7-2007

Hierarchical Collective Agent Network (HCAN) for efficient 3 fusion and management of multiple networked sensors

Qiuming Zhu
University of Nebraska at Omaha, qzhu@unomaha.edu

Stuart L. Aldridge

Tomas N. Resha

Hierarchical Collective Agent Network (HCAN) for efficient fusion and management of multiple networked sensors

Qiuming Zhu [a,*], Stuart L. Aldridge [b,1], Tomas N. Resha [b,1]

[a] *Department of Computer Science, College of Information Science and Technology, University of Nebraska at Omaha, Omaha, NE 68182-0500, USA*

[b] *21st Century Systems, Inc., 6825 Pine Street, Omaha, NE 68106, USA*

Abstract

Agent-based software systems and applications are constructed by integrating diverse sets of components that are intelligent, heterogeneous, distributed, and concurrent. This paper describes a multi-agent system to assure the operation efficiency and reliability in data fusion and management of a set of networked distributive sensors (NDS). We discuss the general concept and architecture of a Hierarchical Collective Agent Network (HCAN) and its functional components for learning and adaptive control of the NDS. Sophistication of a HCAN control environment and an anatomy of the agent modules for enabling intelligent data fusion and management are presented. An exemplar HCAN is configured to support dynamic data fusion and automated sensor management in a simulated distributive and collaborative military sensor network for Global Missile Defense (GMD) application.

Keywords: Data fusion; Sensor management; Learning and adaptive control; Agent technology; Hierarchical collective agent network

1. Introduction

An increasing number of military systems employ multiple sensors with similar employment characteristics or different incongruent requirements on single or multiple platforms to concurrently perform distinct functions. Various missions and operating environments may require dynamic selection of the sensor operating mode, platform attitude, degree of autonomy, and network connections for optimal performance of the overall system. Several of these functions require feedback from the signal processing algorithms to the sensor management functions to optimize the allocation of resources between co-located sensors and sensors on other platforms in the network while carrying out the competing missions of surveillance, target detection, tracking, and discrimination.

Historically speaking, military sensor management and fusion was accomplished in the head of the operator. But, with the increase in sensor capabilities, modes, and volume of data produced; the workload increased exponentially and now overwhelms the warfighter [13]. Automated optimization tools are thus in great demand. These tools must recognize the interdependent networks from a network of functional elements including sensors, communication resources, processing nodes, and engagement systems while adapting to a variety of threats and environments. Useful tools for optimized sensor management must also ensure a minimum level of functionality when faced with threats and environments outside the design optimization space. A key concept to optimization is the DoD Joint Directors of Laboratories (JDL) designated Data Fusion Level 4,

* Corresponding author. Tel.: +1 402 554 3685; fax: +1 402 554 3400.
E-mail addresses: zhuq@unomaha.edu (Q. Zhu), stuart@21csi.com (S.L. Aldridge), tom.resha@21csi.com (T.N. Resha).
[1] Tel.: + 1 402 505 7901.

Process Refinement, for network-centric system-of-systems configuration, in that the "system" is the mission-configured platform and the related "subsystems" are the networked platforms, sensors, and communication links.

There has been significant research into data fusion and data integration of multi-sensor outputs over the years. However, there has been less research effort put into efficient sensor management concepts and algorithms. A cornerstone (albeit dated) work in sensor fusion, the seminal work of Multisensor Data Fusion by Waltz and Llinas, only touched upon sensor management in general terms [18]. The JDL Data Fusion Group formalized the Data Fusion Model and depicted the sensor management concept in a process refinement level that is essentially a feedback path from the other levels to allow system control and performance management [16]. The JDL Level 4 designation provides a path for optimized techniques at the other data fusion levels, especially in cases of multiple sensor control.

Our perception of past research in data fusion and sensor management is that most attention was applied to JDL Level 1 object assessment (track & ID) and shied away from Level 2 situation assessment and Level 3 impact (threat) assessment. It was thought that as Level 2 and 3 algorithms were actively pursued and more understood, then Level 4 Process Refinement would be just a final step to each level and therefore needed little research. We see it differently. The past approach ignores the reality that JDL Level 4 is not a natural progression from levels 1, 2, and 3. We see Level 4 as the key to the modern-day push for a system-of-systems concept. By way of a "systems" analogy, consider the human body as a "platform" with sensors, analysis, decision-making, and motive power to reposition its sensors. Now consider police detectives at a crime scene keeping all their senses open, looking around for clues, smelling, touching, quizzing witnesses, gathering data, and using their brains to visualize what may have had happened. The consummate detective's goal is to find out what really happened, which is often accomplished by means of fusing all available information and eliminating the impossibilities from consideration before drawing conclusions. This analogy is an engineering example of how a system controls separate systems that will bring together the individual sensors, effectors, and the coordinated continual control of analysis, assessment, decision, and command for an integrated purpose (goal and mission).

