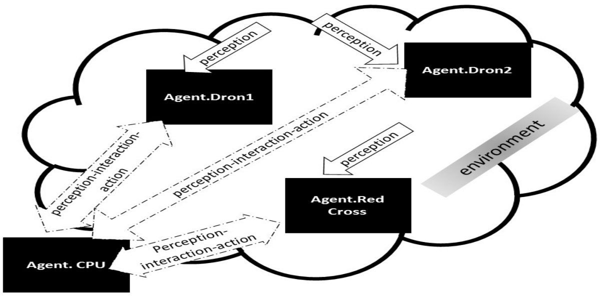


Fig. 1. An abstract model of the multi-agent system environment.

1.1 State of the Art

Agents using architecture BDI are characterized by having mental states(beliefs, desires, intentions) [13]. The great achievements of this architecture is due to the component mixture as a philosophical model of simple human reasoning to understand, a large number of implementations [10] and an abstract logical syntax that has been accepted by the scientific community [9,17,19]. Reviewing the literature we have found various kinds of advances in the area of artificial intelligence and even more in the implementation of the theory of agents for improving processes, these developments are as follows:

The air traffic management system (OASIS) [8] are integrated mostly of agents, all these designed to following the structure of agents in any of their category. The odd in this system is the selection of a flight's trajectory. In this system the agents collaborate to choose the best path reduce the fuel consumption and rise the flight's productivity. This ensures the target in the estimated time of arrival. This system has been used in the Sidney airport. Tasks Control Air Traffic are divided into sub-blocks in architecture OASIS and an agent is selected to solve each one of those sub-blocks. Each agent solves its part independently and the social cooperation with the rest define the course of the Global System. An operation agent will then exist for each airplane other : In addition to agents acting as controllers path, sequencers, Wind Moderators, Coordinators [3]. A modeling system dogfight with the name SWARMM [16] which can encode the tactics adopted by pilots in combat areas and generate agents that real pilots can participate in simulation studies. The system models from the physical aspects of airplane like the tactical knowledge and way of thinking of the pilots.

In [5] a deliberative architecture the methodological framework in systems that provide case-based reasoning is observed. The system of reasoning leads building plans from the beliefs and intentions. Regarding the problem to be solved and methods used in each of the four stages defined by the system of reasoning, the agent can plan on the completion time using variational calculation [11] .This model has been handled in the development of industrial applications, the construction sector [4] and the promotion of mobile applications [5].

Agents comprising the AFRAS System(A Fuzzy Reputation Agent System) are dedicated to the buying and selling of services. The AFRASS system aims to simulate the operation of human society in the management and communication of this prestige generated by people when making a purchase, and the informal bargaining that takes place between individuals [2].

Agents were presented and formally defined a BDI architecture for high-level control of land mobile robots soccer . This architecture builds on top of a layer system . This system architecture provides the necessary level of abstraction that makes cognitive reasoning. This design allows abstraction and modulation of the different aspects of the complex domain represents the robotic football [12].

To the research area of UAVs, where the research results propose an integration between a UAV team and a camera for image processing automatically [20].

Das [7] proposes the implementation of an existing computer architecture to control the behavior of autonomous underwater vehicles (AUVs), reactive deliberative approach is the most effective and meaningful to control vehicle behavior. However, little work has been done in the field of modeling the system to simulate and analyze the dynamic behavior run hybrid control architecture.

Christopher sets out the implementation of a distributed system client slave collaboration of three marine robots which maintains control of a formation preserving the distances between them and their positions [14].

The first steps for the integration of a system of distributed discrete event simulation within a framework of intelligent software agents are proposed. It begins with a single agent that implements the BDI approach which is applied to the monitoring for TBM excavation project [15].

The goal was to design and equalize a model based on a BDI architecture for collaborative search tasks between two drones. The innovation this model is the use of targeted agents under BDI architecture for collaborative work on unmanned aerial vehicles.

