Fuzzy Reasoning in a MultiAgent System of Surveillance Sensors to Manage Cooperatively the Sensor-to-Task Assignment Problem

José M. Molina¹, Jesús G. Herrero¹, Francisco J. Jiménez², José R. Casar³

¹Dpto. Informática, Universidad Carlos III de Madrid.
Avda Universidad 30, 28911, Leganés, Madrid, SPAIN
Tel:34-91-6249116, FAX:34-91-6249129, e-mail:molina@ia.uc3m.es

²Indra S.A. Torrejón, SPAIN

³GPDS-DSSR- ETSIT, Universidad Politécnica de Madrid, SPAIN

ABSTRACT: In this work, a surveillance network composed by a set of sensors and a Fusion Centre is designed as a multiagent system. Negotiation among sensors (agents) is proposed to solve the task-to-sensor assignment problem (the allocation of tasks to sensors), addressing several aspects. First, the Fusion Centre determines the tasks (system tasks) to be performed by the network at each management cycle. To do that, a fuzzy reasoning system determines the priorities of these system tasks by means of a symbolic inference process using the fused data received from all sensors. Besides, a fuzzy reasoning process, similar to that performed in the fusion centre, is proposed to evaluate the priority of local tasks (sensor tasks), executed now by each sensor. The network co-ordination procedure will be based on the system-task priorities, computed in the Fusion Centre, and on the local priorities, evaluated in each sensor. Priority values for system and sensor tasks will be the basis to guide a negotiation process among sensors in the multiagent system. The validity of the fuzzy reasoning approach is supported by the fact that it has been able to manage environmental situations in a similar way as experienced human operators do. Included results illustrate how the negotiation scheme based on task priority, measured through their time-variant priority, allows the adaptation of sensor operation to changing situations.

Keywords: Surveillance Sensors, Radar Management, Distributed Coordination, Multiagent Systems, Fuzzy Inference Systems.
1. Introduction

Modern Air Surveillance systems [Bla86] [BP99] are composed of several sensors that provide data (detections, attributes, processed local tracks, etc.) about all interesting objects (targets) located within their covered areas. Both in civil and defence applications, sensors (primary and/or secondary TWS radars, passive sensors, image sensors, IFFs, ESMs, multifunction radars, etc.) are sources of data to be merged at the Fusion Centre before presentation to controllers. In the Fusion Centre [WL90] [Lli93], the Data Fusion function combines sensor detections, together with related information, in an optimal or suboptimal way to achieve better inferences than those which would come from a single sensor [HL01] [WL90] [Rei85] [LG83]. Therefore, multisensor integration is becoming an essential aspect of modern surveillance systems in order to collect the information necessary and develop, by means of Data Fusion techniques, the appropriate perception of the scenario situation [Bar92].

Among available sensors, some kinds of advanced ones, such as multifunction radars, are able to simultaneously carry out different types of tasks (search, track, identification, backscanning, reacquisition, etc.), taking their own management decisions to perform all of them. Each task is defined by a set of execution parameters (performance objectives to achieve) which will require a certain amount of the resources available in the sensor. Some examples are track quality/accuracy for track-update tasks, reliability of track identity for identification tasks, early probability of detection for search tasks, etc. In the same way as there are execution parameters for each sensor task, if we consider a network of this kind of sensors (in a general case), system tasks will define its own system-performance objectives. The coordination in the network, basically aimed at achieving an appropriate allocation of system tasks among available sensors, depending on time-varying conditions, will be the central problem addressed in detail in this work.

It must be noticed that sensor information cannot always be reduced only to numerical quantities; in fact, human management operators evaluate data in a subjective way. The reasoning process of such an expert could be emulated using Artificial Intelligence techniques [MM87] [Yan85] at the Fusion Centre, determining the importance of each task based on subjective information. In this work, the formal method selected to represent the variables involved in the decision process has been the
theory of possibility and fuzzy sets [KF92] [MMS96], since it offers a unified framework to represent uncertain knowledge and emulate expert reasoning processes.

The integration of information and its subjective interpretation can be used to coordinate sensors in the network with the goal of optimising acquisition processes at each sensor [Bar92] [PB87] [BP99] [Cow87]. The main system tasks to be performed by the network considered in this work have been the following: Track Update (obtaining new plots to maintain tracks representing each detected target), Identification (obtaining target IDs) and Search (obtaining new targets entering in the global coverage area). Other types of tasks, such as backscanning, confirmation, etc., have been considered as local to the sensors, to optimise its time response. So, tasks will be classified as system tasks (referred to the network) and sensor tasks (locally referred to a specific sensor) [WL90]. For example, a track update system-task (a target whose track should be updated by the network with a certain rate) could be assigned to several sensors, each one of them having to execute a different track update sensor-task over the same target. The global amount of data received from all assigned sensors would achieve the desired system-task goals. In the same way as a sensor task is defined by its type, priority and execution parameters (power, time period, etc.), system tasks can be defined by certain global performance objectives to be achieved through the execution of related sensor-tasks. This system-level quality objective will be determined at the Fusion Centre [Bar92].

Automatic management systems for coordination of sensors can be structured according to two types of architectures: centralised and decentralised [WL90]. In any case, the fusion centre will be always the node with the most complete information about the environment and therefore it should be the most influential in the net. The system tasks must be defined in this node and a (symbolic) reasoning system [PB87] [Bar92] [MJC98a] [BP99] active at this node could be able to evaluate the importance of each system task considering inferred information of various abstraction levels. Two different types of information are considered in the fusion centre: global or high-level knowledge, which is a result of the data fusion process; and local knowledge, internal to each sensor (such as current sensor load), that should be estimated if a centralised scheme will perform the assignments [Bar92] [WL90].
Generally, most proposed architectures for the coordinated operation of multiple sensors are based on the centralisation of management at the fusion centre, which take decisions affecting to all deployed sensors. In a centralised architecture, the multisensor system manager (MSM) has to decide the set of tasks to be accomplished by the system in the next management cycle and the full list of tasks that each sensor should execute [WL90] [MJC95a] [BP99]. So, in centralised architectures the reasoning process is implemented in the fusion centre and the sensors are slaves, with only one way to communicate decisions: from the fusion centre to the sensor nodes. The problem of sensor-to-task assignment, as well as the determination of parameters that define each task, has been approached from different perspectives within this centralised framework [WL90] [LH78] [Bar92] [PB87]. In all of them, the system manager calculates first the priority of each system task and then, based on this priority, it has to assign the tasks. Priorities are obtained as a function of the pre-determined system performance objectives and the capacity/load assumed for each sensor. The sensor-to-task assignment problem, assuming the priorities have been already computed, has been dealt by some authors [WL90] as a transportation problem and solved using Operations Research methods. Alternative approaches are based on the study of the utility that the accomplishment of a concrete task represents for each sensor [Bar92]. The quantification of the utility is made in [Bar92] by numerical estimation of the cost of each assignment and, in others such as [MD94], by the measure of the entropy of the system information.