The sensor management role involves several tasks (more or less depending upon the researcher) [14]. Malhotra includes the tasks of generating options (sensor/task pairings), prioritizing those options, scheduling and communicating (cueing), and monitoring health and availability [13]. In [11] McIntyre stated that the sensor manager was expected to ease the operator workload by automating allocation, prioritizing measurements, aiding in data fusion, and supporting reconfiguration in the event of partial/total loss of a sensor. In general, the sensor manager should perform resource allocation (sensor/task pairing), sensor cueing (scheduling and communication), performance monitoring, and overall system management. However, the state of the research into automated sensor managers is not to the point where a completely automated application can perform the required tasks adequately. While intelligent software agents are perfect for the tasks of resource allocation and scheduling, the overall system management task is another issue.

An intelligent agent is a software component that functions continuously, proactively, and autonomously in a particularly designated environment [10]. Agent systems have many important attributes, such as a reactivity mechanism, an inferential capability, a collaborative behavior, a goal orienting and objective reaching striving, etc. [1,12]. Many intelligent control and decision support systems can be effectively constructed by employing agent-based techniques. For example, Knoblock and Ambite [9] reported an agent-based approach for information gathering from distributed resources. Because of its adaptive features, an agent-based approach is also suitable for the complex and dynamic system control. desJarins [2] described an agent model of autonomous learning in probabilistic domains. The model incorporates techniques for using the agent's existing knowledge to guide and constrain the learning process and for representing, reasoning with, and acquiring probabilistic knowledge. In agent-based software developed by Geri and Zhu [7], the agent helps users to initiate reasoning queries upon request from users and consequently form better decisions through a learning process. An approach for using agent-based software architecture for combat performance under overwhelming information inflow and uncertainty was introduced in [8,21]. Many of these agent systems employ multi-agents that perform either similar or quite different functionalities, in the concept of "system of systems."

Research in multi-agent systems has concentrated on domain-independent frameworks, standard protocol definitions, handling of uncertainty, and extensive models of collaboration [17]. Giampapa et al. [6] described a model of autonomous interoperation for agents operating in a multi-agent architecture. The model incorporates techniques for using the agent's existing knowledge to guide and constrain the interactions. Rodriguez and Poehlman [15] explored the use of multiple inference-driven agents cooperating over a network. Research on learning in probabilistic domains has a certain effect for agents representing, reasoning with, and acquiring probabilistic knowledge. It is important to note that information uncertainties can be handled effectively by the agent technique applying probabilistic models. A multi-agent system is able to provide an assessment for a set of strategies and advice on a coherent plan of military action under the constraints of operation efficiency and optimization. However, methods for solid information-theoretic model of agents learning, adaptation, control and collaboration that is critical to sensor management are still lacking.

Our *Sensor Manager* concept utilizes a sophisticated multi-agent collaborative structure called Hierarchical Collective Agent Network (HCAN). Combined with a feedback mechanism with which to gauge performance and drive system configuration, the HCAN can optimize the management of a networked distributive sensor (NDS) system in question and relative to other systems that would be affected on the platform. HCAN can also consider management at both the sensor level and the higher "system" level of the total platform capability and its mission. We applied the HCAN to sensor fusion and management tasks on a simulated Global Missile Defense (GMD) platform (interceptor, space-based, or airborne) to demonstrate the capability to optimize sensor management and/or adaptive processing. In this platform, the agents of HCAN continually monitor the singular and integrated performance of the system's resources, sensors, communications, and effectors. It recommends the best overall use of sensors resources to perceive and extract the information from the observations, and schedules all sensors and platform resources relative to its current mission and prime goal to accomplish the mission.