2 Methodology

According to [6] the organization in the multi-agent system (M.A.S.) depends on the kind of communication and the way agents cooperate between them, as well as the kind of agent the group comprises. Generally speaking there are three basic organizational configuration types:

1. Centralized Structure: In this type of configuration exist an agent that controls the interaction of the other agents of the system because it has the information or the functionality to do so.
2. Horizontal structure.
3. Hierarchical structure.
4. AD HOC structure.

The main goal that was accomplished with the model was to see how is the behavior of the entities based on the BDI architecture and how speaking actions will be executed to generate a collaboration and coordination environment between mobile flying robots. The table 1 describes PEAS (Performance, Environment, Actuators, Sensors) of the task environment that is used in this model and where some of the characteristics of the implemented agents are presented.

Table 1. PEAS description of the task environment for the multi-agent system.

Type Agent	Performance Measures	Environment – Forest	Actuators	Sensors
Agent — Controller — CPU-brain	Planning time	persons, agents	Sending messages	Message receiver
Agent — Receptor — DRON1,2	Minimize lost people searching time	Agents	Alarm	Camera, GPS
Agent — Receptor — Red-Cross	Number of rescued persons.	Persons, Agents	Motors figure	GPS

In the table 2 shown the task environment characteristics for this model are extended and described [18].

Table 2. Task environments and their characteristics.

task environment	Observable	Deterministic	Episodic	Static	Discreet	Agent
Searching environment	Totally	Stochastic	Sequential	Dynamic	Continue	Cooperative Multi-agent

- It is considered totally observable, due to the fact that sensors give the agent access to the complete state of environment in every moment of his route.
- It is considered the stochastic system, due to the fact that the next medium state isn't totally determined by the actual state and the executed action by the agent because it can't be possible to exactly predict neither human behavior nor the atmospheric or climatic conditions.
- It is considered sequential, due to the fact that decisions can affect future actions.
- It is considered dynamic, due to the fact that environment can change when the agent is delivering.
- It is considered Continuous, due to the fact that location of the drones goes through a range of continuous values in a soft way over time.
- It is considered Cooperative Multi-agent, due to the fact that finding missing people maximizes performance measure of the agents in a multi-agent environment that is partially cooperative.

2.1 Agents Characteristics and the Environment

Communication in multi-agent systems is usually based on the speaking actions paradigms and communication languages between agents KQML and FIPA [1]. The main speaking actions of the implemented agents in the model are represented in figure 2. These actions help us to generate a collaboration environment in order to reach the objective assigned to the agents.

States of Agents Tables 3, 4, 5 present how agent planning is defined and table 6 presents the environment methods.

Table 3. Agent CPU.

Plan	Description
+! Find (object)	This plan execute's the searching system starts working.
!+End (research1)	When this plan is executed, assumptions of a lost item are lowered.
! + Finish (base1):	When this plan is executed,assumption of returning to the base are lowered since it's already there.
+ Saving (person)	When this plan is executed, Red Cross is notified that has a person to rescue.
+! Protected (person)	When this plan is executed, Red Cross is notified that there isn't any person to be rescued.

Agents and Perceptions

- CPU Agent.
  - This agent is responsible for distributing and coordinating the main task.

