The INGENIAS Methodology for Advanced Surveillance Systems Modelling

José M. Gascueña[1] and Antonio Fernández-Caballero[1,2]

[1] Instituto de Investigación en Informática de Albacete (I3A)
[2] Escuela Politécnica Superior de Albacete
Universidad de Castilla-La Mancha, 02071-Albacete, Spain
{jmanuel,caballer}@dsi.uclm.es

Abstract. The use of surveillance systems has grown exponentially during the last decade. Moreover, the agency paradigm has shown to be suitable for the design and development of complex systems such as surveillance systems. They provide autonomy, reactivity, social ability and pro-activeness to carry out surveillance tasks in a semi-automatic way, collaborating with users in a more effective manner. Agents provide coordination mechanisms, solve conflicts, and determine through negotiation processes the more appropriate distribution of the surveillance tasks. Existent agent-based surveillance systems do not really use agent-based methodologies to develop them. In this paper, our experience for modelling advanced surveillance systems using the INGENIAS methodology is described.

1 Introduction

The use of surveillance systems has grown exponentially during the last decade, and has been applied in many different environments [7]. These systems are focused on (1) monitoring the environment, (2) informing of detected pre-alarm situations, and (3) performing the most suitable actions, in collaboration with the users, when a real alert situation is confirmed. Nowadays, there is also an increasing interest in researchers in integrating some capabilities to interpret scenes, that is to say, to understand dynamic situations and events that appear in the global scene. The purpose is to inform of the presence of certain previously defined situations, which go one step further than simply detecting and tracking the objects. This is precisely called advanced surveillance.

On the other hand, one important aim of multi-agent systems (MAS) is to provide a high-level abstraction software engineering paradigm for designing and developing complex software systems [14]. Advanced surveillance systems are precisely a good example of it, as they include elements that coordinate the execution of the activities for solving sufficiently complex tasks and achieving the outlined goals. There are proposals for several methodologies that allow facing the development of complex systems. Remember, for example, MAS-CommonKADS, ZEUS, Gaia, MaSE, or Prometheus, among others. In this article we describe our experience for modelling an advanced surveillance system using the INGENIAS methodology [9].

J. Mira and J.R. Álvarez (Eds.): IWINAC 2007, Part II, LNCS 4528, pp. 541–550, 2007.

2 Multi-agents in Surveillance Systems

Franklin and Graesser [4] state: *"an autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future"*. According to this definition, any agent should satisfy four properties: autonomy, social ability, reactivity and pro-activeness. Usually one single agent does not solve problems by itself. Generally the cooperation between various agents is essential to tackle a complex problem more easily. In this kind of process probably the agents ask and provide data, request to carry out actions, enter into negotiations, etc.

A MAS is formed by a set of agents possessing the previously cited properties. Thus, in a visual surveillance system, the software that manages the operation of a camera group might easily be modelled as a multi-agent system because it satisfies the properties. For instance, each camera could be associated to an agent that controls it, that is to say, it is monitoring what is happening in its field of view (autonomy), it reacts to changes that take place in the scene (reactivity), it may communicate with agents associated to others cameras to obtain additional information about a common target (social ability), and it may rotate the camera with the aim of looking for new objects (pro-activeness). In this example it is necessary to coordinate a set of cameras in order to solve conflicts that may appear, and to determine through negotiation processes the more appropriate distribution of tasks aimed to locate and track at any moment the targets. These problems have widely been studied in the area of agents; this is another powerful reason that justifies their use in surveillance systems. This example can be extended to even more complex multi-sensorial surveillance systems [8], which also include non-visual sensors (e.g., temperature, infrared or biometric sensors) and cameras mounted on moving robots. Consider also that in order to control a surveillance system a decentralized architecture is preferred to a centralized one, as this allows to obtain a more robust, scalable, flexible and fault tolerant system. Multi-agent systems are a good solution to face more easily the complex problem of the distribution because they provide a natural way of solving problems that are inherently distributed [14].