The paper is organized in the following way. Section 2 presents the basic HCAN architecture. We discuss its distinct features in the context of comparison with other multi-agent interaction and collaboration system topologies. Section 3 describes our HCAN configuration and its functional modules for the distributed sensor network management. Section 4 discusses an implementation of the HCAN in a simulated GMD application. Section 5 contains conclusion remarks.

2. Multi-agent cooperation architectures

This section explains why we think the HCAN architecture is more appropriate than other multi-agent system structures for sensor fusion and management.

A general understanding of multi-agent systems (MAS) is that (i) each agent has a partial capability to solve a problem, (ii) there is no global system control, (iii) data and knowledge for solving the problem are decentralized, and (iv) the agent computation is asynchronous [5]. In MAS, the agents need to work collectively so that, as a group, their behavior solves the overall problem without disruption, conflict, and glitches. When a task is assigned, the agents often need to find the other agents to collaborate with. Such a task is easy if they know exactly which agents to contact and at which location. However, a static distribution of agents is very unlikely to exist for most real world applications. For dynamic multi-agent systems, agents need to know how and where to find the other agents [6]. Proper structural topology thus plays a critical role in these MAS systems. The topology determines how the agents interact with each other, and how data and knowledge are shared and communicated among the agents. In [20], the authors studied three major MAS topology models according to the criteria: (1) the ways of activation, supervision, and communication between the agents; (2) the dependency of the agents to complete a task; and (3) the ways of sharing data, knowledge, and other resources. These models are shown in Figs. 1–3. For the purpose of ease of comparison, we give a brief description of these models here.

(1) *Web-like topology*: In a Web-like topology, every node has a connection to all other nodes, forming a complete graph, as shown in Fig. 1. Note that this topology does not necessarily mean that there are physical links between any two agents in the system. The topology is formed when a MAS employs an agent-invocation–activation scheme, or called request-and-service protocol, a blackboard kind of communication and task activation approach. In this topology, every agent can call other agents to perform a requested task, or to response to calls issued by other agents to perform specific tasks. That makes the agents seemly directly connected.

(2) *Star-like topology*: In a star-like topology, the activities of the agents are coordinated or administered by some supervisory (or facilitator) agents designated in the assembly. Only agents that have connections built and specified in the structure can interact with each other. That is, the agents are under more control and stipulation than those in the Web-like topology, where communication and cooperation among the agents are not brokered by one or more facilitators. The facilitators in a Star-like topology are responsible for matching requests from users to agents, with descriptions of the capabilities associated with the agents. A structural diagram of this topology is shown in Fig. 2.

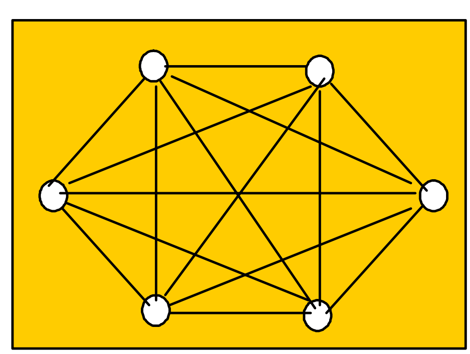

Fig. 1. Web-like topology of agent cooperation.

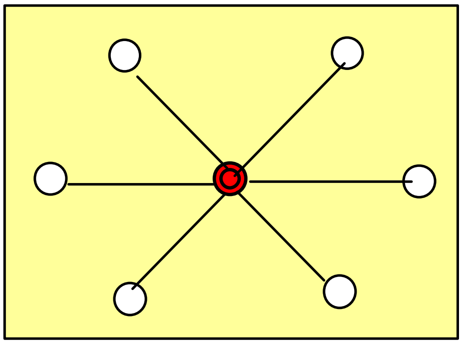

Fig. 2. Star-like topology of agent cooperation.

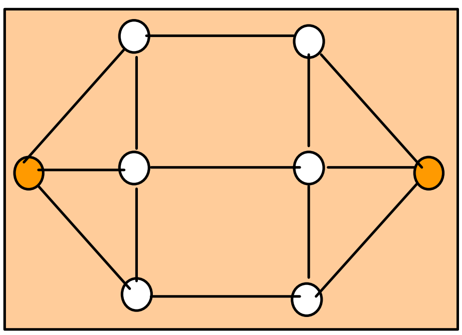

Fig. 3. Grid-like topology of agent cooperation.