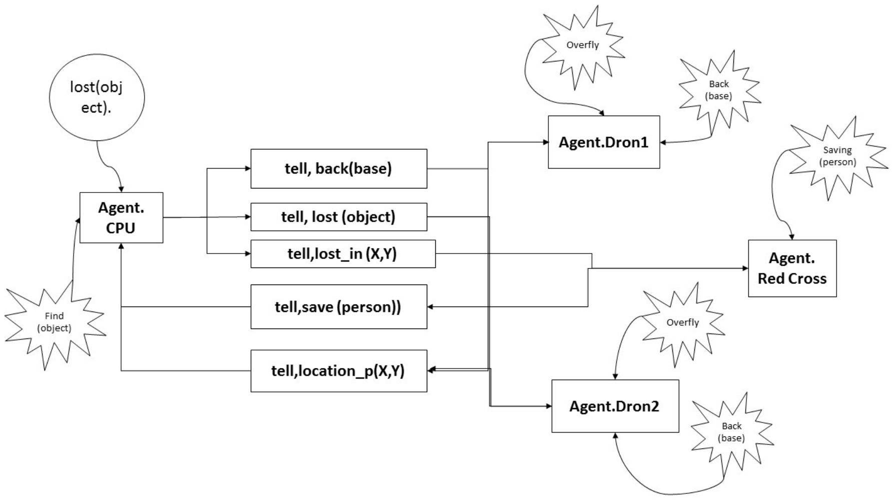


Fig. 2. System overview diagram.

Table 4. Agent Red Cross.

Plan	Description
+ Saving (person)	This plan is executed, a variety of rescuing plans are made.
+! Carry_to (R)	This plan is executed, victim is located in order to be rescued.
+ Attending (person)	This plan is executed, there is a transportation to the place where the person is.
+! At (L)	Corroboration about being in the same place as the victim is done.
+! Carry_to1 (R)	When this plan is executed, victim is returned to the right place.
! Take (S, L)	This plan is executed, a simulation of grabbing the victim is done.
+! Ensure_obj (S)	This plan is executed, a simulation of the victim being in the vehicle is done.
+! At1 (L)	This plan is executed, victim is returned to the starting spot.

  - Perceptions of this type of agent are present in the speaking messages with the agent, mainly because perceptions depend on the other agents messages.
- UDOO1,2 Agent.
  - This agent is in charge of reviewing the assigned area by the CPU in the search for missing people.
  - Perceptions of this type of agent are generated by cameras and GPS.
- Red Cross Agent .
  - This agent is in charge of bringing help to people
  - Perceptions of this kind of agents are observed using a GPS.

Environment structure How environment in the JAVA programming language is structured, to go from the modeling state to the simulation part.

- Environment.
  - Libraries.- Java and Jason libraries were used in order to be able to use methods of both programming languages.

Table 5. Agent Dron1,2.

Plan	Description
+! Overfly (agent_CPU, object)	When this plan is executed, agents do the selected tour.
+ End_r (r2)	When this plan is executed, the CPU is notified that the tour is over.
+ Camera (r2)	When this plan is executed, a person is found on the map.
+! Back (base)	When this plan is executed, drones have to return from the selected point to the starting spot.

Table 6. Agent Environment.

Plan	Description
UpdatePercepts ()	This method includes the methods that are called by the agents.
NextSlot ()	Move one space to UDOO1 inside the grip.
NextSlot1 ()	Move one space to UDOO2 inside the grip.
MoveTowards (int x, int y)	Move the Red Cross to the victim spot.
To return()	Move one space from the grip to UDOO1 base.
To return1 ()	Move one space from the grip to UDOO2 base.
Pickobj ()	Grab the victim in the map.
Dropobj ()	Place the victim in the base.
Burnobj ()	Register the victim as sheltered.

- Environment extends Environment.- The model are declared as well as the starting locations that are sent to the agents "updatePercepts()".
- MarsModel extends GridWorldModel.- Locations are asigned to the graphic model and the methods to answer to a executeAction class calling.
- MarsView extends GridWorldView.- This class brings life to our simulation because color and labels to the model.
- ExecuteAction.-This type is where sent directions from the agents are classified. Directions have to be reflected on the environment.

In table 7 Speaking actions applied to the model are presented as well as one agents activities cycle, starting from CPU agent assuming the believing of having a missing person and going through the requirement to the drones of flying over certain area, the assistance of the Red Cross agent and concluding when drones return to the base.

3 Results

The results are presented as a simulation held in Jason software, hereafter it is presented a of images where the interaction among the agents and environment.The algorithm executed down filing a joint policy for multi-agent where making a comparison in the simulation between the algorithm with a single unmanned aerial vehicle in the environment and 2 UAV, which is presented below, we saw that the search time is reduced 50% on the search area.

