Multi-agent Based Distributed Semi-automatic Sensors Surveillance System Architecture

Jesús Tejedor, Miguel A. Patricio, and José M. Molina

Applied Artificial Intelligence Group, Universidad Carlos III de Madrid,
Avd. de la Universidad 22., 28270 Colmenarejo, Spain
e-mail: {jtejedor,mpatrici}@inf.uc3m.es, molina@ia.uc3m.es

Abstract. In the present paper, we describes a semi-automated and decision support sensor surveillance architecture used to develop an intelligent sensor surveillance system. The proposed architecture is grouped in three agents layers: the sensors agents layer, sensor processing agents layer and finally, the support assistant agents layers. The sensor agents layer is formed by sensor managing agents and sensor data flow agents that they control the sensor devices and retransmit data streams to upper layer respectively. In sensor processing agents layer is an agents collection that process data flows produced by sensors, allowing elements tracking. The last layer is formed by special agents for helping and supporting the user monitoring and user choice. This architecture proposes a fully decentralized multi-agent system using FIPA Agent Communication Language.

Keywords: Multi-Agent System, Sensor Surveillance, Monitoring and Control, Agent, System Architectures, User Decision Support.

1 Introduction

This proposed architecture is a surveillance system that integrates sensor analysis and agent technology. The architecture is projected for outdoor conditions where is necessary tracking and processing elements (like land vehicles, ships) through many sensor (especially video cameras). The main goal is to coordinate video sensor and other sensor types (like radar, global positioning system, thermal cameras, etc) to tracking these desired targets and facilitates user operations, for example to maintain a constant vision frontier with minimum overlap. Valera Espina and Velastin had written a concise review of totally automated visual surveillance systems [1].

Human monitoring in surveillance functions is expensive and quite ineffective because each sensor provides a huge quantity of information [2]. Even trained users would lose concentration and miss a great amount of critical events in minimum lapses. Therefore, surveillance operators should be helping by support automatic assistant, filtering data to provide only most relevant information to user. Even, replace them exclusively by software systems [1].

Some authors like Henry Detmold propose surveillance middleware architecture, based on service-oriented architecture (SOA) [3] for all computational tasks. The SOA inconvenience lies in data re-encapsulation by simple object access protocol, not allowing a fully and correct communications between agents and data flows.

In this paper, we proposed a multi-agent architecture with two flows: The first flow, to maintain sensor controlling and managing (such as movement, focus or zoom in cameras, change parameters in radars, etc) through coordination between agents, and another flow to maintain a visual stream for the system users. The interaction between agents is controlled by a BDI - like architecture [8].

The coordination between agents would be based in FIPA ACL message [4] for keep flexibility and adaptability that they are desired conditions to system. At the same time ACL is a communication language based on the speech-act theory [5] and used in communications between last layers.

In section 2 we explain optimal conditions that system requires and mains goals that the system complains. Section 3 is focused on the architecture itself and we describe the different agents and their functionalities. Section 4 presents our conclusions.

2 System Characteristics

The perfect desired architecture must group not only essential surveillance requirements, but also complain other requests such as integration, net security, scalability, availability, flexibility and intelligent image processing.

This architecture is:

— Open, any agent can be adding to system thanks to standard language to communicate between agents.
— Flexible, the system can incorporate new functionalities and layers without changing previous work
— Scalable, increase the number of sensors or agents has a linear growth in computational complexity and an agent does not often communicate with all other agents.

The system should compute a huge data from different sensors types. Also can manage all sensors to coordinate them avoiding overlap, allowing a correct tracking and showing most important sensor signals to user. These are system main goals.

3 System Architecture

We have designed an open and generic multi-layer architecture for semi-automatic surveillance systems.