However, the existence of “intelligent” sensors, such as the mentioned multifunction radars, capable of taking their own local management decisions, allows for the possibility of conceiving alternative decentralised architectures [Wes88]. In a distributed architecture, the reasoning process is placed in every available node, which will reason about its local tasks accordingly to the perceived environment and local conditions. In this case, each sensor should be able to redirect tasks to other sensors (cueing) with overlapped coverage, and/or additionally, to transfer tasks (track update, identification or search tasks) from a sensor to another (hand-off) [Bar92]. Here we will propose to consider a decentralised architecture (having the sensors a certain degree of autonomy) as a Multiagent System (MAS). In a MAS the decisions are taken as a result of the interaction of all the nodes [Dur91]. The development of a Multiagent System is based on Distributed Artificial Intelligence (DAI) techniques [Gas92]. The general philosophy of DAI is the decomposition of a problem to be solved by several subsystems, and a distributed system could be defined through the
following characteristics [Cam88]:

- The system consists of a collection of subsystems (agents). Each agent has various skills, including sensing, communication, planning and acting.
- The group as a whole (the system composed by agents) has a set of assigned tasks (goals) to be solved.
- Each subsystem (agent) has only limited knowledge, there is no place in which all the knowledge is contained.
- Each subsystem (agent) has different capabilities and, then, differing appropriateness for a given problem or subproblem.

With these DAI context, in a Multiagent System (MAS) [BZW98] [DLC89] [BG88] [Gas92] the global problem is decomposed in subproblems that are dynamically solved by different agents as functions of its appropriateness. A MAS system is based, fundamentally, on two ideas: the agent concept and the coordination of agents to achieve common goals. An agent could be defined as a system with the following properties [WJ95]: autonomy (an agent is able to decide what to do using its local control over its actions and internal state), social ability (agents interact with other agents using some kind of agent communication language), reactivity (an agent is able to sense the environment and respond to the changes on it), pro-activeness (agents have their own goals). The advantages of MAS are related, mainly, to the characteristics of distributed problems:

- They are able to solve big-size problems, especially those where classical systems are not successful.
- They allow different systems to work interconnected and cooperate.
- They provide efficient solutions where information is distributed among different places and computation is asynchronous.

Once we have conceived surveillance networks as MAS, we will design a negotiation process among sensors to solve the task-to-sensor assignment problem. As in centralised approaches, task allocation will be based on priorities. In this case with MAS architecture, tasks’ priorities will be used to direct the negotiation process among sensors to assign each task. Each sensor will be able to evaluate the local priority of each task (that actually is being executed) to negotiate [DS83] with the rest of sensors [MJC98b]. This negotiation process allows the coordination among nodes [LS95].
This contribution proposes a distributed task-allocation methodology in multisensor scenarios with a two-fold approach. A MAS has been designed to address sensors management and coordination based on task priorities, and a fuzzy reasoning system has been developed to evaluate these task priorities at two different levels: system and sensor. To obtain the priority for each task, the reasoning process relies on a decision tree whose nodes are linguistic variables that represent the intermediate concepts that would be handled by a human operator to determine the task priority. These fuzzy values are useful to optimally perform the sensor-to-task assignment in a distributed way [MJC2001].

The rest of paper is organised as follows. In section 2, the distributed architecture of the system is described. In this section, multiagent ideas are developed and, agents and its internal architecture are defined. In section 3 the fuzzy reasoning system is explained, and the fuzzy priority evaluation for track update task illustrates the procedure. Section 4 describes the coordination process based on the cooperation among the sensors. In section 5, some experiments shown the validity of the proposal, and, finally in section 6 some conclusion derived from this work are presented.

2. Decentralised Management in Multisensor Networks: a Multiagent Approach

As indicated above, the problem of multiple surveillance sensor coordination will be approached from a cooperative perspective, following a multiagent methodology to achieve the assignment of system tasks among the available sensors. The decision-making process in a multisensor network, considered as a MAS, will be distributed among several network nodes, each one considered as an agent [MJC97]. This strategy can be useful for two managing purposes. First, it allows the potentially autonomous operation of each sensor node, taking advantage of the local sensor-manager capability to take the final decisions about its own sensor-level tasks. Secondly, the multisensor manager (Data Fusion centre) will only need to perform a general supervision of the accomplishment of the demanded system-level task performance objectives, remaining free of the sensor-level task scheduling function.

The coordination of the surveillance network will be mainly based on task priorities, as in the
centralised approaches. The fusion centre will periodically deliver a list of system-level tasks to be fulfilled, each one with its system-level priority, and with an indication of the system-level performance objectives (expressed in different terms depending on the nature of tasks). The local manager at each sensor will also decide list of sensor-level tasks to execute, each one with its sensor-level priority and its sensor-level performance objectives. In both cases, fusion centre and sensors, a symbolic fuzzy reasoning process working on the data from sensors and (external) intelligence information will infer, respectively, the priorities of system-level and sensor-level tasks. Finally, the sensor(s)-to-task(s) assignment, involving the system-level to sensor-level task decomposition and system-level to sensor-level performance objectives translation, will result from a negotiation process among sensor agents, initiated on the information previously sent to them by the fusion agent.

The coordination of different agents is a complex problem [Syc98]. In a distributed architecture, the coordination to obtain the solution of global goals is obtained by means of its local control and the communication among agents. Basically two dimensions, the control and the communication, ought to be addressed [DK87]. The control defines the cooperation, the organisation of agents and the dynamics of the control organisation in time. The communication specifies the protocols, the contents of the message among agents and the paradigm by which the communication takes place.

So, the architecture of an agent could be decomposed in three components [LS95]: task solving component (agent body), the cooperation super-strate (agent head) and the communication functionality (agent communicator). These components must be defined for each type of agent for a given problem in order to specify its capacity in two dimensions: what the agent can do (skills) and how the agent can coordinate with other agents (control and communication). Next we specify the system architecture, structures for the types of agents proposed in this application, and the basis of the negotiation strategy to achieve the coordination goals.

2.1 System Architecture
The surveillance network is composed of two types of agents [MJC97][MJC98b]: Sensor agents and the Fusion (centre) agent. Each sensor agent will have knowledge about the Fusion agent and the subset of Sensor agents to which it is connected (its potential negotiators, those sharing
overlapped coverage). The Fusion agent will have knowledge about all sensor agents, and so it is connected to all of them.

Each agent’s architecture will follow a generic structure with three levels: Execution, Planning and Communication, being the Communication level based on an agenda containing linguistic acts that, when processed, generate messages to other agents or modify the internal state of the agent (through the Planning Level). The objectives of each level in the two types of agents conceived for this application are detailed next:

- **Fusion Agent Architecture**

  In the Fusion agent, the Execution Level is devoted to the Data Fusion (DF) and Situation Assessment (SA) functions, which integrate low-level measured data sent by the sensors and/or additional information potentially provided by the operator. As tasks could be decomposed and executed by several sensors, the Execution Level monitors that fused results achieve the required quality. The Planning Level (based on the information provided by the Execution Level) determines the set of system-level tasks and their priorities to be performed during the next system management cycle by the multisensor network. Finally, the Communication level sends to sensor agents the system-level tasks (including their priorities and performance objectives) and receives the final task-to-sensor assignment resulting from negotiation among the sensor agents.

- **Sensor Agent Architecture**

  In the Sensor agents the Communication Level supports the distributed decision (task negotiation) process. When a task is finally accepted, the Communication Level transfers the execution parameters resulted from negotiation to Planning Level. The Planning Level takes decisions about the set of sensor-level tasks to be performed during the next sensor management cycle and their priority, based on the tasks transferred from Communication Level. This level manages the sensor when operating autonomously and it is built as a fuzzy-rules based inference system. The Execution Level is responsible for the execution of sensor tasks. The Execution Level functions are different depending on the type of sensor.

**2.2 Coordination through Negotiation**
The proposed decentralised management scheme is based on the negotiation strategy outlined next (see Figure 1). First, the Fusion Centre (system manager), based on the global perception provided by DF and SA functions, sends to each sensor the system-level tasks that have to be accomplished by the sensor on its own, and those that could be jointly performed with other sensors (whose identities are also sent).

