COOPERATIVE AUTONOMOUS UNMANNED AERIAL VEHICLES FOR DATA COMMUNICATION AND FOR MONITORING UNSTRUCTURED ENVIRONMENTS

Paulo Fernando Ferreira Rosa
Instituto Militar de Engenharia Praça General Tibúrcio 80, Praia Vermelha Rio de Janeiro - RJ - CEP 22290-270
rpaulo@de9.ime.eb.br

Paulo César Pellanda
Instituto Militar de Engenharia Praça General Tibúrcio 80, Praia Vermelha Rio de Janeiro - RJ - CEP 22290-270
pellanda@ime.eb.br

Carlos Alberto Padilha Pinheiro
Instituto Militar de Engenharia Praça General Tibúrcio 80, Praia Vermelha Rio de Janeiro - RJ - CEP 22290-270
carlos_pinheiro@terra.com.br

Abstract. *This article reports our progress in the developing of a system of cooperative autonomous unmanned aerial vehicles using the Multiagent Systems Engineering (MaSE) methodology. Illustrative fragments of the system models produced during the analysis phase are presented. Some topics of the MaSE methodology are discussed.*

Keywords: *robotic, multiagent, blimp, MaSE*

1. Introduction

The use of unmanned aerial vehicles has grown in recent years. Surveillance, exploration, monitoring, and transportation tasks are among potential uses for this vehicles class (Gomes and Ramos, 1998). Data communication is another obvious potential use for these devices. We intend to use this type of vehicle in environmental monitoring and data communication. This article reports our progress in the developing of a system of cooperative autonomous unmanned aerial vehicles using the Multiagent Systems Engineering (MaSE) methodology.

There are many feasible approaches for modeling a set of unmanned aerial vehicles. We chose the multiagent paradigm. Agents have been used to model this kind of vehicle (Huff, Kamel, and Nygard, 2003). In our study, several vehicles organize themselves to perform tasks. Both circumstances, vehicles modeled as agents, and several vehicles working together, induce a multiagent approach to the problem. In fact, a set of cooperative autonomous unmanned aerial vehicles is viewed here as a multiagent system.

We decided to use the MaSE methodology (DeLoach, Wood, and Sparkman, 2001) to model the system. There are many reasons for that, one of them being that MaSE is a multiagent methodology and another one being that our system is viewed as a multiagent system. We prefer to work in top-down approach. Top-down approach is a way to deal with complexity. In this approach, information is recoded into larger and richer chunks – level of abstraction –, so it is possible to maintain a larger amount of information in designers and analysts immediate memory (Miller, 1956). Top-down approach, is a natural approach, since it is in accordance with human conceptual limits. Further, top-down approach is an easy way to assure accomplishment of the actor goals. MaSE is a top-down methodology. MaSE also has a tool, namely agentTool, which facilitates modeling. Finally, MaSE covers all lifecycle of the system development. Therefore, MaSE and multiagent approach choices enforce and complement mutually one another one.

This paper focuses on the high-level design of a set of cooperative unmanned autonomous aerial vehicles. This set of vehicles must provide communication facilities, monitor facilities, and must have cooperative navigation capabilities.

2. Preliminary Requirements

Our system will be used in aerial environmental monitoring studies that require low speed, and low-altitude data gathering vehicles. Such vehicles must also be able to hover above an area and must have extended airborne capabilities for long-term studies. They must generate very low vibration, so as not to disturb the environment that is being measured and monitored and to reduce sensor noise and hardware malfunction. They must be able to take off and land vertically, so that maintenance and refueling can be done without the need for runways – this way, remote or difficult to access regions with limited logistics support can be monitored. They must be highly maneuverable, have a large payload capacity, and have a low operation cost. Of the four possible aerial vehicles, – airplanes, helicopters, airships, and balloons – airships are more suited to environmental monitoring tasks than the others (Elfes et al., 1998) are. Accordingly, airship is our air vehicle choice.