(3) *Grid-like topology*: In a grid-like topology, each agent cooperates with a group of agents in its neighborhood (in terms of functional connections) that is a subset of agents in the assembly. That is, each agent has direct connections with a group of agents in its neighborhood (logically). Each group may be administered by a supervisor/facilitator. Interaction among agents not in neighborhood must pass through the neighboring agents in cascade. This is more like the concept of "system of systems." Fig. 3 shows a diagrammatic illustration of this topology.

Each of the above topological models of MAS has advantages and disadvantages. Zhu et al. [20] gave a qualitative assessment of the above three models in terms of their capability of facilitating intensive knowledge embedding, accumulation, and incorporation. They found that the Web-like topology associated with its indiscriminative behavior of agent activation is often inefficient, and many times undesirable. In the star-like topology, though control and coordination limits the boundary of cooperation the agents can reach, it is desirable when efficiency of cooperation needs to be ensured. The star-like topology is suitable for an environment and applications where part of the MAS is to act as a central planner that involves team negotiation and awareness of what each agent knows, needs, and does. On the minus side, there is the potential for a facilitator to become a communication bottleneck, or a critical point of failure.

In [20], a fourth topology, named Hierarchical Collective Agent Network (HCAN) model is also presented. The HCAN, as shown by diagram in Fig. 4, possesses the properties that (1) Agents are grouped in layers, (2) The layers are organized in hierarchy, (3) Agents in each layer are weakly connected, (4) Agents between layers are strongly connected, and (5) The control and coordination of the agent at each layer are carried out through to the agents at the higher level. Whereas "weakly connected" means that interactions between the agents are mainly data communications only, no control function (call or instruct) takes place there, while "strongly connected" means that agents on the two ends of the link have both data exchange activities and control relations (e.g., client and server, mediator and mediatee, etc.).

The collective nature of the agents in the HCAN paradigm overcomes some of the difficulties of the other agent system topologies. For example, it relieves the burden of intensive data-exchange between fellow agents in star-like topology by limiting agent communication to vertical layers of the assembly only. The collective nature of agent relation in the hierarchical architecture simplifies the functional design of the agent interactions and enhances the security and efficiency of the information processing, an advantage over the Web-like and Grid-like topologies. The HCAN architecture thus strikes a balance between the centralized control and distributed computation by allowing distributive agent operation within layers of the hierarchy and enforcing centralized control between the layers of the hierarchy, thus creating a federated agents integration structure.

In most applications, the agents in a MAS need to be responsible for on-site analyses of the collected data and extraction of information that is useful for the control agent to coordinate the actions of the distributed agents or agent groups. The HCAN architecture facilitates these operations. Basically, there are three types of agent interaction control schemes that can be enacted in a HCAN:

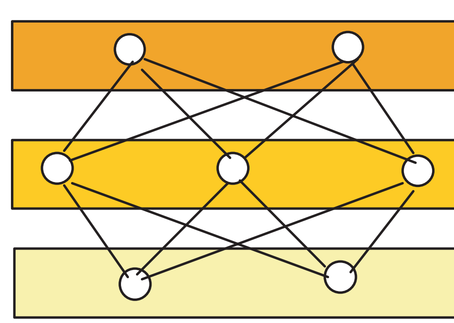

Fig. 4. HCAN topology of agent cooperation.

(1) *System-centered control*: In this control scheme, the system control agent at a high level knows and determines what actions/sub-actions each agent is to perform at certain time and place. The agent can employ a traditional control invocation scheme. Obviously, this schema is not advantageous for maximal utilization of the agent functionality and the autonomous abilities of the agents.
(2) *Agent-centered control*: In this scheme, each agent knows its responsibility, and interactions with other agents when necessary. That is, the agents coordinate the interactions by themselves autonomously within its group or scope of cooperation. This is an object-oriented control approach.
(3) *Request-for-service-centered control*: In this scheme, either the central control agent or an individual agent can issue requests to all other agents specified in the problem domain in situations when cooperation is needed. The invoked agent performs the service requested, or issues requests to other agent to cooperatively accomplish the task. This is a hybrid approach of the above two.