![Figure 1: Communication for Agents Coordination](image)

Each sensor (Sensor Manager), according to its own necessities, negotiates with the rest of involved sensors the final decomposition of system tasks into sensor tasks. Finally, each sensor will execute its final list of assigned sensor tasks. The process is supervised by the fusion agent (Coordinated Operation Scheme), which receives reports from sensors and so keeps models about sensor possibilities (coverage, load, etc.).

Each sensor task will be defined by its execution parameters, needed to reach the demanded system-level performance objectives of their parent system-level task. A system-task could be decomposed in several sensor tasks, each one with different execution parameters, guaranteeing that the quality of the system task achieves the values defined by the fusion agent. This decomposition will depend basically on the global task priority specified by the fusion agent, and local priorities perceived by each sensor. During the management cycle, the fusion agent only monitors each system-task quality, in order to solve low-quality values through communicating with the sensors that execute those sensor tasks related with the monitored system-task.
The task(s)-to-sensor(s) assignment that is agreed among the sensors will be maintained until the next group of system-level tasks is sent by the fusion centre. The details of negotiation process will be given in section 4. Before that, next section presents the proposed fuzzy systems to compute the system and local task priorities through a possibilistic reasoning process.

3. Fuzzy System to Evaluate Task Priority

The proposed management scheme is based on sensor negotiation, where proposals are elaborated using basically the values of task priorities. These task priorities are calculated in each agent (both Sensor and Fusion) using a symbolic representation of track and sector data [Bar92] [Bla86], derived from available sensor detections and conventional data processing algorithms. The fuzzy reasoning system will determine a priority for each task, to be accomplished in each decision cycle. The evaluation of task priority will use an inference process similar to the process that would be performed by a human controller [BR94]. The design of a knowledge based system [GD93] to evaluate tasks priorities involves three fundamental steps: (1) identification of variables representing relevant Tactical Situations, (2) definition of relations among variables and its values, (3) development of the fuzzy-rule base that implement the relations among variables. Steps 2 and 3 can be represented in a decision tree. In this tree, the nodes are variables that represent the intermediate concepts used by the human operator to define the task priority. The fuzzy system will emulate the decisions taken by an expert under the design conditions (steps (1), (2)), but also will be able to generalise and provide adequate outputs in intermediate situations (interpolation capability of fuzzy systems, [NS98]).

The variables involved in this decision process will be represented using the fuzzy set theory [Zim90]. In this way, given a numerical parameter of i-th variable (v_i), V is defined as the range of all possible values of the computed parameter (universe of discourse for that parameter). To better cope with the intrinsic variable uncertainty, the numerical values v_i could be mapped into qualitative symbolic labels, through a fuzzification process [Zim90], transforming the numerical parameters into linguistic variables. A linguistic variable [KF92] is a variable whose values are sentences in a natural or artificial language, that is, a concatenation of atomic terms: labels (adjectives), hedges (modifiers such as very, much, slightly, etc), the negation and markers (parentheses). Each variable (named linguistic
variable in fuzzy theory) is defined by means of a set of labels, for example: NEAR, HIGH, LOW, etc. A label is defined by a fuzzy subset, $A$, of a universe of discourse, $U$, and it is characterised by a membership function $\mu_A: U \rightarrow [0,1]$ which associates with each element $y$ of $U$, a number $\mu_A(y)$ which represents the degree of membership of $y$ in $A$. The operation of fuzzification (application dependent) has the effect of transforming a nonfuzzy set or quantity into a fuzzy set.

System rules (IF THEN rules) translate the degree of membership from IF-part to THEN-part. The rules are represented in a fuzzy relational algorithm (FRA) [Zim90] that will store the knowledge required to obtain the task priority under different input conditions. The FRA will be composed of a finite set of fuzzy conditional statements of the form IF (antecedent) THEN ($V_{\text{PRIORITY}}$ is label $k$), where (antecedent) can be conjunctions and/or disjunctions of fuzzy statements about linguistic variables of the form (Variable $i$ is label $v_{ij}$). The Mamdani implication [KF92] has been chosen to assign the meaning to these fuzzy conditional statements. Finally, the defuzzification process on $V_{\text{PRIORITY}}$ (with a “Centre of Area” procedure) will transform the fuzzy output variable into a numerical value to be used in the agent negotiation process.

Next, in section 3.1, the relevant Tactical Situations (steps 1 and 2 of knowledge based system development) are described for the considered type of tasks (Track Update, Identification and Search). In section 3.2 some details of the fuzzy system (step 3 of knowledge based system development) will be given, illustrating the whole process for the specific case of Track-Update tasks priorities.

### 3.1 Relevant Tactical Situations

The identification of relevant variables representing interesting Tactical Situations is the result of a design process including several knowledge-engineering sessions with a team of experts in surveillance systems. During this process, they were faced with several representative tactical situations to synthesise variables and relations that should be considered to define the priority of each type of task under certain conditions. The resulting variables and relations are represented by means of a decision tree, where the leaf nodes represent the linguistic variables associated with information (uncertain in nature) directly supplied by Tracking, Threat and Situation Assessment (SA) functions (all situated at Data Fusion levels). Information, such as the following items, should be obtained from...
the SA function: Friend Degree, Target ID, Declaration Uncertainty and New-target Rate. Additional required information such as Friend Forces Location, Guided Missiles in Flight, Enemies on Target, Own Weapon Systems Capacity and Static Priority of each Surveillance Sector will be obtained from ancillary data bases accessed by the Situation Assessment function.

The set of rules that define the fuzzy decision trees, whose fuzzy antecedent and consequent establish the relationship between the tree nodes, constitute the knowledge base of the fuzzy system. Starting from the initial fuzzified (DF+SA) information, the inference engine of the knowledge-based sensor manager proceeds up (data-driven) using forward chaining through the trees generating the intermediate conclusions until a linguistic value is assigned to the priority of all system-level tasks of each type informing about its subjective necessity. Relevant situations and decision trees have been defined, taking account of the most relevant aspects considered by experts faced to representative situations, for each one of the considered types of tasks: Track Update, Identification and Search:

- Tactical Situations for Track-Update Tasks

The selected tactical situations represent the most relevant aspects that should be considered for the definition of the track-update task priority: (1) guarantee of track continuity maintenance, (2) availability of the target identity, (3) type of target trajectory. Regarding the first aspect, a given minimum illumination rate of a target will be always necessary to maintain its track, taking into account its manoeuvring capability. With respect to the type of target trajectory, it will increase or reduce the priority/importance of the Track-Update task, considering also the target identity. Three types of relevant situations have been distinguished for this aspect:

1. Trajectory that could be dangerous in a future time. In this way, a target, placed far from a sensor and directed towards it with a high speed represents a hostile trajectory.
2. Trajectories followed by a target surrounding a sensor, although not directed towards it. This concept could be extended to targets that are close to critical positions, for example, because the positions are defined as forbidden or, because the targets are close to defended positions.
3. Trajectories that are actually dangerous, defined by the typical paths of missiles, enemy groups and strong manoeuvres.
In Figure 2, the decision tree for evaluation of Track-Update and Identification task priorities, considering these relevant situations, is depicted. The aspects mentioned here to derive track-update task priority in the reasoning process are represented with linguistic variables, including the bottom level for fuzzified (DF+SA) information, intermediate concepts and the final concepts at the top. These variables and their relationships will be detailed in section 3.2.