2.1. Blimp Overview

Today's airships are non-rigid, and are often called "blimps". Besides theirs flight characteristics just mentioned above, blimps have others interesting features. They are non-rigid, as within the envelope (which is the aerodynamically shaped "skin" that we see) there is no structure. Within the envelope, there are just two gases, nonflammable helium and air. These two gases are separated from each other as the air is contained in a relatively small inner envelope called a "ballonet". The ballonet serves a very important purpose: it accommodates the expansion and contraction of the helium as the airship rises and descends, and to a lesser extent due to increases/decreases in temperature. If it were not for the ballonet, as the airship gains altitude, the helium would expand and force open the over pressure valve, letting the lifting gas escape into the atmosphere; with the ballonet system, the expanding helium merely forces air out of the ballonet. On descent, ram air or electric blowers force air back into ballonet, maintaining the desired pressure (Brandreth, Jr, 2000).

Gas pressure within the envelope is very low, so even if the envelope of the blimp is punctured, there are no practical consequences to the flight. Many envelopes of advertising airships (as seen in sports events) get bullet holes in them, and these holes are not noticed until they are discovered during routine maintenance. Helium, which is a natural fire extinguisher, provides most of the lift in blimps. The aerodynamic shape of the envelope contributes to an additional lift, directly related to the speed and angle of attack of the airship (Brandreth, Jr, 2000).

Blimps are truly lighter than air, so blimps always fly at or near balance. They normally take off slightly "heavy" and can land somewhat "light". For most airships, the typical range is ~700 lbs. heavy ~300 lbs. light. That is the impact during the course of the mission; therefore, the airship is restricted to loosing about 1000 lbs. due to fuel burn, ballast, or other mission requirement such as releasing disposable sensors (Brandreth, Jr, 2000).

3. Literature Review

New applications of AI, which include Intelligent Agents are used to developing autonomous flight for UAV – Unmanned Aerial Vehicle. Later research efforts began to divide the many tasks involved in the control of UAV flight into manageable steps. Concentrated efforts of applying logic schemes, similar to Fuzzy Logic and Neural Networks, are being applied to improve the mathematical solutions for flight control (Dufrene, 2004).

Unmanned Autonomous Blimps – aerial robots that are controlled by computer based on information gathered by sensors – a kind of UAV offer several possibilities for original research (Coelho, and Campos, 1999),. One of them is the study of visual navigation for this kind of vehicle. Visual sensors could be necessary not only to perform data acquisition as part of the mission such as taking pictures of predefined spots in an environmental surveillance mission (Fukao, Fujitani, and Kanade, 2003), but they could also help on the autonomous navigation of the dirigible, supplying data to perform it in situations where more conventional, navigation techniques, like those using inertial and GPS (Hygounenc, and Souères, 2003). In addition, the research of cooperative control of multiple unmanned autonomous blimp poses significant theoretical and technical issues (Xia, and Corbett, 2004).

(Huff, Kamel, and Nygard, 2003) shows that an agent based framework can be used to model autonomous unmanned blimps. (DeLoach, Wood, and Sparkman, 2001) is about MaSE. MaSE use Cockburn's form of use case (Cockburn, 1997). (Booch, Rumbaugh, Jacobson, 2000) is about UML, and so is (Guedes G. T. A., 2004), this last one having a practical approach to the language. A methodology helps human being to act on a kind of system, so a methodology must adjust itself to the system, and to the human being; and human being capabilities and limitations must orient the design of the methodology. (Miller, 1956) shows our limitations for processing information – processing information is essential in analysis and design activities - and how bypasses them. How our limitations for processing information can be bypassed and why this must be done is the foundation of the top-down approach. A brief overview of modern blimps is presented in (Brandreth, Jr, 2000). A comparative analysis of aerial vehicles searching for the best environmental monitoring vehicle is done in (Elfes et al., 1998). (Gomes, and Ramos, 1998) presents a dynamic blimp model. We will use this paper in the design phase, nevertheless it subsidies us in analysis phase as it treats of the unmanned blimps applications.

4. Modeling a Set of Blimps

The modeling is comprised of four categories of interrelated systems: the *s*ystem of *u*nmanned *a*utonomous *b*lim*p*s (SUAB), the *u*nmanned *a*utonomous *b*limp (UAB) – a subsystem of SUAB –, the navigation subsystem of the SUABs and the navigation subsystem of the UAB. The SUAB consists of a set of UABs capable of acting autonomously in the pursuing of mission fulfillment. Each UAB of the SUAB operates autonomously and interacts cooperatively with the others by means of communication. Thus, the SUAB is composed by the UABs plus the interaction between them.