The first layer of our architecture called Devices Layer, it is formed by a set of agents that interact with the sensors and monitors. Each agent (especially sensor device agents) delivers output and receive requests for changes their parameters its
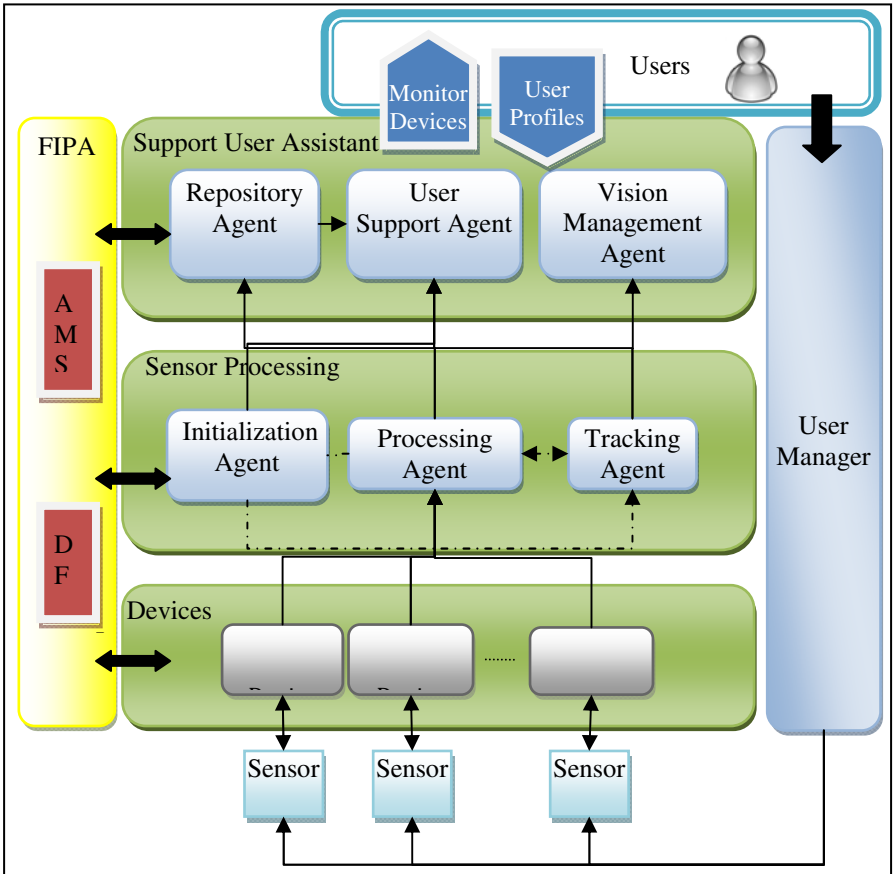
Fig. 1 Multi-Agent Based Distributed Semi-Automatic Surveillance System Architecture. This image shows the three different layers which compounds the whole system relying in FIPA platform.

associated device. Each monitor, camera or other sensor has associated in this layer only one agent.

The tasks of second layer, Sensor Processing Layer, are the construction of marked target, introduces different sensor data to maintain tracking in the proper way to facilitate the users visualization. This layer refines the sensor data producing semantic description to interpret for upper-layer and coordinates three agents, initialization agent, processing agent and tracking agent.

The third layer, support user assistant, analyses bottom-layer data in order to produce actions and suggests to user different ways to perform a correct management. Generally, these actions could result in generate movement in sensors or change their parameters for example, avoiding possible hole in global line vision, keep a tracking or show user most relevant image in monitor system.

The proposed architecture is a semi-automatic intelligent multi-agent system, where each function is performed by an agent or combination thereof. The global task is giving support and assistant to user by cooperation between agents. The whole system lies in an agent platform that provides generic tasks and decentralized cooperation mechanisms. As discussed we have chosen FIPA platform because is a well-known standard and open platform.

4 Agents Description

In this section we explain the different agent involved in the system and all architecture layers.

4.1 FIPA Platform Agents

A FIPA platform is a middleware that provides a collection of services. These services allow an improvement in the production of multi-agent systems. This platform has two mandatory agents to perform coordination:

— The AMS - Agent Management System
— DF - Directory Facilitator.

The AMS provides services to management creation, registration and erased of agents in the platform (and the system itself).

The DF agents provide information about agents in the platform such as services offered by agent, name of agent, etc. DF offers a yellow pages service. Additionally AMS and DF provide a message transport service.