Due to all these reasons, in the last years there have been several efforts to create agent-based visual surveillance systems. There is the tracking system proposed in [12], where each moving object equipped with a Global Positioning System (GPS) is tracked by an agent. Also in [13] a group of active vision agents, whose fields of view may be overlapped, have the capability to track in real time multiple simultaneously moving objects, whereas the traffic surveillance system Monitorix uses non overlapped cameras [1]. A framework to understand the dynamics of a scene have also been introduced so far. There is the proposal by Remagino and colleagues [10] based on the cooperation between agents, where persons and vehicles appear entering or leaving an area; and the architecture shown in [2] uses wireless visual sensors, region agents searching for objects of interest (persons, vehicles and animals) and object agents tracking them to determine if the scene contains a threat pattern, and performing a appropriate action. On the other hand, robots and other sensors can also be used for monitoring and security tasks. For example, robot and sensors controlled by agents have been used for fire detection [11], [6] or in rescue operations [5]. Our critique to the approaches cited previously is that there is a lack of methodological development of the multi-agent systems. Our concern is that using a methodology is absolutely useful as it offers solutions on techniques, guidelines, and tools necessary during the life cycle stages of any system under development.

3 MAS Description Using the INGENIAS Methodology

Our proposal is based on INGENIAS methodology for several reasons: (a) it is based in a previously tested MESSAGE methodology [3], (b) it is largely documented [9], (c) useful tools that support the analysis, design and code generation of based-agent software are included in the INGENIAS Development Kit, (d) it uses a model-driven approach that facilitates the independence from the implementation platform, (e) it is not oriented towards a particular agent platform, and (f) it has been validated in real applications (e.g. information access system, tourist guide services using mobile devices, scheduling of a ceramic tile factory production process, or knowledge management). The models created using INGENIAS are based on five meta-models which define different views and concepts to describe the agency: organization, agent, goals and tasks, interaction, and environment. Some elements of the agency may appear in different meta-models. This repetition of entities across different views induces dependencies among meta-models.

3.1 System Requirements

In our approach the system does not replace the human personnel, but it has to serve as his complement to jointly perform surveillance and security tasks. The user of a surveillance system should be informed of relevant events; this way, a semi-automatic system helps in the arduous task of permanently looking at the screens. Figure 1 shows the functionalities offered to the security guards: watching the observed environment and informing about the interactions produced

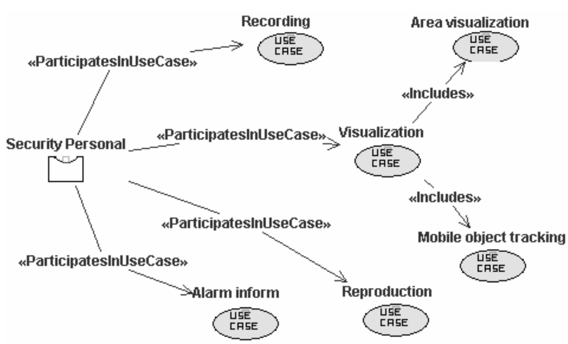

Fig. 1. Functional requirements for the security guard

between monitored targets (*area visualization* use case), tracking the selected targets (*mobile object tracking* use case), recording video sequences (*recording* use case) to reproduce later on (*reproduction* use case), as well informing about detected alarm situations (*alarm inform* use case). After identifying the system functional requirements we have performed the decomposition into agents in order to carry out the surveillance system functions.

3.2 Organization Model

In a multi-agent system modelled in INGENIAS the organization model is the equivalent to the system architecture. It defines how the agents are grouped, the system functionality and which restrictions are imposed to the agents' behaviour. From a structural point of view, the organization is a set of entities with relationship of aggregation and inheritance. It defines a schema where agents, resources, tasks and goals may exist. In this model, groups may be used to decompose the organization, plans, and workflows to establish the way the resources are assigned, which tasks are necessary to achieve a goal, and who has the responsibility of carrying them out.

Thus, the scheme of the organization shown in figure 2 describes the global system architecture. Notice that agents belonging to groups that gather information of the monitored place (*CameraGroup*, *RobotGroup*, *OtherSensorGroup* groups) are distinguished. These agents send the information to agents included in other groups that have the functionality of showing information to the user about what is happening in the monitored environment (*DeviceMobileGroup*, *BuildingCentral*, *PrincipalCentral* groups). The surveillance system, represented as organization *Surveillance System*, has the goals to monitor the environment, to inform about suspicious events, to interpret what is happening in a scene and to handle alarms that are fired (all represented in form of circles).