In the present case, there are two kinds of navigation system: the first one comprises the navigation subsystem of the SUAB and the second comprehends the navigation subsystems of the UABs. The SUAB navigation subsystem goal is to move (to take off, to land, to hover, to position) the set of UABs. To execute these movements, SUAB uses the individual capacities of navigation of each UAB, plus the intercommunication capabilities that each UAB possesses; therefore, the navigation subsystem of the SUAB directs the UAB navigation subsystems.

The navigation subsystem of the UAB has as objective to put into motion (to take off, to land, to hover, to position) a single UAB. For such, the system interacts with the sensors and actuators embarked in the UAB. Independently of the missions that are attributed to this subsystem, it permanently pursues goals such as auto preservation of the SUAB - by diverting it automatically from obstacles to prevent collisions and preservation of the telecommunication net - by maintaining the UAB in an adequate position in relation to the adjacent UABs in the telecommunication net.

4.1 SUAB Model

The model of the SUAB consists of several partial models produced during the phases of analysis and design of the system. These partial models focus on particular aspects of the system. A complete model of the system consists of all partial models put together. In MaSE, the partial models of the analysis phase are Goal Hierarchy Diagram, Use Cases, Sequence Diagram, Role Diagram and Concurrent Task Diagram – each one mentioned partial model is described after in the text. These models have been elaborated for the SUAB with the help from agentTool, and some of them, totally or partially, are shown in this text. The models of the project phase are been elaborated.

4.1.1. Goal Hierarchy

A Goal Hierarchy Diagram is a directed, acyclic graph where the nodes represent goals and arcs define a sub-goal relationship. A goal hierarchy is not a tree, since a goal may be a sub-goal of more than one parent goal. Figure 1 presents the SUAB Goal Hierarchy Diagram.

Two basic use cases can be derived from the goal hierarchy, one concerning monitoring environments and another one concerning providing access to the Internet. There are at least two alternatives of modeling for these two cases. On the first one, the operations that prepare the execution of the cases – as, by example, UABs positioning – are chosen, calculated, and started by an actor. On the second one, the two cases of use are extended by others, in such a way that the preliminary operations are chosen, calculated, and started by the own SUAB. The level of autonomy and the complexity of the SUAB are much higher on the second alternative than on the first one.

The first alternative is the one we will be developing. The first model is simpler than the second one, increasing the probability of the project being successful. Another reason for the choice for the first alternative is that a project that satisfies the requirements for the second one can be derived from a project that satisfies the first. That means that, by developing the first alternative, we have the bonus of being closer to achieving the goal of developing the second one.

When a goal does not have a corresponding role, its node in the Goal Hierarch Diagram is a gray box. When this happens, the goal is accomplished by its sub-goals. This procedure derives from the importance that MaSE gives to the fulfillment of the goals, assuring each one of them corresponds directly - or indirectly - to a role. Goals without roles are called partitioned goals. When a goal is a partitioned goal, only its sub-goals will have corresponding roles in the final system. In Figure 1, the partitioned goals are 1 Monitor Environment and Communicate Data, 1.2 Enable Internet User and 1.4 Position SUAB.

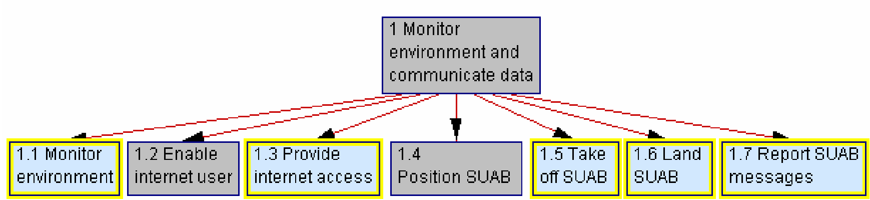

Figure 1. Goal Hierarchy Diagram

The names of the goals in the diagram are self-explanatory; however, it could be convenient to note that goal 1.2, Enable Internet User, is going to admit in the system an IP point that can demand communication with the Internet; and that goal 1.7, Report SUAB Messages, informs to the MCP, under request, control messages occurred inside the system, and messages interchanged with the MCP.