Figure 2: Decision tree for Track Update Task Priority Evaluation.

- Tactical Situations for Identification Tasks

The priority of identification tasks is determined by the system needs of identifying each target. Any target within the coverage and which has not yet been reliably identified ought to be identified as soon as possible. Therefore, the concept that mainly determines the identification tasks priority is the identity of the target and the probability of error in the identity, measured as the identity uncertainty. Thus, for example, at the initiation of a track, the identification task will receive very high-priority if at that moment the target is considered unknown. As the track evolves with time, the uncertainty of track identity will evolve as a function of the radar capacity to identify the target and the target
capacity to deceive the system. Among all these targets whose identity is unknown, or have a high uncertainty degree, those targets that perform a potentially threatening trajectory should have a greater identification priority. The decision tree is similar to Track Update Tasks (see Figure 2).

- Tactical Situations for Search Tasks

The coverage area is divided in sectors, which should be explored at a certain periodic rate to guarantee a certain capability for early detection of new targets. In addition to the consideration of the sensors scan rate, the need to perform a search task at a sector is determined by other three factors: (1) Previous Knowledge about the Sector, (2) New Target Appearance Rate (per area), (3) Discrimination between (close) Targets. The first factor besides includes two types of information: the physical location of the sector within the whole overlapped multisensor coverage area, and collateral information supplied by the operator. The first information is the same that would be used to assign static a priori priority to each sector. The second information originates directly from operator criteria. For example, once the current particular situation of the environment is known, it is possible for him/her to identify the zone from which enemy targets tend, at this time, to appear. Those sectors, where the new target appearance rate (number of initiated tracks) is greater than in others must be searched more frequently than sectors with lower new target rate. The new target rate is computed based on the number of targets that enter the sector volume during the time elapsed from the last sector search. In Figure 3 the decision tree for Search Task Priority evaluation, considering the relevant situations indicated above, is depicted.
3.2 Fuzzy System to Evaluate Track Update Tasks Priority

Among the three types of tasks indicated, we will illustrate here the details of the fuzzy reasoning system computing priorities of Track-Update tasks. Each one of the concepts identified in section 3.1 will be represented as linguistic variables in a tree (Figure 2), taking each one different possible values which are labels of fuzzy sets (characterized by a membership functions) in the universe of discourse (domain) of the represented concept. For example, the linguistic variable [Track Update Task Priority], whose universe of discourse ranges from 0 to 100 (numerical degree of necessity), will have 5 linguistic values (labels): VERY HIGH (VH), HIGH (H), NORMAL (N), LOW (L), VERY LOW (VL), with equally spaced same-width trapezoidal membership functions. In Figures 4-9 the definition of variables for Track Update Task Priority are shown, together with rules defining the relations among them.

The relationships between these concepts are the connections between nodes, which graphically represent the set of rules relating them. In the case of Track Update Task Priority, the existence of three types of trajectories relevant for track-update task priority, as indicated in section 3.1, is reflected in the tree (Figure 2) through three different concepts taken into account to compute the final priority:
1. Approach of the target to a defensive position (sensor): represented by the concept [Potential Hostility]. This concept relates the possibility that the target accomplishes an approximation to a defensive position and the possibility that the target will be enemy or unknown, due to a high value of uncertainty in its identity.

2. Target proximity to a critical position: represented by the concept [Relative Threat]. This concept informs about the relative location of targets respect to sensors and several positions/areas forbidden for hostile targets.

3. Danger inferred from the trajectory developed by the target: represented by the concept [Target Threat]. It relates the information about a trajectory threat with the knowledge of the target identity. A trajectory, which represents a potential attack, could be considered as threatening if the target is enemy or not known.

As we can see in the tree, these concepts depend on other ones located below, detailed next from the lowest levels up to them. The concept [Approach] considers the relative track speed with respect to a defensive position (a sensor deployment). This variable is a numerical input that calculates the approaching speed to a position. The mentioned concept [Potential Hostility] is after inferred relating the values of [Approach] with the possibility that the target is enemy or unknown, represented with the concepts [Friend Degree] and [ID Uncertainty]. The variable [Friend Degree] fuzzifies the value of the friend declaration probability and the variable [ID Uncertainty] fuzzifies the difference between the two probabilities (friend/foe probability) that represent the uncertainty of the statement.

The fuzzy variable [Relative Threat] evaluates the distance from the target to a defended position. For each one of these positions, a minimal inviolability space threshold is defined to represent the safety range, $R_{SAF}$, where no hostile target should enter.

The concept [Trajectory Threat] represents the degree of possibility that the target flies according to one of the four trajectories defined as dangerous: missile, diving, group flight and manoeuvre. The linguistic variable associated to this concept is fuzzified in four linguistic labels, one for each type of dangerous trajectory, respectively, [Missile], [Diving], [Group Flight] and [Manoeuvre Capability]. The concept [Missile] is defined through the target flight altitude as well as by the target size.
(concepts [Flight Altitude Increase] and [Target Size]). The incorporation of the target identity and
the associated uncertainty is accomplished through the concepts [Friend Degree] and [ID
Uncertainty], which are added to [Trajectory Threat] to obtain the value for [Target Threat].

Once we have defined the relevant tactical situations, the concepts that allow to represent all
situations and the relationships among them, the labels of each linguistic variable and the rules relating
all of them are detailed next. In all cases, each set of rules is represented with a table with all input
combinations and resulting output value, when there are two variables connected in the antecedent
(and conjunction). If there are three variables, a table is presented for each value of the third
variable. In Figure 4, the inference rules to obtain [Diving] from [Flight Altitude Increase], [Velocity
Increase] and [Approach] are shown. The possibility that the target is a missile, [Missile], is
calculated from [Target Size] and [Altitude], with the set of rules shown in Figure 5. The variable
[Trajectory Threat] collects the possibility of each one of the possible trajectories [Missile], [Diving],
[Group Flight] and [Manoeuvre] through the rules in Figure 6.

![Figure 4: Fuzzy Rules to obtain [Diving].](image)

![Figure 5: Fuzzy Rules to obtain [Missile].](image)
Figure 6: Fuzzy Rules to obtain [Trajectory Threat].

The last level of the inference chain relates [Potential Hostility], [Target Threat] and [Relative Threat] (see Figure 9). [Potential Hostility] is inferred from [Approach], [Friend Degree] and [ID uncertainty] following the rules of Figure 7. Finally, [Target Threat] depends on [Trajectory Threat] (obtained previously with the rules of Figures 4, 5 and 6), [Friend Degree] and [ID uncertainty] as can be seen in Figure 8.
Figure 7: Fuzzy Rules to obtain [Potential Hostility].

Figure 8: Fuzzy Rules to obtain [Target Threat].

Similar fuzzy systems have been implemented to evaluate the priorities for the other types of tasks considered, Identification and Search. Each sensor manager performs the whole reasoning process to evaluate the priorities of its local tasks, and the Fusion Centre to obtain priorities for global tasks. As indicated in section 2, these priorities will be the basis, together with the performance objectives, to coordinate the assignment of tasks among the available sensors, employing the multiagent negotiation strategy detailed in the following section.
Figure 9: Fuzzy Rules to obtain [Track Update Task Priority].

4. TASKS-TO-SENSOR ASSIGNMENT

As we indicated in section 2.2, the tasks-to-sensor assignment problem has been divided in two steps: firstly, the Fusion agent informs Sensor agent(s) about the system tasks to be performed in the next management cycle and, secondly, the final distribution of tasks among sensors is carried out by a negotiation among themselves (which is also supervised by the Fusion agent). The negotiation process will result in the final sensor-level task-to-sensor distribution, including the sensor-level
performance objectives corresponding to each task, that achieve the system-level performance objectives demanded by the Fusion agent.