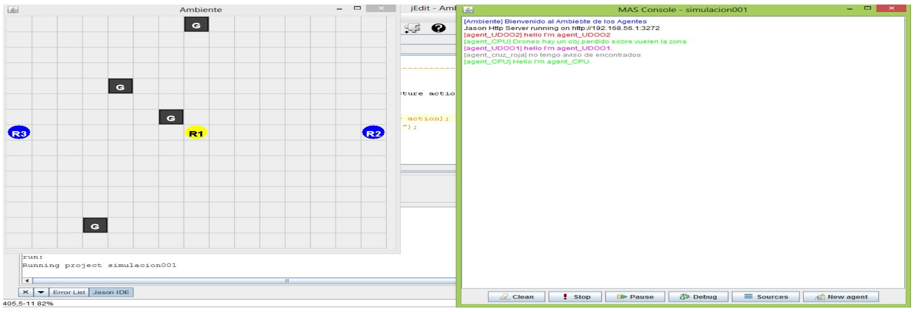
Table 7. Speech acts within one simulation cycle.

Cycle	Agent	Actions	Belief	Goals	Plans
1	CPU		lost (object)		
2		.send ([agent_UDOO1, agent_UDOO2], tell, lost (object))			
3		.send ([agent_UDOO1, agent_UDOO2], achieve, overfly (agent_CPU, object))			
4	UDOO1,2		lost (object)		
5					! Overfly (agent_CPU, object)
6	Red Cross			saving (person)	
7	UDOO1,2				+ camera (r2)
8		.send (agent_CPU, tell, location_p (X, Y))			
9		.send (agent_CPU, achieve, save (person))			
10	CPU		location_p (X, Y)		
11					+ saving (person)
12		.send (agent_Red_Cross, tell, lost_en (X, Y))			
13		.send (agent_Red_Cross, achieve, save (person))			
14	Red Cross		lost_en (X, Y)		
15					save (person)
16		.send (agent_CPU, achieve, protected (person))			
17	CPU				protected (person)
18		.send (agent_Red_Cross, untell, lost_en (X, Y))			
19		.send (agent_Red_Cross, unachieve, save (person))			
20	Red Cross		-lost_en (X, Y)		
21					-save (person)
22	UDOO1,2				+ end_r (r2)
23		.send (agent_CPU, achieve, terminate (research1))			
24	CPU				end (research1)
25		.send ([agent_UDOO1, agent_UDOO2], untell, lost (object))			
26		.send ([agent_UDOO1, agent_UDOO2], tell, back (base))			
27		.send ([agent_UDOO1, agent_UDOO2], achieve, return (base))			
28	UDOO1,2		-lost (object)		
29			return (base)		
30					return (base)
31	UDOO1,2				+ end(r2)
32		.send (agent_CPU, achieve, terminate (base1))			
33	CPU				end (base1)
34		.send (agent_UDOO1, untell, back (base))			
35	UDOO1,2		-back (base)		

Time 1 Shows the beginning of the search within the organized environment by the CPU agent, thus, the collaboration between air agents UDOO1 and UDOO2, to make path in the area.

Time 2 Shows how the red cross land agent goes to the point located by air agents UDOO1, 2 where a person is.

Time 3 Shows how the red cross land agent returns with the victim to a safe area.

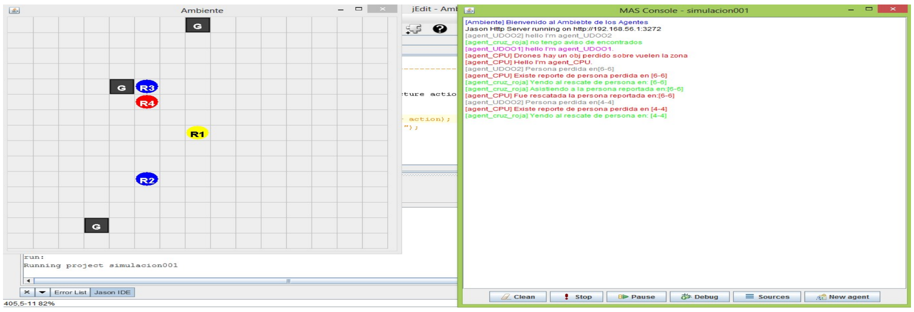


```
run:
Running project simulacion001
```