The Camera group incorporates a series of different agents. An Image agent (*AImage*) captures the image of the environment, does a pre-processing and sends the results to the Blob agent (*ABlob*). This one is in charge of identifying image regions that correspond to objects of interest (vehicles, people, animals, and so on). An Object agent (*AObject*) is responsible for organizing the information associated to the objects of interest and to classify them. An Activity agent (*AActivity*) describes the activities of the objects identified in the scene, whereas a Social agent (*ASocial*) describes the interaction between objects. A Camera Manager agent (*AManager*) communicates with the previous agents, and may establish a communication with a mobile device, a Building Alarm System or Central Alarm System agents (*APDA*, *AMPhone*, *ABCentral*, *APCentral*). Moreover, in the Robot group there is an agent that controls the robot (*ARobot*) whose goals are to obtain a vision of the objects of interest from a different perspective, and to accede to zones that are inaccessible to fixed cameras or that are dangerous for the humans. Finally, the administration of the information provided by non visual sensors is carried out by the corresponding associated agents (*temperature*, *volumetric* agents, among others).

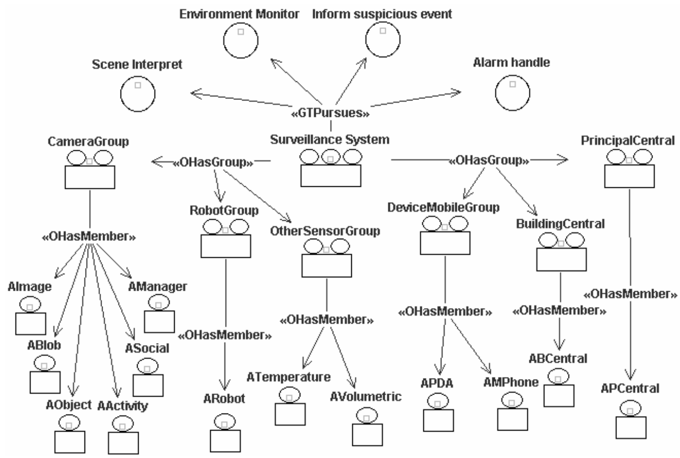

Fig. 2. Representation of the surveillance system organization

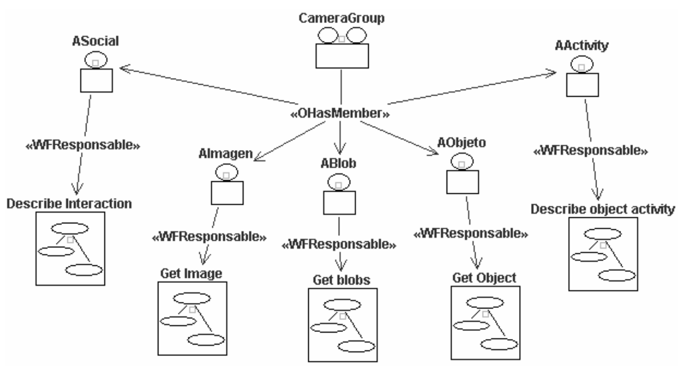

Fig. 3. Plans of the agents belonging to Camera group

The security guard may be equipped with a mobile device (PDA, mobile phone) wirelessly connected to the surveillance environment. This way he can be informed while he freely walks around the monitored indoor or outdoor zone. Therefore, we consider an agent that is working in a mobile device receiving information from the manager, the robot or other sensor agents. Moreover, this agent interacts with the security guard to alert him of detected suspicious events, to send images or any other required information.

In the INGENIAS-based organization model, plans realizing tasks can also be included. As an example, figure 3 shows some plans for the agents belonging to the Camera group. Plans define associations among tasks and general information about their execution. Thus, for instance, consider figure 4 where the tasks execution sequence associated with plan *"Get blobs"* is shown. Its goal is to segment the input image, represented by a circle, in order to obtain the blobs (moving patches) that appear in an image. It is decomposed into two