4.1.2. Use Cases

In MaSE, there is no use cases diagram (DeLoach, Wood, Sparkman, 2001). In MaSE, each use case is written in a structured narrative format (Cockburn, 1997). Nevertheless, we will use a graphic format for use cases, in UML style (Booch, Rumbaugh, Jacobson, 2000), to facilitate the understanding of the whole system. In MaSE, a use case encloses multiple scenarios. These scenarios are developed in sequence diagrams. Each use case is detailed and complemented by such sequence diagrams.

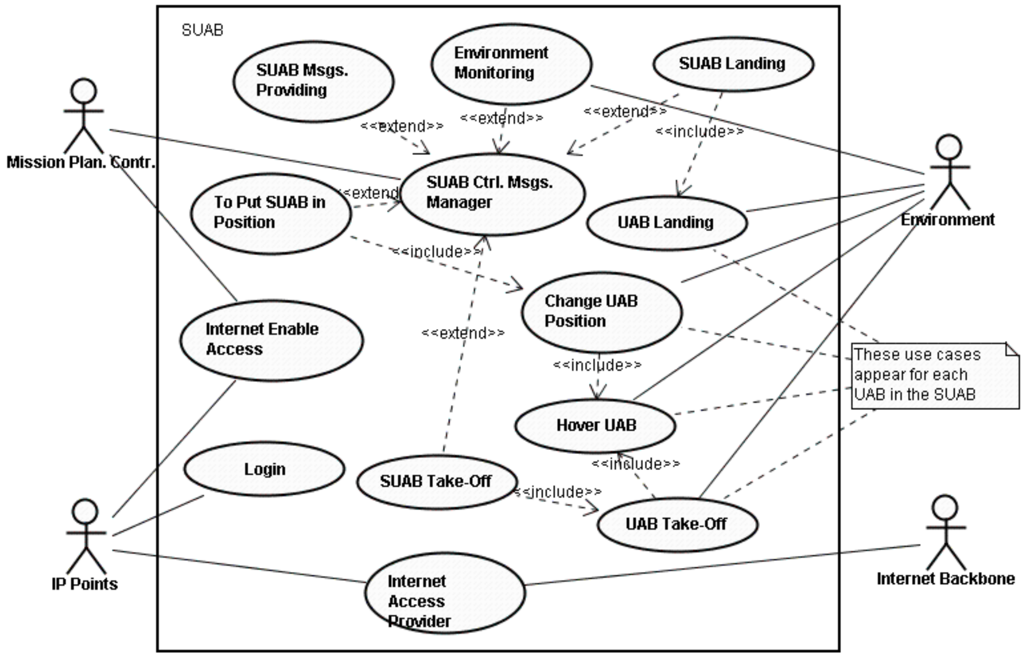

Figure 2. Use Cases Diagram

The primary actors (Cockburn, 1997) are the Mission Controller and Planner - MCP -, person or organization responsible for the definition and control of the SUAB missions and the computers requesting access to the Internet, here called collectively IP Points - see Figure 2. The MCP requests the actions the SUAB must perform, such as positioning the SUAB in specific coordinates, having the SUAB take off and land; receiving data from the system regarding the control messages exchanged by the primary actor with the system and controlling messages occurred inside the system. The messages supplied to the MCP enable obtaining information about the SUAB trajectory and about messages that have caused some type of imperfection in navigation. The other primary actor, IP Points, requests Internet access to the SUAB. It is necessary that the IP points be in an area covered by SUAB, so that system can provide access to them.

The SUAB has two secondary actors (Cockburn, 1997), as shown in the diagram of use cases: one is Internet Backbone and other is Environment. Environment is a secondary actor, which exposes subsets of its states for monitoring. This actor can also provide raw data of marked beings, such as by an animal with data transmitter device on it. The secondary actor, Internet Backbone, assists the SUAB connecting to the Internet.

SUAB has an inherent functionality that is its locomotion capacity. SUAB is a fleet of airships and also an autonomous fleet of UABs whose components act autonomously and cooperatively. Moreover, SUAB has two supplementary functionalities: the capacity to monitor environments and the capacity to provide access to the Internet. These functional extensions are accomplished with the assistance of the capacity of locomotion of the fleet. During the locomotion, the SUAB must keep the UABs intercommunication and the SUAB-MCP intercommunication functioning.