MAS Console - simulacion001

```
[Ambiente] Bienvenido al Ambiebte de los Agentes
Jason Http Server running on http://192.168.56.1:3272
[agent_UDOO2] hello I'm agent_UDOO2
[agent_CPU] Drones hay un obj perdido sobre vuelen la zona
[agent_UDOO1] hello I'm agent_UDOO1.
[agent_cruz_roja] no tengo aviso de encontrados
[agent_CPU] Hello I'm agent_CPU.
```

Fig. 3. Graphical interface of the multi-agent system environment.

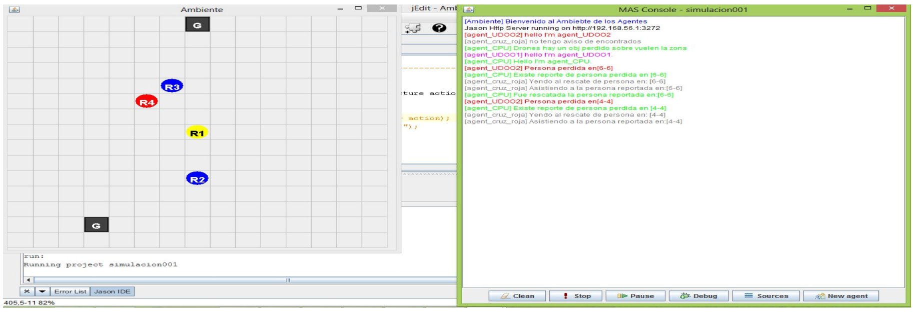


```
run:
Running project simulacion001
```

MAS Console - simulacion001

```
[Ambiente] Bienvenido al Ambiebte de los Agentes
Jason Http Server running on http://192.168.56.1:3272
[agent_UDOO2] hello I'm agent_UDOO2
[agent_cruz_roja] no tengo aviso de encontrados
[agent_UDOO1] hello I'm agent_UDOO1.
[agent_CPU] Drones hay un obj perdido sobre vuelen la zona
[agent_CPU] Hello I'm agent_CPU.
[agent_UDOO2] Persona perdida en[6-6]
[agent_CPU] Existe reporte de persona perdida en [6-6]
[agent_cruz_roja] Yendo al rescate de persona en: [6-6]
[agent_cruz_roja] Asistiendo a la persona reportada en:[6-6]
[agent_CPU] Fue rescatada la persona reportada en:[6-6]
[agent_UDOO2] Persona perdida en[4-4]
[agent_CPU] Existe reporte de persona perdida en [4-4]
[agent_cruz_roja] Yendo al rescate de persona en: [4-4]
```

Fig. 4. Graphical interface of the multi-agent system environment.



```
run:
Running project simulacion001
```