Each system-level task, $T_i$, is defined by its priority, $P(T_i)$, and its system-level performance objectives, $O(T_i)$. $P(T_i)$ is a real defuzzified value, obtained after the process described in previous section. The system-level performance objectives are defined by means of four linguistic labels qualifying the figure of merit that has been selected for each type of task. For example, for Track Update tasks, the associated figure of merit is the accuracy, with the following labels: Very High (VH), High (H), Low (L) and Very Low (VL), each label referring to a subset of values for the time interval of next target illumination. Sensor-level tasks are defined in the same way. Each sensor, $S_j$ ($j = 1, ..., M$), is able to infer its desired defuzzified priority, $P_{Sj}(T_i)$, for each one of its sensor-level tasks, $T_i$ ($i = 1, ..., k_j + 2n_j$), being $k_j$, $1 \leq k_j \leq K$, the number of sectors covered by sensor $S_j$ and $n_j$, $1 \leq n_j \leq N$, the number of targets within its coverage. The performance objectives for sensor-level tasks, $O_{Sj}(T_i)$, and their final execution parameters will result from a negotiation process relying on the sensor-agent(s) local priorities and system/sensor-level performance objectives demanded by the Fusion agent, or other Sensor agent(s) before/during the negotiation.

System-level tasks can be classified in three categories regarding their possibility of being shared among different sensors.

- **Isolated tasks**, that must be executed by a certain sensor agent without possible cooperation with any other agent(s). For example, Sector Search tasks on a sector covered only by a single sensor.
- **Joint tasks**, that could be executed by several sensor agents. For example, Track Update or Identification tasks for targets simultaneously illuminated by several sensors.
- **Individual tasks**, those originally joint tasks that finally can be executed only by a single sensor. For example, when all sensors, except one, are inhibited or limited (due to jamming, excessive load, etc.). The fusion agent knows at a certain time that a system-level task can only be accomplished by a specific sensor since the affected sensors have previously reported to the Fusion agent about its negative decision.

The allocation process begins once the Fusion agent has sent to each $j$-th sensor agent the list of
system-level tasks it can execute, \( \{T_i\} \) \((i =1,\ldots,m_j; \text{with } m_j \leq k_j+2n_j)\), being each system-level task already labelled as Isolated, Joint or Individual, and finishes when the Sensor agents obtain the sensor performance objectives of all sensor-level tasks that allow to accomplish each system-level task with its corresponding required performance objectives. Next, we will describe in two separated subsections the interactions between Fusion and Sensor agents to handle tasks not included in the negotiation process, namely Isolated and Individual tasks, together with the supervision process (subsection 4.2), and the negotiation carried out among Sensor agents to perform the Joint tasks (subsection 4.3). Before that, subsection 4.1 summarizes the communication mechanism implemented in all agents and the internal control rules.

### 4.1 The Fusion and Sensor Agents Communication Level and Agents Control Rules

As mentioned in 2.1, each agent’s architecture follows a generic structure with three levels: Execution, Planning and Communication. The Communication level is based on an agenda containing linguistic acts that, when processed, generate messages to other agents or modify the internal state of the agent. To do that, this level is divided into two main elements (see Fig. 10): the first one stores the agent internal state and the second one manages the inter-agent negotiation process.

![Figure 10: The Communication Level architecture.](image)

The information stored as internal state represents both the real current agent situation, provided by the reasoning process performed at the Planning level, and the required agent situation (demanded tasks that should be performed by the sensor agent and their performance objective). Besides it
includes the information sent by other agents about their situation.

The management of this information, both the reports to send and storage of received messages, and the time-scheduling of the actions (named acts) that support the negotiation process among sensor agents is carried out by means of an agenda structure [LS93], which is the core of the agent Communication Level. Different kinds of events result in the inclusion of acts in the agenda (waiting for activation):

- Communication of demanded/proposed action(s) to other agents (REQUEST.ORDER, REQUEST.PROPOSE acts).
- Reception of demand(s) for action(s) from other agents (REQUEST_TO_DO acts).
- Reception/Communication of internal agent(s) state information (INFORM, SUPPLY_INFO acts).
- Necessity of sharing the execution of some sensor-level task(s) with other Sensor agents (REFINE acts).
- Necessity of subdivision of action(s) into subaction(s) (such as internal DO acts).

When the agenda control mechanism decides to activate an act, some information is transmitted from/to the Communication Level to/from the Planning Level or a message is sent to other agent(s). Besides, the knowledge of how to subdivide actions into subactions is stored in another element of the Communication Level, the Skills base (see Fig. 10).

The rule base that drives the ordering of acts in the agenda determines the negotiation process among agents, as will be detailed in 4.3. The structure of high level (petitions) and low level (agent state information) messages exchanged among agents during the negotiation process is similar to the messages structure of CooperA [SAL89]: identifier, type, content, sender and receiver.

Finally, the internal order of activation of pending acts in each agent is determined by the control rules. The proposed scheduling strategy is FCFS (First Come First Served) with priorities, being the priority between acts of the same type given by the corresponding task priority. The set of control rules (applied sequentially for Sensor agent \( S_j \)) are:
1. Process $\text{INFORM}(T_i, P_{S_j}(T_i), O_{S_j}(T_i))$ acts of Isolated tasks $T_i$, to let the Fusion agent know that the task has not been performed. This situation is detected by the Sensor agent after a certain number of delays in the (not performed) task execution.

2. Process $\text{REQUEST\_TO\_DO\_ORDER}(T_i, P(T_i), O(T_i))$ acts of Individual tasks. The Sensor agent should perform the task as soon as possible because only one sensor may execute it.

3. Process $\text{REQUEST\_TO\_DO\_PROPOSE}(T_i, P(T_i), O(T_i))$ acts and its subacts.

4. Process the rest of acts in the FCFS way.

The subacts of $\text{REQUEST\_TO\_DO\_PROPOSE}$ acts wait for the reception of other agent(s) message(s) and will be processed when these message(s) are received, according to the task priority included in the $\text{REQUEST\_TO\_DO\_PROPOSE}$ that originated them.

The set of rules that, applied sequentially, control the Fusion agent are:

1. Process $\text{DO}(T_i, P(T_i), O(T_i))$ acts generated as a result of the system-level tasks management process at the Planning Level. Each act will be divided into subacts that will generate the messages sent to each involved Sensor agent.

2. Process $\text{SUPPLY\_INFO}(T_i, \text{NOT } S_j)$ acts of Isolated tasks received from each sensor agent $S_j$ that could not accomplish them.

3. Process $\text{ACCEPT}(T_i, P(T_i), O(T_i))$ acts of Individual tasks.

4. Process $\text{ACCEPT}(T_i, P(T_i), O(T_i))$ acts of Joint tasks so that the Fusion agent updates the information about which sensor(s) are performing each joint task.

5. Process the rest of acts in the FCFS way.

Subacts generated when a DO act is being processed have the same priority of the parent DO act (valid for subacts of immediate execution and for subacts that are waiting for an (acceptation) answer from another Sensor agent). When these messages arrive to the Fusion agent, these last subacts are processed according to the system-level task priority included in the DO act.