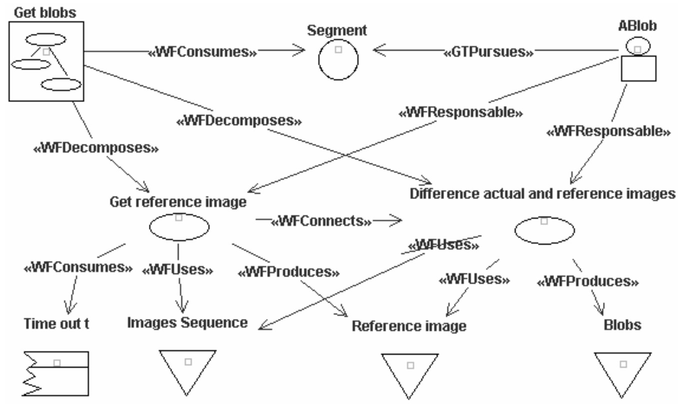

Fig. 4. Representation of plan *Get blobs*

tasks, and a *Blob* agent is responsible for its execution. Task *"Get reference image"* requires an image sequence (resource represented as a triangle) to produce a reference image that is consumed by task *"Difference among current and reference image"* to produce blobs (resource *blobs*) extracted from the current image. The reference image is gotten each certain time; this is indicated by event *Time out t*.

3.3 Agent Model

In INGENIAS, particular agents are defined in the agent model, excluding the interactions with other agents. This model is centred in defining the tasks to execute and the goals to pursue, that is to say, the agent functionality (associations of plans, tasks, goals and roles with agents), and it defines which mechanisms ensure the tasks execution within the decided parameters, that is to say, its control design (type of control, mental state specification and its evolution).

For each agent modelled previously in the organization model, an agent model describing it is created. Figure 5 shows a graphical representation of the Image agent model. It is responsible for acquiring images and pre-processing them (*get image* and *image pre-processing* goals). It possesses the capability to carry out tasks in order to achieve these goals. Moreover, it has a mental state manager (*MSManagerAImage*) with the aim of developing the mental state evolution through operations "create, destroy, modify and monitor agent knowledge"; and a mental state processor (*MSProcessorAImage*) to make decisions of which task has to be executed (agent control). It contains an initial mental state representing the camera initial position (fact *Camera initial position*). Finally, through internal application *APICamera* it can turn left, right, up, or down its associated physical camera, and apply zoom in order to modify its field of view.

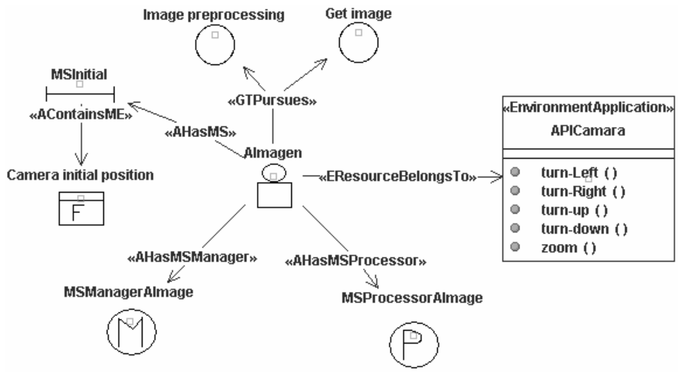

Fig. 5. AImage agent model

3.4 Goals and Tasks Model

The goals and tasks model has the purpose of gathering the motivations of the multi-agent system, to define actions identified in the organization, interactions or agents models, and how these actions affect his responsible. It also expresses which consequences have to execute tasks and why they should be executed. Diagrams from this view can be used to explain how the mental state processor and manager work.

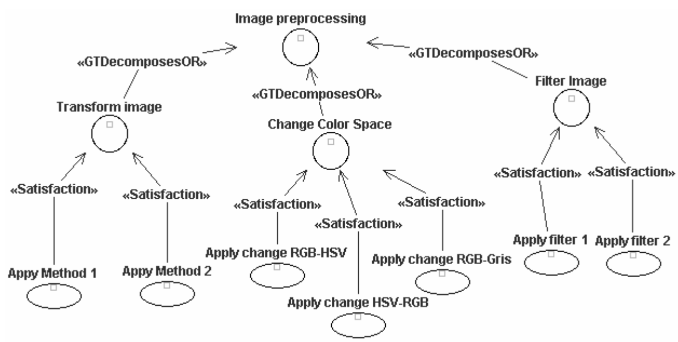

Fig. 6. Goals and tasks model

For example, goal *"Image pre-processing"* may be decomposed into three goals as shown in figure 6: (1) Transform image, (2) Filter image, and (3) Change colour space. It disposes of several tasks to achieve each sub-goal, but not all of them have always to be executed; this depends on the scenario where the camera is placed and the proper camera characteristics, among other features.