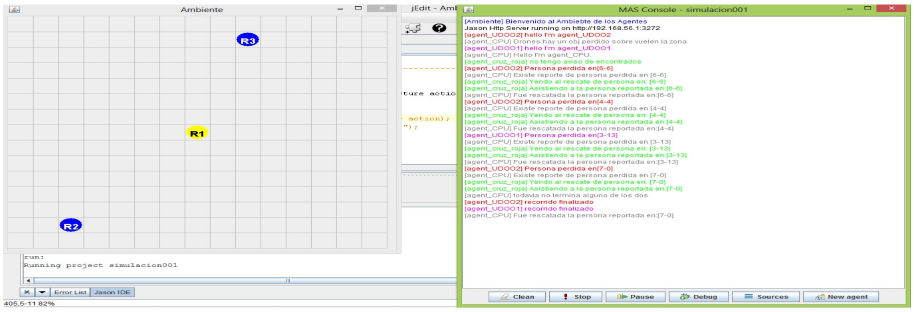
MAS Console - simulacion001

```
[Ambiente] Bienvenido al Ambiebte de los Agentes
Jason Http Server running on http://192.168.56.1:3272
[agent_UDOO2] hello I'm agent_UDOO2
[agent_cruz_roja] no tengo aviso de encontrados
[agent_CPU] Drones hay un obj perdido sobre vuelen la zona
[agent_UDOO1] hello I'm agent_UDOO1.
[agent_CPU] Hello I'm agent_CPU.
[agent_UDOO2] Persona perdida en[6-6]
[agent_CPU] Existe reporte de persona perdida en [6-6]
[agent_cruz_roja] Yendo al rescate de persona en: [6-6]
[agent_cruz_roja] Asistiendo a la persona reportada en:[6-6]
[agent_CPU] Fue rescatada la persona reportada en:[6-6]
[agent_UDOO2] Persona perdida en[4-4]
[agent_CPU] Existe reporte de persona perdida en [4-4]
[agent_cruz_roja] Yendo al rescate de persona en: [4-4]
[agent_cruz_roja] Asistiendo a la persona reportada en:[4-4]
```

Fig. 5. Graphical interface of the multi-agent system environment.

Time 4 Shows how air agents UDOO1, 2 when arriving to the final point, they return to base.

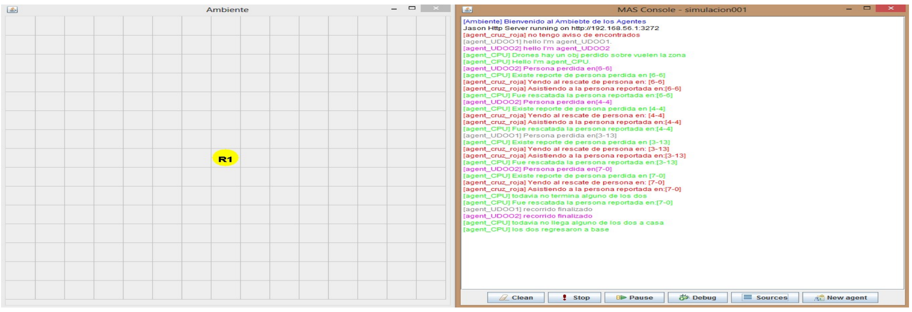
Time 5 Shows all speech acts of agents in the environment as a result of the collaboration among them.



```
run:
Running project simulacion001
```

```
[Ambiente] Bienvenido al Ambiebte de los Agentes
Jason Http Server running on http://192.168.56.1:3272
[agent_UDOO2] hello I'm agent_UDOO2
[agent_CPU] Drones hay un obj perdido sobre vuelen la zona
[agent_UDOO1] hello I'm agent_UDOO1.
[agent_CPU] Hello I'm agent_CPU.
[agent_cruz_roja] no tengo aviso de encontrados
[agent_UDOO2] Persona perdida en[6-6]
[agent_CPU] Existe reporte de persona perdida en [6-6]
[agent_cruz_roja] Yendo al rescate de persona en: [6-6]
[agent_cruz_roja] Asistiendo a la persona reportada en:[6-6]
[agent_CPU] Fue rescatada la persona reportada en:[6-6]
[agent_UDOO2] Persona perdida en[4-4]
[agent_CPU] Existe reporte de persona perdida en [4-4]
[agent_cruz_roja] Yendo al rescate de persona en: [4-4]
[agent_cruz_roja] Asistiendo a la persona reportada en:[4-4]
[agent_CPU] Fue rescatada la persona reportada en:[4-4]
[agent_UDOO1] Persona perdida en[3-13]
[agent_CPU] Existe reporte de persona perdida en [3-13]
[agent_cruz_roja] Yendo al rescate de persona en: [3-13]
[agent_cruz_roja] Asistiendo a la persona reportada en:[3-13]
[agent_CPU] Fue rescatada la persona reportada en:[3-13]
[agent_UDOO2] Persona perdida en[7-0]
[agent_CPU] Existe reporte de persona perdida en [7-0]
[agent_cruz_roja] Yendo al rescate de persona en: [7-0]
[agent_cruz_roja] Asistiendo a la persona reportada en:[7-0]
[agent_CPU] todavia no termina alguno de los dos
[agent_UDOO2] recorrido finalizado
[agent_UDOO1] recorrido finalizado
[agent_CPU] Fue rescatada la persona reportada en:[7-0]
```