### 4.2 Information/Supervision

- Isolated Tasks
Every sensor tries to execute all Isolated tasks it has been ordered whenever possible. If sensor $S_j$ could not accomplish an Isolated task, $T_i$, the sensor agent includes in its agenda an $\text{INFORM}(T_i, \text{Fusion})$ act which, when processed, will generate a message to the Fusion agent containing the information about the non-executable system-level task $T_i$. In that case, when this message arrives to the Fusion agent, its Communication Level puts a $\text{SUPPLY_INFO}(T_i, \text{NOT } S_j)$ act in its agenda which, when processed, will include this information into the Fusion agent Planning Level knowledge base.

- **Individual Tasks**

The management of Individual tasks takes place as follows: the Fusion agent, at each system management cycle, will decide for each Individual task $T_i$ the corresponding system-level priority, $P(T_i)$, and required performance objectives, $O(T_i)$, including in its agenda a $\text{REQUEST.ORDER}(T_i, S_j, P(T_i), O(T_i))$ act per Individual task, $T_i$, and sensor, $S_j$, that could accomplish it. When the act is processed, the Fusion agent sends this order to each sensor $S_j$. When the message arrives to a Sensor agent, $S_j$, its Communication Level generates a $\text{REQUEST.ORDER}(T_i, P(T_i), O(T_j))$ act which, when processed, triggers:

(a) the calculation/update of the sensor-level task priority, $P_{S_j}(T_i)$, based on the perceived local environment (with the described fuzzy reasoning process) and, by means of local tables, the sensor-level performance objectives, $O_{S_j}(T_i)$, and sensor-level task execution parameters as a function of $P_{S_j}(T_i)$, to reach $O(T_i)$;

(b) the generation of an $\text{ACCEPT}(T_i, P(T_i), O_{S_j}(T_i))$ act which, when processed, sends an answer message to the Fusion agent. In the case that the sensor-level task priority $P_{S_j}(T_i)$ were very different (fuzzy concept) from the system-level task priority, $P(T_i)$, the Sensor agent Communication Level would generate an $\text{INFORM}(T_i, P_{S_j}(T_i), \text{Fusion})$ act informing the Fusion agent about the discrepancy.

- **Joint Tasks**

Once the priority and system-level performance objectives for a Joint task, $T_i$, are obtained at the Fusion agent, this information and the set of sensors $\{S_j, j=1,2,\ldots,n, n \leq M\}$ that could accomplish task $T_i$ is sent to all the sensors in the set by means of $n$ messages $<\text{Fusion}, S_j, \text{REQUEST.PROPOSE}(T_i, S_j, P(T_i), O(T_i))>$ and $n$ messages $<\text{Fusion}, S_j, \text{INFORM}(T_i, \{S_j\})>$. 

24
These messages result from the processing of \( n \) acts \( \text{REQUEST.PROPOSE}(T_i, S_j, P(T_i), O(T_i)) \) and \( n \) acts \( \text{INFORM}(T_i, \{S_j\}) \) put in the Fusion agent agenda. Then, a DO act is generated at the Fusion agent to wait for the Sensor agent(s) answer(s) to each REQUEST.PROPOSE of each system-level task. Each REQUEST.PROPOSE message, once received, generates in the agenda of each Sensor agent, \( S_j \), the appearance of the acts \( \text{REQUEST_TO_DO.PROPOSE}(T_i, P(T_i), O(T_i)) \) and \( \text{SUPPLY_INFO}(T_i, \{S_j\}) \). Thus, the knowledge about all the Sensor agents that could participate in the negotiation process is available at all involved sensors by means of the execution of the SUPPLY_INFO acts. Since then, the Fusion agent does not participate anymore in the negotiation process, which will be detailed in the following subsection 4.3.

Both for Individual and Joint tasks, when an INFORM or ACCEPT message arrives from a Sensor agent, the Fusion agent data base is updated with the information about sensor-level priority and performance objectives of the accepted, now sensor-level task, and the waiting DO act compares the system-level (demanded) and sensor-level performance objectives. Once the execution of the system-level task is verified (the demanded performances are accomplished, which can be checked analysing the DF&SA outputs) all pending acts related to the task are killed.

4.3 Negotiation

The negotiation process to share Joint tasks among the \( n \) involved Sensor agents implies a set of REQUEST, ACCEPT, REFINE and REJECT acts, which, when processed, result in an interchange of messages depending on: each agent’s situation (internal and perceived), cooperation rules and skills.

The cooperation rules that determine the negotiation capability of Sensor agents (applied independently to each task \( T_i \)) are formally represented with a finite automaton with several states and transitions, depicted in Figure 11. The automaton is composed by five states: (1) **idle state** (each Sensor agent, \( S_j \), is initially considered at the idle state for task \( T_i \), waiting for information from the Fusion or other Sensor agent), (2) **negotiation state**, to share tasks when there exists a high sensor load, (3) **negotiation state**, to share tasks when priority discrepancy appears, (4) **reject state**, and (5) **accept state**. Each state (except state 1) defines a message to be sent to other sensor agents: refine, reject or accept. The transitions are represented by arrows and each transition
is defined by a received message (in bold capital letters) and the relations between the parameters of
the message and the agent internal parameters.

Figure 11: Finite Automaton to represent Negotiation

The full description of the cooperation rules, the linguistic acts of Communication Level and the
transitions in the finite automaton is detailed next:

1. If the locally inferred sensor-level priority for the joint task \( T_i \), \( P_{Sj}(T_i) \), is similar to the
proposed system-level priority, \( P(T_i) \), and the sensor is able to execute the task with the
required system-level performance objectives, \( O(T_i) \), then the task is accepted. So, the initial
REQUEST_TO_DO.PROPOSE act (for Track Update, Identification and Sector Search
tasks) is decomposed at Sensor agents into: a DO act, to verify if the sensor-level priority
(computed at \( S_j \)) is similar to the system-level priority of the task received (Sensor and
Fusion agents agree in its importance); and into two mutually exclusive acts to ACCEPT (if
both conditions are hold) or to REFINE (otherwise) the task. These acts are included in the
sensor agenda linked to their parent REQUEST_TO_DO.PROPOSE act. Once the DO act
is processed, one of both acts is activated and the other is killed. In the case of acceptance,
in Sensor agent \( S_j \) it is generated an ACCEPT(T_i, P_{Sj}(T_i), O(T_i)) act which, when processed
by \( S_j \), will send the following messages: acceptance messages \(<S_j, S_k, ACCEPT(T_i, P(T_i),
O(T_i))>\) to all Sensor agents, \( \{S_k\} \), that were included in the INFORM message originally
sent from the Fusion agent, and to the Fusion agent: \(<S_j, \text{Fusion, ACCEPT}(T_i, P_{Sj}(T_i), O(T_i))>\). Then \(S_j\) comes back at the idle state for \(T_i\) until next management cycle.

2. In the case that sensor-level priority is similar to system-level priority (\(P_{Sj}(T_i)\) is similar to \(P(T_i)\), which is fuzzy relation between fuzzy concepts), but the sensor is not able, due to a sensor high load, to execute the task with the required system-level performance objectives, \(O(T_i)\), then the Sensor agent proposes to share the task, moving to state (2). In this case, Sensor agent \(S_j\) generates a \(\text{REFINE}(T_i, P_{Sj}(T_i), O_{Sj}(T_i))\) act, where the demanded sensor-level performance objectives, \(O_{Sj}(T_i)\), are reduced with respect to \(O(T_i)\), but guarantee the achievement of \(O(T_i)\) in the case the task is shared. As a result of the execution of the \(\text{REFINE}\) act, a message \(<S_j, S_k, \text{REFINE}(T_i, P_{Sj}(T_i), O_{Sj}(T_i))>\) is sent to all involved sensors, \(S_k\), and a DO act is generated at \(S_j\) which, when executed, moves \(S_j\) into a negotiation state for \(T_i\) (state (2) in the automaton) which forces \(S_j\) to wait for (acceptance) proposals from the rest of Sensor agents.