The movement use cases include two levels, the SUAB and the UAB. The UAB deals with the individual movement of each UAB. The use cases of the SUAB level are helped by the use cases of the UAB level to perform the movement operations. For example, the SUAB Land use case is extended by UAB Land use cases. Actually, this means that SUAB Land use case drives the land of each UAB.

The SUAB Control Messages Manager use case manages control messages of the system. These messages occur inside the SUAB and between SUAB and MCP. That use case maintains the fleet logbook.

MaSE uses narrative format to describe use cases, but no defines one to use. Cockburn suggests frames to the narrative format (Cockburn, 1997). We decide to use a narrative format to describe the each use case based in Guedes (Guedes, 2004). The narrative format, which we decide for, is in accordance with the Cockburn's format frame.

In Table 1 and Table 2, two illustrative SUAB use cases are described. The Monitoring Environment use case defines the UABs behavior, when the UABs are monitoring an area. In the Table 1, there is a description of this use case. In the table, there is also the single sequence of actions that produces the results of the use case. This sequence is a sketch of the actual sequence in the implemented system – several execution sequences can correspond to a single one use case, but does not happen always. As we can see in the Table 1, there is a one single sequence for this use case.

Table 1. Environment Monitoring

Use Case	Environment Monitoring
Goal	1.1.
General Use Case	
Primary Actor	MCP.
Secondary Actors	Environment.
Description	The monitoring UAB of minimal cost is chosen, the monitoring resources of this UAB are initialized, the communication of the MCP with the resources is established, and the control of the resources is passed to the MCP. Among the control commands, MCP can use the monitoring session close. This control command releases the resources for use by other applications.
Preconditions	The SUAB must be hovering over the area to be monitored.
Post-condition	
Primary Actor Actions	**System Actions**
1 – Request monitorial services.	
	2 – Select minimal cost UAB.
	3 – Initialize monitoring resources.
	4 – Confirm UAB resources activation.
	5 – Enable resources control by MCP.
6 – Close monitoring session.	
	7 – Disable resources control.
	8 – Turn off resources.
	9 – Confirm monitoring session close.
Fail Condition	(1) SUAB cannot hover over an area; (2) The data communication link is not enabled; (3) The monitoring equipment does not send data.
Note	The accomplishment of precondition requires that MCP calculates the coordinates of each UAB.

There are several cost factors as, e.g., the distance of the UAB to the area to be monitored; the quantity of fuel in the UAB; and the existence, or not, of monitoring resources in the UAB.

The Table 2 contains the description of the To Put SUAB in Position use case.

Table 2. To Put SUAB in Position

Use case	To Put SUAB in Position
Goal	1.4
General Use Case	
Primary Actor	MCP.
Secondary Actors	
Description	This use case directs the actions concerning put UABs in position. These actions obey the steps to obtain initial SUAB position, to plan trajectory of SUAB, to move SUAB along the planned trajectory, and to hover on final coordinates.
Preconditions	The SUAB must have took-off.
Post-condition	
Primary Actor Actions	**System Actions**
1 – Ask for put SUAB in position.	
	2 – Plan trajectory each UAB.
	3 – Put each UAB in position.
	4 – Confirm the final position of each UAB.
	5 – Inform to MCP the final SUAB position.
Fail Condition	(1) SUAB did not take-off; (2) SUAB does not maintain UABs intercommunication.
Note	

4.1.3. Sequence Diagram

A sequence diagram depicts the sequence of messages (or events) that are transmitted between roles identified from use cases as shown in Figure 3. The boxes at the top of the diagram represent system roles and the arrows between the lines represent events passed between roles. Time is assumed to flow from the top of the diagram to the bottom.