The agents in the platform only need to communicate call a specific API that allows them to send messages to other agents in the same platform, even in others. A possible implementation of this platform could be JADE [6], perform in Java language. JADE has the property of portability and it does not depend on native computer architecture.

4.2 Device Agent

This agent perform controlling and management of sensor. It provides the image to next layer for corresponding processing. Definitely the agent has two tasks: provides sensor data to system and receives the possible data that modifies their attributes.

The agent should have negotiation capacities for communicate with other devices agent in the same layers.

This agent should offer the same interface for any sensor to facilitate inclusion of new sensor devices with minimum impact in the architecture. However, when communicating with the top layer should be informed the services it can offer. The different sensor types provides different data types.

4.3 Initialization Agent

The main goal of this agent is initializing tracking through targets detection. This function starts when the user notifies to system about some element to track or sensor data processing like radars.

When the target is located, the information is transmitted to tracking agent and processing agent. This agent also classifies the element to track such as suspicious target or not suspicious target with a confidence factor (high when the user requests this tracking).

4.4 Tracking Agent

This agent is a multi-process agent, with the purpose of tracking the targets trajectory and speed, in order to predict their position for commutate between sensors to keep target tracking when actual sensor that it performs tracking cannot follow this functionality.

It is possible to use a large number of algorithms to perform this tracking, one example may be the work done in [7] by Luis Botehlo. This prediction process is formed by two algorithms: prediction algorithm thanks a model and learning model to readjust the model factors.

4.5 Processing Agent

The vision frontier must be constant, avoiding overlap and hole in its vision. Therefore it is necessary managing the movement and other sensor characteristic to maintain this frontier.

This agent manages the movement and sensor range, and should keep the maximum distance in its vision range trough the sensor intervals.

The final functionality that this agent has is to provide sensor data to others agents in the same or upper layer. It could manipulate the sensor data to visualize them better.

4.6 Repository Agent

The principal aim of this agent is to store content offered by sensor and accumulate the performed reports about the system activity (tracking, changes in vision line…).

This agent offer a great historical data that the user can consult, even is possible discard target thanks to recognize in this repository a possible target that has no impact on tracking process because it is not suspicious.

This data could be accessed by the user through agents in upper-layer. Even this agent in upper-layer could be accessed this information to perform a possible choices.

4.7   User Support Agent

To facilitate the user work for inspecting unusual or suspicious target in a concrete environment, this rational agent compiles and processes the information in repository agent and tracking agent.

Thanks to initialization agent, it could suggest more suspicious target to track with sensor devices, perform different report types such as last target tracking or sensor incidents or state and even it suggest stop tracking a concrete target because it is not suspicious.

This agent needs learning algorithms to adapt different user profiles and promote a proper sync with the user.

4.8   Vision Management Agent

In large security rooms where there are many monitor devices is difficult appreciate all events that produce in vision line. Also many sensor devices are most interesting than others. For this reason, it is necessary selecting most important sensor device to view in monitor devices.

This agent provides this capability due concise analysis about user preferences and relevant data extracts to repository. Monitor characteristic are different between them, even if all monitor devices have same characteristics, their locations changed and some specific location offer a better view than others. In consequence, the monitor characteristics and location are useful for users because they could watch a limited number of screens.

For example, tracking an objective is most important with a sensor device than others where nothing unusual is reporting. The target data that sensor is tracking thus view in a screen.

Also it can perceive the shutdown or occupation in other activities of sensors to avoid holes in the frontier. Also it can perceive an excessive overlap between sensors and order to separate the vision sensor ranges

The agent should integrate proper algorithms that provides to user these capacities and must be in this agent.