Fig. 6. Graphical interface of the multi-agent system environment.



```
[Ambiente] Bienvenido al Ambiebte de los Agentes
Jason Http Server running on http://192.168.56.1:3272
[agent_cruz_roja] no tengo aviso de encontrados
[agent_UDOO1] hello I'm agent_UDOO1.
[agent_UDOO2] hello I'm agent_UDOO2
[agent_CPU] Drones hay un obj perdido sobre vuelen la zona
[agent_CPU] Hello I'm agent_CPU.
[agent_UDOO2] Persona perdida en[6-6]
[agent_CPU] Existe reporte de persona perdida en [6-6]
[agent_cruz_roja] Yendo al rescate de persona en: [6-6]
[agent_cruz_roja] Asistiendo a la persona reportada en:[6-6]
[agent_CPU] Fue rescatada la persona reportada en:[6-6]
[agent_UDOO2] Persona perdida en[4-4]
[agent_CPU] Existe reporte de persona perdida en [4-4]
[agent_cruz_roja] Yendo al rescate de persona en: [4-4]
[agent_cruz_roja] Asistiendo a la persona reportada en:[4-4]
[agent_CPU] Fue rescatada la persona reportada en:[4-4]
[agent_UDOO1] Persona perdida en[3-13]
[agent_CPU] Existe reporte de persona perdida en [3-13]
[agent_cruz_roja] Yendo al rescate de persona en: [3-13]
[agent_cruz_roja] Asistiendo a la persona reportada en:[3-13]
[agent_CPU] Fue rescatada la persona reportada en:[3-13]
[agent_UDOO2] Persona perdida en[7-0]
[agent_CPU] Existe reporte de persona perdida en [7-0]
[agent_cruz_roja] Yendo al rescate de persona en: [7-0]
[agent_cruz_roja] Asistiendo a la persona reportada en:[7-0]
[agent_CPU] todavia no termina alguno de los dos
[agent_CPU] Fue rescatada la persona reportada en:[7-0]
[agent_UDOO1] recorrido finalizado
[agent_UDOO2] recorrido finalizado
[agent_CPU] todavia no llega alguno de los dos a casa
[agent_CPU] los dos regresaron a base
```

Fig. 7. Graphical interface of the multi-agent system environment.

4 Discussion

In the prior art can be seen that multi-agent developed in sequential programming models are more complex than those currently proposed to the programming paradigm agent oriented, where, as it is easier to the programmer to give the knowledge to the agent, since working with a high-level language, as for the agent that helps it having more autonomy to obtain information from the environment where it is and act accordingly. Having this kind of study done, it is expected it can channel new multi-agent models applicable to air mobile robots on this new type of programming paradigm.

5 Future Work

Future work will focus on bringing this simulation to practice and to be able to materialize using embedded cards as UDOO NEO as well as also to generate collaboration and coordination between UAVs as Parrot Bebop Drone and the laptop,which is shown in Figure 8.

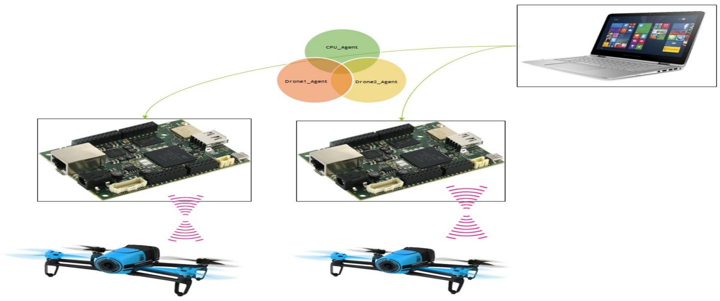


Fig. 8. Scheme of the multi-agent system.