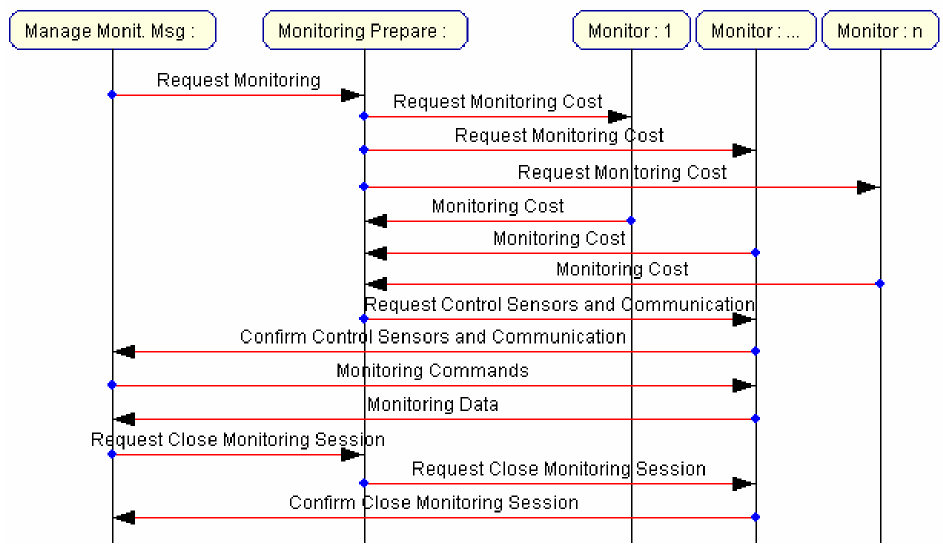

Figure 3. Environment Monitoring Sequence Diagram

Figure 3 concerns to the Environment Monitoring sequence of Environment Monitoring use case.

The sequence diagram in the Figure 3 details the sequence of events for a monitoring to be carried out. In the diagram, instances of the following roles are represented: Monitor, Prepare Monitoring, and Manage Monitoring Messages. Several instances of the Monitor role are represented in the diagram. These instances calculate the corresponding UAB cost to monitor an area – under request of the Prepare Monitoring role instance. The Prepare Monitoring role instance chooses the UAB agent of smaller cost to monitor the area. The Monitor role instance that was chosen by Prepare Monitoring role instance acts like an interface between the MCP and the sensorial resources of the UAB. The Monitor role instance chosen has a number of responsibilities. We intend, e.g., this Monitor role instance receives environment monitoring control messages, decodes them, and activates properly the environment monitoring devices of the corresponding UAB.

4.1.4. Role Diagram

The functional requirements in MaSE are abstracted as system structured goals. Role is a system entity responsible for to fulfill, or to help to fulfill, the system goals. Therefore, roles are derived from functional requirements, since goals are derived from functional requirements, and roles are derived from goals. This scheme contributes to system effectiveness. A role is constituted of tasks. Tasks are implemented in different threads, so tasks are concurrent.

Figure 4 is a fragment of the system role diagram. In a role diagram, rectangles represent roles, and numbers into the rectangles represent goals or sub-goals. A role consists of tasks. Oval shapes attached to a role represent tasks of that role. Lines (possibly named) between tasks denote communication protocols that occur between the tasks. The arrows denote the initiator/responder relationship of the protocol with the arrow pointing from the initiator to the respondent. Solid lines indicate peer-to-peer communication, which are generally implemented as external communication protocols. External protocols involve message passing between roles that may become actual messages if their roles end up being implemented in separate agents. Dashed lines denote communication between concurrent tasks within the same role.

The role diagram fragment in Figure 4 involves monitoring roles, i.e., the roles responsible for the 1.1 goal, monitor environment, and its sub-goals. These roles are Monitor, Prepare Monitoring and Manage Monitoring Messages. Monitor role regards goal 1.1, Monitor Environment; Manage Monitoring Messages role regards sub-goal 1.1.1, Deal with Monitoring Control Messages; and Prepare the Monitor role, regards sub-goal 1.1.2, Set SUAB to Monitoring. The tasks of Monitor role are Monitor, Calculate Monitor Cost, and Activate Monitoring Resources tasks. Request Monitoring Cost is a protocol that initiates at Distribute Monitoring Control Messages task of the Preparing Monitoring role, and ends at Calculate Monitoring Cost task of the Monitor role. The line of this protocol is solid because the tasks are in different roles.