3. While Sensor agent \(S_j\) is in state (2), it compares, through a waiting DO act, the sensor-level performance objectives received from other Sensor agents \(S_k\), \(O_{S_k}(T_i)\), with the own objectives sent, \(O_{Sj}(T_i)\). The sensor will make as many propositions as necessary, varying the sensor-level performance objectives, until the task is finally shared. If after a reception of the \(\text{REFINE}\) message sent by \(S_j\), sensor-level task \(T_i\) is accepted by a sensor \(S_k\) (similar sensor-level priority of the task at both agents and performance objectives demanded by \(S_j\) can be achieved by \(S_k\)) the Sensor agent \(S_k\) generates an \(\text{ACCEPT}(T_i, P_{Sj}(T_i), O_{Sj}(T_i))\) act whose result, when processed, is an acceptance message sent to the Fusion agent and to each sensor included in the original \(\text{INFORM}\) message. When this \(\text{ACCEPT}\) message is received by \(S_j\), it also generates an act in its agenda of the form: \(<S_j, \text{Fusion, ACCEPT}(T_i, P_{Sj}(T_i), O_{Sj}(T_i))>\). Otherwise, if the sensor \(S_j\) that sent the \(\text{REFINE}\) message receives, from all involved sensors, \(\text{REJECT}\) messages or \(\text{REFINE}\) messages with similar priority but \(O_{Sj}(T_i) > O_{S_k}(T_i)\) for all \(k\), \(S_j\) will put a new \(\text{REFINE}\) act in its agenda with a smaller \(O_{Sj}(T_i)\), even if \(O(T_i)\) cannot be achieved, and will remain in the negotiate state (2). When at least one of the \(\text{REFINE}\) messages received has \(O_{Sj}(T_i) \leq O_{S_k}(T_i)\), \(S_j\) will generate an \(\text{ACCEPT}(T_i, P_{Sj}(T_i), O_{Sj}(T_i))\) in its agenda, coming back to the idle state for \(T_i\).

4. If, being a Sensor agent in idle state (1), the requested task has a system-level priority very different (greater) from the local sensor-level priority, the sensor proposes to share the task
in order to reserve its capacity (for other tasks in the two previous cases indicated above). In this case, the Sensor agent $S_j$ generates a $\text{REFINE}(T, P_{S_j}(T_i), O_{S_j}(T_i))$ act, where the sensor-level performance objectives required, $O_{S_j}(T_i)$, are modified (reduced). The execution of the $\text{REFINE}$ act at $S_j$ results again in the messages $\langle S_j, S_k, \text{REFINE}(T, P_{S_j}(T_i), O_{S_j}(T_i)) \rangle$ sent to all involved sensors and a DO act that this time will move $S_j$ into the negotiation state (3) for $T_i$.

5. Being the sensor in state (3), the $\text{REFINE}$ proposal made by $S_j$ will be accepted by other Sensor agent $S_k$ only if the proposed sensor-level performance objectives, $O_{S_j}(T_i)$, are similar to its own objectives, $O_{S_k}(T_i)$. In that case, Sensor agent $S_k$ will generate an $\text{ACCEPT}(T, P_{S_k}(T_i), O_{S_k}(T_i))$ act which, when processed, makes $S_k$ to send an acceptance message to all sensors included in the INFORM message sent by the Fusion agent. In this case, an $\text{ACCEPT}(T, P_{S_j}(T_i), O_{S_j}(T_i))$ act will be executed by $S_j$ when it receives this acceptance moving to state (5). Otherwise, if the task is rejected by any sensor, $S_j$ moves to rejection state (4). In the case that $S_j$ receives rejection messages $\langle S_k, S_j, \text{REJECT}(T, P_{S_j}(T_i), O_{S_j}(T_i)) \rangle$ from all sensors $\{S_k\}$ in the INFORM set, Sensor agent $S_j$ finally will have to accept the task, generating an $\text{ACCEPT}(T, \text{Fusion}, P_{S_j}(T_i))$ act, and coming $S_j$ back to the idle state for $T_i$.

In all cases, the $\text{ACCEPT}$ act modifies the sensor-level task priority and the sensor-level performance objectives of the accepted sensor-level task at the acceptant Sensor agent, setting them to those values received in the PROPOSE or REFINE request.

5. PERFORMANCE EVALUATION

The proposed management architecture has been evaluated by means of a simulation platform to generate multitarget multisensor scenarios. A scenario composed by two multifunction radars has been considered, with the first sensor, $S_x$, located on the x axis at 400 km from the coordinate origin, and the second sensor, $S_y$, located on the y axis at 400 km from the coordinate origin. Each sensor coverage is divided into 32 sectors with the same dimensions in range, azimuth and elevation. In Figure 12 the projected sectors onto the horizontal XY plane are shown (the grey color
represents the Sx sensor coverage and the white color the Sy sensor coverage). The fusion centre is located at the coordinates origin.

![Figure 12: Horizontal projection of Sx and Sy sectors](image)

A target database has been defined for the analysis of the system. From this database, a subset of trajectories has been chosen to illustrate the system performance in representative situations of interest: (a) different targets that follow a similar trajectory and represent an enemy group (t4, t5 and t6 in Figure 13); (b) a missile (t10 in Figure 14).

![Figure 13: Group Trajectory for Evaluation](image)
The system performance has been analysed measuring the evolution of system and local priorities with time, and the result of the cooperation. The first measure is the priority value computed in each agent (sensors and fusion centre), to validate the knowledge based fuzzy system. The priority value for each management cycle shows the behaviour of the fuzzy system and its similarity with the evaluation that a human operator would perform. The second measure is the time elapsed between consecutive tasks in each sensor agent. This measure shows how negotiation leads to the execution of joint tasks as a function of the situation (in this case the available capacity in each sensor).

5.1 Evaluation of Fuzzy System for Track Update Tasks

To prove the system capacity to evaluate the Track-update priority of system tasks, a comparison between the priorities calculated by each sensor manager and that obtained by the fusion centre has been carried out. Figures 15 and 16 show the time evolution of the Track-Update task priorities for the target representing a missile, t10 and for target t6 in the group, all of them computed both at Sx, Sy and at the Fusion Centre. As we saw in section 3, the Fusion Centre obtains the system task priority for the next management cycle considering the final fused results, while each sensor only uses its local tracks derived from the own measurements available.
The Track Update task priority depends fundamentally on the target trajectory. Therefore, the priority value for missile t10 presents four different behaviours (see Fig. 14): (a) the target appears close to the fusion centre, (b) the target is directed towards Sx, (c) the target is directed towards Sy and finally (d) the target returns to Sx. In Figure 15, the cases (b) and (d) are clearly represented when only Sx scans the surveillance space. In these cases the local priority value for missile is high in segments (b) and (d) but it is very low during the rest of time. In an opposite way, while only Sy is performing the surveillance tasks, the track behaviour in segment (c) produces a high priority value. Clearly, all the behaviours are correctly represented (Fig. 15) when the two sensors and the fusion centre are coordinated; in this case, the global priority value is high, about 90, at first instants when the target appears close to the fusion centre, because the fuzzy variable [Relative Threat] has a high degree in <Very Near> and <Near>. After this situation, the priority value decreases to 60 and it keeps constant during the target manoeuvre in (b) and (c) behaviours; the target has a degree of [Potential Hostility] <Medium> or <High> but the value of [Target Threat] is <Medium> and [Relative Threat] is <Medium>. When the target speed is very high and the target heading is directed to Sx the priority value is very high, about 90, in this case is [Potential Hostility] and [Target Threat]
with a high degree in <Very High> and [Relative Threat] evolves from <Medium> to <Very Near>.