6 Conclusion

It is concluded that collaborative communication environment may be generated among air mobile robots using intelligent agents so that we can get closer to human intelligence reproduction with the help of new programming paradigms as are, oriented agents and BDI architectures. It was, it was observed that the multi-agent systems facilitate solving tasks due to integration of diverse robotic entities within the same objective. In their daily basis because they can delegate to more complex robotic entities responsibilities. All the described above shows that these models are a new research tool with a very extensive application field in years to come.And we can conclude that the simulation based on the BDI architecture can generate a collaboration and coordination, including UAV's. because in the simulation UAVs share their tasks to achieve its objectives.

References

1. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason, vol. 8. John Wiley & Sons (2007)
2. Carbo, J., Molina, J.M., Davila, J.: Trust management through fuzzy reputation. International Journal of Cooperative Information Systems 12(01), 135–155 (2003)
3. Corchado, J.M.: Agencia: Una puerta hacia la convergencia de la inteligencia artificial. Universidad de Salamanca (1999)
4. Corchado, J.M., Laza, R.: Constructing deliberative agents with case-based reasoning technology. International Journal of Intelligent Systems 18(12), 1227–1241 (2003)
5. Corchado, J.M., Pavón, J., Corchado, E.S., Castillo, L.F.: Development of cbr-bdi agents: a tourist guide application. In: European Conference on Case-based Reasoning. pp. 547–559. Springer (2004)
6. Corchado, J.M.: 2 modelos y arquitecturas de agente
7. Das, S., Shome, S., Nandy, S., Pal, D.: Modeling a hybrid reactive-deliberative architecture towards realizing overall dynamic behavior of an auv. Procedia Computer Science 1(1), 259–268 (2010)
8. Georgeff, M., Rao, A.: Rational software agents: from theory to practice. In: Agent technology, pp. 139–160. Springer (1998)

9. Georgeff, M., Pell, B., Pollack, M., Tambe, M., Wooldridge, M.: The belief-desire-intention model of agency. In: International Workshop on Agent Theories, Architectures, and Languages. pp. 1–10. Springer (1998)
10. Georgeff, M.P., Lansky, A.L.: Reactive reasoning and planning. In: AAAI. vol. 87, pp. 677–682 (1987)
11. Glez-Bedia, M., Corchado, J., Corchado, E., Fyfe, C.: Analytical model for constructing deliberative agents. Engineering Intelligent Systems for Electrical Engineering and Communications 10(3), 173–185 (2002)
12. Gottifredi, S., Tucat, M., Corbatta, D., García, A.J., Simari, G.R.: A bdi architecture for high level robot deliberation. In: XIV Congreso Argentino de Ciencias de la Computación (2008)
13. Haddadi, A., Sundermeyer, K.: Belief-desire-intention agent architectures. Foundations of distributed artificial intelligence pp. 169–185 (1996)
14. Mas, I., Kitts, C.A.: Dynamic control of mobile multirobot systems: The cluster space formulation. IEEE Access 2, 558–570 (2014)
15. Ourdev, I., Xie, H., AbouRizk, S.: An intelligent agent approach to adaptive project management. Tsinghua Science & Technology 13, 121–125 (2008)
16. Rao, A.S., Georgeff, M.P.: An abstract architecture for rational agents. KR 92, 439–449 (1992)
17. Rao, A.S., Georgeff, M.P.: Decision procedures for bdi logics. Journal of logic and computation 8(3), 293–343 (1998)
18. Russell, S., Norvig, P.: Inteligencia artificial: Un enfoque moderno (2004)
19. Schild, K.: On the relation between standard bdi logics and standard logics of concurrence. In: Proceedings of the Fifth International Workshop on Agent Thories, Architectures and Languages (ATAL-98), Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg (1999)
20. Srikanth, M., Bala, K., Durand, F.: Computational rim illumination of dynamic subjects using aerial robots. Computers & Graphics 52, 142–154 (2015)