3.5 Interaction Model

In the interaction model there are elements such as agents, roles, goals, interactions and interaction units. It offers details about the way the agents communicate

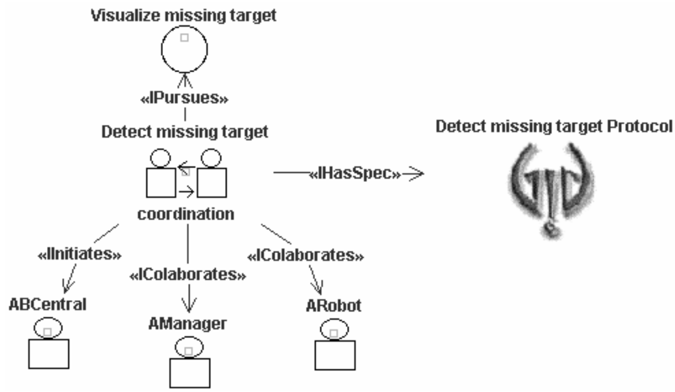

Fig. 7. Interaction *"Detect target missing"*

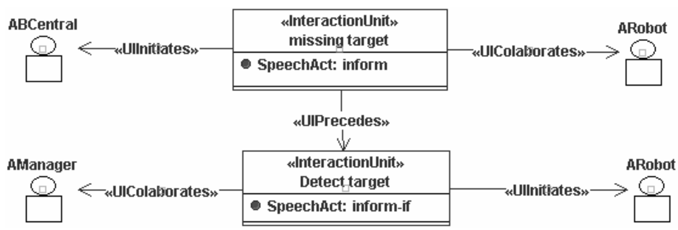

Fig. 8. Protocol *"Detect missing target"*

and coordinate. The interaction model addresses the exchange of information or requests between agents, or between agents and users. Indeed, some tasks require the collaboration of other agents to carry them out. Interaction units are performed in the interactions where there is one initiator and several collaborators. Moreover, the participation of the actors in the interaction and the existence of the interaction itself are justified through goals.

For example, figure 7 shows the interaction *"Detect missing target"*. In a certain moment a possible target can be outside the field of view of the installed cameras; so this fact is communicated to a Robot agent. The Robot agent co-operates with the Camera Manager agent responsible of controlling the camera installed on the robot in order to detect and visualize the target again. In first place, figure 8 describes protocol *"Detect missing target"*, where a Central Building agent (*ABCentral*) requests the Robot agent to start navigating in order to detect the missing target (*missing target* unit interaction). Next, the Robot agent asks the Camera Manager agent if it has to visualize the target looked for (*detected target* unit interaction). Through UIPrecedes relation we indicate that the *missing target* unit interaction precedes the *detected target* unit interaction.

3.6 Environment Model

The environment model addresses what surrounds the system modelled in INGENIAS. The environment only includes agents, resources (e.g., required number of processes or threads, number of necessary connections with data bases) and applications whose main utility is to express perceptions and actions of the agents. Agents act on the environment by invoking the methods or procedures defined in the applications, and they define their perceptions indicating which events are produced by the applications they listen to.

A part of the environment model of a surveillance system is shown in figure 9. The Robot agent is associated an external application named *RobotApplication*. This application controls the robot by providing operations drive straight, left, right, specifying to advance forward, back, at a given speed, etc. In the same way, sensors are associated some Application Programming Interfaces (API) that can be used by agents to carry out their tasks.

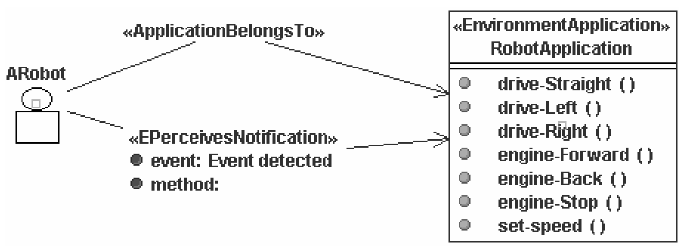

Fig. 9. Environment model

4 Conclusions

Agents offer characteristics such as autonomy to reach a given objective, communication between agents, and capabilities to react before changes in the environment, as well as to take the initiative to perform certain tasks. Therefore, agent-based surveillance systems are essential to assist users in a semi-automatic way in target detection and monitoring tasks. Multi-agent systems provide the necessary foundations to reduce the complexity, to increase the flexibility, scalability and the necessary tolerance to failures in a surveillance system, and they are a natural way of solving distributed problems.