A similar analysis can be done for target t6 (Figure 16), now with only two behaviours: firstly the trajectory is near Sy and after the target is directed towards Sx. While only one sensor is scanning the zone where the target is moving, only one behaviour is considered, but the coordination of sensors in the fusion centre produces the global consideration of the target.

The priority calculated by the fusion centre is similar to the priority calculated in an isolated sensor when the target follows a trajectory that could be dangerous for it, but the fusion centre evaluates all the alternatives in order to obtain a global priority considering both the sensor positions and the fusion centre location. For targets t10 and t6, the most appropriate evaluation of the threatening possibility is carried out at the fusion centre.

5.2 Evaluation of The Negotiation Process

Figure 16. Track Update Task Priority for t6, in Sx, Sy and Fusion Centre.
In this section, it is analysed the time interval elapsed between two consecutive executions, versus simulated time, of the same task in different sensors. In next figures, it is presented under different sensor loads, and displayed in the y axis (in seconds, upper for Sx and lower for Sy), considering the duration of system management and sensor management cycles both of 15 seconds.

Figure 17: Sensor-level Track Update tasks on Missile, with Sx and Sy under similar load

As it is shown in Figure 17, at the beginning of the scenario, Track Update task for the missile $t_{10}$ is executed by the Sx sensor, because the target can only be detected by this sensor (Isolated task). After, the missile Track Update task is executed only by the Sy sensor (notice that the target then is directed towards Sy). At this moment, the system-level priority value is similar to the Sy sensor-level task priority value, and therefore the missile Track Update task is transferred to this sensor (because the demanded system-level sensing performance objectives can be guaranteed by Sy). When the missile is manoeuvring to point to sensor Sx, the execution of its Track Update task is carried out back again by the Sx sensor, which also maintains the system-level performance objectives.
In Figures 18-20 it can be seen that, while in common coverage, the tracking of targets t4, t5 and t6 is performed by both sensors. At the beginning, both sensors share the three system-level tasks with the same sensor-level priority. The negotiation process is assigning almost all the system-level Track Update task of t4 to Sx, that of t6 to Sy and that of t5 is being divided into two sensor-level tasks simultaneously executed. When the targets heading is directed towards Sx, its tracking is performed only by Sx (with highest sensor-level priority due to the approach geometry and with smaller illumination rate to maintain the system-level performance requirements).
A load increment of a certain sensor directly will modify the behaviour of all the Sensor agents in the net, although the rest of the Sensor agents do not perceive this load change. For example, if the load of sensor Sy is increased, the execution time instants of the sensor-level Track Update tasks shown in Figures 17 and 18 then evolve to the values shown in Figures 21 and 22, where it can be seen that sensor Sx executes all the tasks, leaving Sy manager at the idle state.

In Figure 21 it can be seen that, when the missile t10 enters the Sx coverage, this sensor has to execute it (Isolated task with high system/sensor priority), although the load of Sx is higher than Sy.
Figure 22: Sensor-level Track Update tasks on t4, when Sy is loaded double than Sx

If, instead, sensor Sx load is double than Sy, sensor Sy executes the tasks releasing sensor Sx, as it can be seen in Figure 22.

Figure 23: Sensor-level Track Update tasks on the Missile, when Sx is loaded double than Sy

In Fig. 23, sensor Sy takes over the execution of the missile Track Update task once the target enters its coverage, and sensor Sx has to execute the task when the target was only inside its coverage only. From the moment the target enters back sensor Sx coverage, sensor Sx resumes the execution of the task.

The time evolution of the value of the time interval between consecutive executions is analogous for Target Identification tasks, shown in Figures 24-26 for the missile target under different load situations. In Figure 24, when the target leaves sensor Sy coverage, the task does not return to be executed by sensor Sx since its priority is very low and the sensor Sx, that should accomplish, it is very loaded.
Figure 24: Sensor-level Identification tasks on Missile, with Sx and Sy under similar load

Figure 25: Sensor-level Identification tasks on Missile, when Sy is loaded double than Sx

Figure 26: Sensor-level Identification tasks on Missile, when Sx is loaded double than Sy
The same effect can be observed for Sector Search tasks. In Figures 27-29, the time interval between consecutive executions of search tasks is displayed under different sensor loads. The most loaded sensor accomplishes less executions than the other. In Figure 27, at the beginning, the task is shared by both sensors, thereafter sensor Sx executes it solely and, finally, the sensor Sy. Figure 28 only shows results for sensor Sy because sensor Sx does not perform any Sector Search task on that sector. In this case, the values of time intervals evolve in the same way that the priority of the system-level task. Finally, Figure 29 shows the execution of the task mainly by sensor Sy and partially supported by sensor Sx.

Figure 27: Sensor-level Search tasks on sector 20 of Sx/30 of Sy, with Sx and Sy under similar load

Figure 28: Sensor-level Search tasks on sector 20 of Sx/30 of Sy, when Sy is loaded than Sx
All the experiments show that the priority calculated by the autonomous sensors serves as a reference value to evaluate the degree of acceptance of a task proposed by the Fusion agent.

6. Conclusions

The experiments performed to evaluate the behaviour of the proposed decentralised management scheme prove that it is able to globally take profit of the sensor capacity to take decisions autonomously, reducing the amount of decisions taken at the fusion centre, therefore guaranteeing a better survivability of the surveillance net, while maintaining the same volume of timely executed tasks.

The efficiency of the multiagent negotiation process to optimally solve the task-to-sensor assignment problem and to compute the sensor-level performance objectives required to reach the required system-level ones resides in the avoidance of estimating the suitability of each task-to-sensor assignment. Instead, it is based on the knowledge at the sensors of the real values of their internal state (load) (not an estimated one at the fusion centre) and on the received task execution share proposals sent by other Sensors agents (by means a request-answer messages as a result of their acts).

Thus, many unnecessary calculations at the fusion centre disappear. On the one hand, Isolated tasks are transacted by each sensor, eliminating, for example, the calculation for Sector Search tasks of...
not shared sectors. On the other hand, it is not necessary to have an (up-to-date non existing) algorithm at the fusion centre that accomplishes the globally optimum task-to-sensor assignment, since the responsibility is assumed by the netted sensor agent team.

The knowledge of the surrounding environment available at each node of a surveillance sensor net is uncertain. In this work this knowledge and its uncertainty, essential for the sensor management process has been represented as fuzzy variables, applying fuzzy logic on them to deduce new concepts from which sensor tasks priority can be inferred. This fuzzy inference strategy has been applied both at each node of the surveillance net or at the fusion centre, turning the problem of multisensor operation optimisation into a problem of (multi)sensor management based on information fused at a symbolic level.

The evaluation of this reasoning system demonstrates the validity of the fuzzy approach to cope with the multisensor task management problem. It has been verified the capacity of the fuzzy system to infer proper decision under different situations, similar to those given by the interpretation of an human operator. The results demonstrate how the importance of the tasks, measured through their priority, allows the (automatic) manager to adapt sensors operation to changing surveillance situations.

7. References