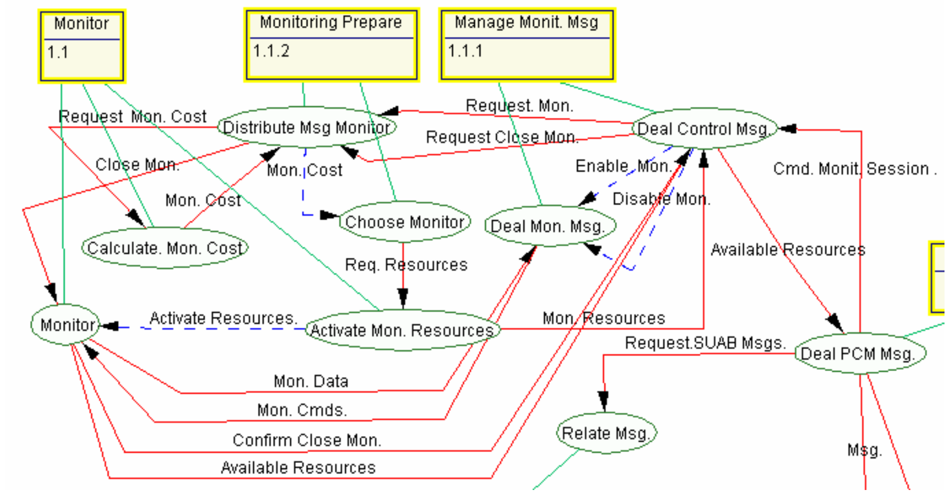

Figure 4. Role diagram fragment regarding to Monitoring Environment

4.1.5. Task Diagram

Task diagram is the last model of the MaSE Analysis Phase. As the SUAB is directed by event, a model of process transition is essential. In MaSE, this model, called Concurrent Task Diagram, uses a finite state automaton model. Concurrent tasks consist of states and transitions. States encompass the processing that goes on internal to the agent while transitions allow communication between agents or between tasks. We are going to show the Deal Control Message task (Figure 5), as an example of SUAB task diagrams.

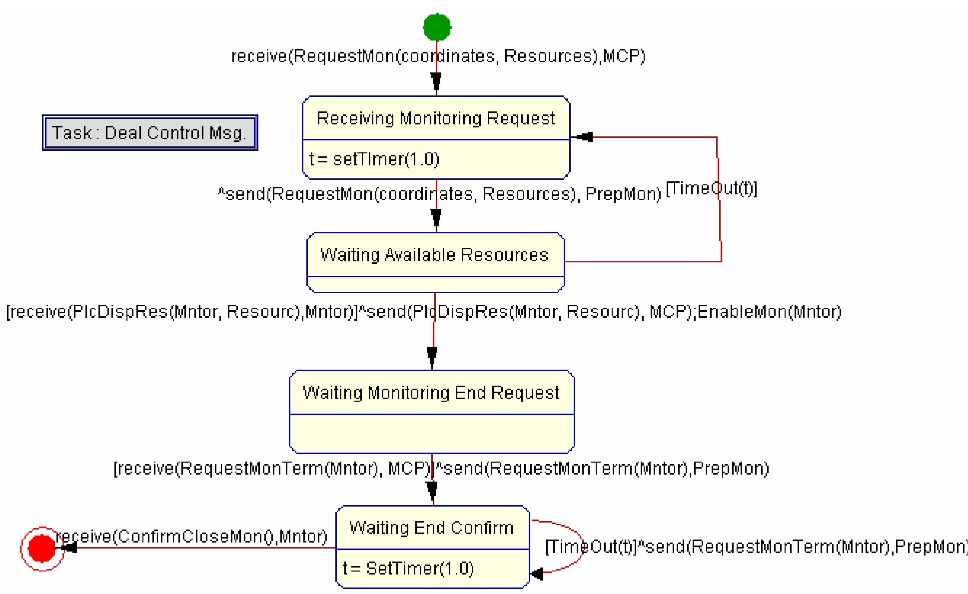

Figure 5. Diagram of Deal Control Message Task

The task is initiated upon receipt of a Request Monitoring message from MCP role – MCP is abstracted as a role. After the message is received, the task goes to the Receiving Monitoring Request state. In this state, the task sets the timer, sends the message received to Prepare Monitoring – this message communicates the coordinates of the monitoring area, and the monitoring resources requested. After this, the task goes to Waiting Available Resources state. The task goes out this state or receiving a message from the Monitor role, or by time out. In the first case, the task goes to Waiting Monitoring End Request when receives from Monitor role the Place Resource at Disposal message. During this state transition, the task transfers the message received from Monitor role to MCP role. During the monitoring, the task stays at Waiting Monitoring End Request state. The MCP monitoring ending request changes the task state to Waiting End Confirm state. In this transition, the task sends an ending monitoring message to Prepare Monitor role. In the Waiting End Confirmation state, the task resends the ending monitoring message at each second, until the monitor confirms the end of the monitoring operation.