5   Agent Management Sample

When this agent perceives any discontinuity, it begins a communication between device agents that stimulates a negotiation between involved agents and continues until the hole or overlap is avoided. Also the users modify the vision range or charge other profiles that specifies a concrete vision range, this agent evaluates and notify to devices agents the new ranges leaving the negotiations between them.
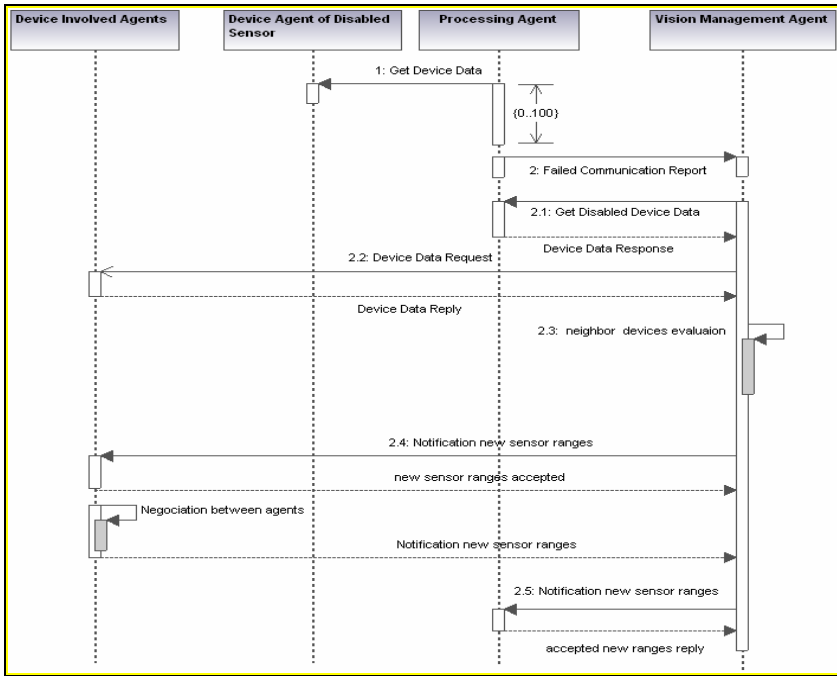
The figure represents this data flow.

Fig. 2 This image shows interaction diagrams that it represents when the system detects a discontinuity

6 Conclusion

We have proposed an intelligent distributed semi-automated system based on technology multi-agent, the architecture avoids a reliable management of different sensor, tracks targets and advises users to facilitate their work.

The main advantage of using agents over video algorithms is:

— The tasks performed by each agent could be incrementally sophisticated without impact on system.
— Autonomous programs could be tested and developed independently because each service and device control is encapsulated in an agent by using middleware technology, which can improve intelligent and automation degree of information processing, which ensure that all raw data could be processed efficiently.

The information that could be processed by users is minor, promoting the user efficiency because the user can focus in most priority events and cases.

This generic architecture provides a middleware to build a concrete implementation in concrete environments depending local circumstances and characteristics.

7

References

1. Valera Espina, M., Velastin, S.A.: Intelligent Distributed Surveillance Systems: A review. IEEE Proc. Vision, Image and Signal Processing, 192–204 (April 2005)
2. Pavón, J., Gómez-Sanz, J.J., Fernández-Caballero, A., Jiménez, J.J.V.: Development of intelligent multisensor surveillance systems with agents. Robotics and Autonomous Systems 55, 892–903 (2007)
3. Detmold, H., van den Hengel, A., Dick, A., Falker, K., Munro, D.S.: Middleware for Distributed Video Surveillance. IEEE Distributed Systems Online 9(2), 1–10 (2008)
4. Foundation for Intelligent Physical Agents, FIPA 97 Specification, Version 2.0 Part 2 Agent Communication Language (1998)
5. Searle, J.R.: Speech Acts. Cambridge University Press, Cambridge (1969)
6. Java Agent Development Framework, `http://jade.tilab.com/`
7. Luis, B.A., Botelho, L., Cavallaro, A., Douxchamps, D., Ebrahimi, T., Figueiredo, P., Macq, B., Mory, B., Nunes, L., Orri, J., José, M.: Trigueiros and Ana Violante, Video-Based Multi-Agent Traffic Surveillance System. In: Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 457–462. IEEE, Los Alamitos (2000)
8. Georgeff, M.P., Rao, A.S.: The semantics of intention maintenance for rational agents. IJCAI, 704–710 (1995)