To date, the modelling of agent-based surveillance systems is usually carried out empirically, without using a methodology for the development of the agents. We have studied and proven the adequacy of using the INGENIAS methodology for the analysis and design of such a complex application as an advanced surveillance system.

Acknowledgments

This work is supported in part by the Junta de Comunidades de Castilla-La Mancha PBI06-0099 grant and the Spanish TIN2004-07661-C02-02 grant. The authors are thankful to Dr. Juan Pavón and Dr. Jorge Gómez-Sanz of the Universidad Complutense de Madrid, Spain, for answering to all their queries concerning the INGENIAS methodology.

References

1. Abreu, B., Botelho, L., Cavallaro, A., Douxchamps, D., Ebrahimi, T., Figueiredo, P., Macq, B., Mory, B., Nunes, L., Orri, J., Trigueiros, M.J., Violante A.: Video-based multi-agent traffic surveillance system. Proceedings of the IIEE Intelligent Vehicles Symposium, IV2000 (2000) 457-462.
2. Aguilar-Ponce, R., Kumar, A, Tecpanectl-Xihuitl, J.L., and Magdy Bayoumi.: A network of sensors based framework for automated visual surveillance. Journal of Networks and Computer Applications (2007) to appear.
3. Caire, G., Leal, F., Chainho, P., Evans, R., Garijo, F., Gómez-Sanz, J.J., Pavón, J., Kearny, P., Stark, J., Massonet, P.: Agent-oriented analysis using MESSAGE/UML. Lecture Notes in Computer Science 2222 (2001) 119-135.
4. Franklin, S., Graesser, A.: Is it an agent, or just a program?: A taxonomy for autonomous agents. Lecture Notes in Computer Science 1193 (1996) 21-35.
5. Farinelli, A., Grisetti, G., Iocchi, L., Lo Cascio, S., Nardi, D.: Design and evaluation of multi agent systems for rescue operations. Proceedings of the International Conference on Intelligent Robots and Systems, IROS 2003 (2003) 3138-3143.
6. Fok, Ch.L., Roman, G.C., Lu, Ch.: Agilla: A Mobile Agent Middleware for Sensor Networks. Technical Report, Washington University in St. Louis, WUCSE-2006-16 (2006).
7. Patricio, M.A., Carbó, J., Pérez, O., García, J., Molina, J.M.: Multi-agent framework in visual sensor networks. EURASIP Journal on Advances in Signal Processing (2007) Article ID 98639.
8. Pavón J., Gómez-Sanz J., Fernández-Caballero A. & Valencia-Jiménez J.J. Development of intelligent multi-sensor surveillance systems with agents. Submitted, Robotics and Autonomous Systems (2007).
9. Pavón, J., Gómez-Sanz, J.J., Fuentes, R. The INGENIAS methodology and tools. In: Agent-Oriented Methodologies. Idea Group Publishing (2005).
10. Remagnino, P., Shihab, A.I., Jones, G.A.: Distributed intelligence for multi-camera visual surveillance. Pattern Recognition 37:4 (2004) 675-689.
11. Roman-Ballesteros, I., Pfeiffer, C.F.: A framework for cooperative multi-robot surveillance tasks. Proceedings of the Electronics, Robotics and Automotive Mechanics Conference, CERMA'06 (2006) 163-170.
12. Shakshuki E., Wang, Y.: Using agent-based approach to tracking moving objects. Proceedings of the 17th International Conference on Advanced Information Networking and Applications, ANIA'03 (2003) 578-481.
13. Ukita, N., Matsuyama, T.: Real-time cooperative multi-target tracking by communicating active vision agents. Computer Vision and Image Understanding 97:2 (2005) 137-179.
14. Weiss, G.: Multiagent Systems: A Modern Approach to Distribuited Artificial Intelligence. The MIT Press (2000).