5. Conclusion and Future Work

The purpose of the work associated to this paper is to build a cheap and reliable aerial system for monitoring remote and very large areas. Accordingly, the most appropriate aerial vehicle was selected, the premises of the model were established, and the methodology best suited to the model premises was chosen. The several models of the analysis phase of the methodology were produced, and some of them were presented here. The analysis phase purpose in MaSE is to produce a set of roles, which is the basis to the design phase. At end of the design phase, the agent architecture will be defined. In our case, architectural requirements impose each UAB be an agent, or agent container, so we are actually defining what goes into each UAB. The subsequent work will firstly produce a simulation model runnig over JADE plataforms. After this, a refined UAB specification, including hardware, software, sensors, actuators and navigational devices, will be generated. Eventually, a physical and operational SUAB will be constructed..

6. References

Booch, G., Rumbaugh, J., Jacobson, I., 2000, "UML – Guia do Usuário", Elsevier Editora Ltda.

Brandreth, Jr, E. J., 2000, "Airships: An Ideal Platform For Human Or Remote Sensing In The Marine Environment", IEEE 2000, pp 1883-1885.

Cockburn, A., 1997, "Structuring Use Cases with Goals", Journal of Object-Oriented Programming, Sep-Oct, 1997, and Nov-Dec, 1997.

Coelho, L. S., Campos, M. F. M., 1999, "Pose Estimation of Autonomous Dirigibles Using Landmarks" XII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '99), October 17-20, 1999, Campinas, Brazil.

DeLoach, S. A., Wood, M. F., and Sparkman C. H., 2001, "Multiagent Systems Engineering", International Journal of Software Engineering and Knowledge Engineering. Vol. 11, No. 3 2001 pp 231-258.

Dufrene, W. R., 2004, "AI Techniques in Uninhabited Aerial Vehicle Flight ", IEEE A&E Systems Magazine, August, 2004.

Elfes, A., Bueno, S. S., Bergerman, M., Ramos, J. J. G., 1998, "A Semi-Autonomous Robotic Airship for Environmental Monitoring Missions", Proceedings of the 1998 IEEE, International Conference on Robotics & Automation, Leuven, Belgium, May 1998, pp 3449-3455.

Fukao, T., Fujitani, K., Kanade, T., 2003, "An Autonomous Blimp for a Surveillance System", Proceedings of the 2003 IEEE/RSJ, Conference on Intelligent Robots and Systems, Las Vegas, Nevada, U.S.A., 2003, pp 1820 – 1825.

Caire, Giovanni, 2003, "JADE Tutorial – JADE Programming for Beginners, TILAB, formerly CSELT.

Gomes, S.B.V. and Ramos, J.J.G., 1998, "Airship Dynamic Modeling for Autonomous Operation", Proceedings of the 1998 IEEE, International Conference on Robotics & Automation, Leuven, Belgium, May 1998, pp 3462-3467.

Guedes G. T. A., 2004, "UML Uma Abordagem Prática", Novatec Editora.

Huff, N., Kamel, A., and Nygard, K., 2003, "An Agent Based Framework for Modeling UAV's", ISCA 16th International Conference on Computer Applications in Industry and Engineering, Las Vegas, Nevada. November 2003, pp 139-144.

Hygounenc, E., Souères, P., 2003, "Lateral Path Following GPS-based Control of a Small-Size Unmanned Blimp", Proceedings of the 2003 IEEE, International Conference on Robotics & Automation, Taipei, Taiwan, September 14-19, 2003, pp 540-545.

Miller, G. A., 1956, "The Magical Number Seven, Plus Minus Two: Some Limits on Our Capacity for Processing Information", The Psychological Review, 1956.

Xia, G., Corbett, D. R., 2004, "Cooperative Control Systems of Searching Targets Using Unmanned Blimps", Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, P.R., China, June 15-19, 2004, pp 1179-1183.

7. Responsibility notice

The authors are the only responsible for the printed material included in this paper.