In the HCAN architecture for sensor fusion and management, the agents at the collective level only accept control from agents at the higher level. Hybrid system control law for this application in sensor management requires that directives to multiple platforms must be synchronized or chaos may occur. Mass effects requirements would further exacerbate the problem since very tight synchronization is required in planning and executing sensor allocation and re-allocation. While a centralized, coordinated operation of the agent group is essential and needs to be strengthened, it is equally important to emphasize and retain a high level of agent autonomy. Thus, the HCAN in the sensor management application will function somewhat differently versus those equivalent components in totally centralized or totally autonomous agent control settings (e.g., the other three topologies discussed above). One distinction is the communication aspect. The HCAN will engage in either a one-to-one, direct-line connection schema or in the entirely open broadcasting approach, and switch according to the specific situation detected by the sensor monitoring agents. The HCAN will also switch between an action–prediction based control strategy and an action–response based strategy [17]. In the action–prediction based control strategy, the HCAN makes predictions of the possible future states of the system upon sensing the battlespace state changes (via the situation assessment process) and applying pre-acquired knowledge in analyzing the collected information, and convey the predictions to involved agents along with the state reports. In the action–response based strategy, the system simply chooses a best reaction alternative upon sensing the battlespace state changes and conveys the state report to involved agents. The action–response strategy would assume more agent autonomous responsibility while the action–prediction strategy provides more information to the agents, though the predictions may not be thorough and perfect. The system control agent of the HCAN decides on which control strategy to use according to the situation assessment and according to its goal of optimizing the overall sensor management functions.

In the following sections, we describe in more detail an application of the HCAN architecture for sensor fusion and management. In performing the sensor fusion and management tasks, the agents assembly will be in charge of determining registered sensors in field, cuing applicable sensors to obtain additional information about objects, take data from various sources and combine them into fused object information, acquire relevant target information, learn better observation and tracking strategies, and provide real-time decision support for the sensor control and management operation.

3. HCAN for sensor management

It is noted that in a typical sensor fusion and control process, a number of functions need to be performed at different levels. Three levels of agent functions are identified in our HCAN implementation of the process. The first is a sensor data acquisition level. It is at this level that connections to the various sensor resources are made. Agent modules are needed to automate the information retrieval and integration from heterogeneous sensor resources. The functions in these modules will also provide an effective means for extracting useful information from the sensor resources and perform filtering operations. At the second level, the reasoning module takes the filtered data from the data acquisition level, performs various correlation and association functions, and distills the data collections. The outcome of this level contains information useful toward target detection, situation awareness, learning and sensor control, as well as representations of decision supporting knowledge. Finally, a control and adaptation level is at the top of the agent hierarchy. The user interface and visualization module of this level facilitates the task coordination and performance monitoring functions of the overall system.

The three level architecture of our HCAN system for sensor network management is illustrated in the block diagram of Fig. 5. In this architecture, as pointed out above:

(1) Agents at the lower level interface directly to the sensor environment and monitor the sensor operations. These agents collect sensor state parameters

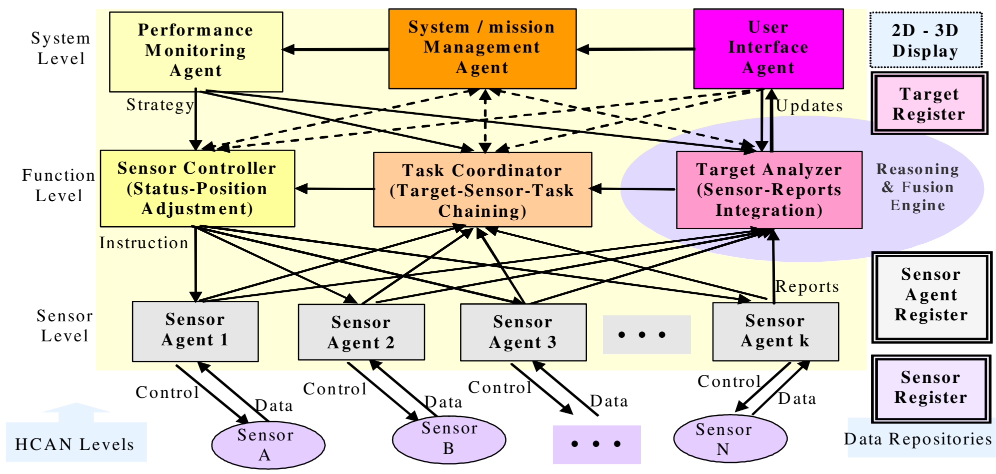

Fig. 5. Overall HCAN system blocks for distributed sensor network management.

and receive control feedback for sensor state adjustment. These agents act in a distributive fashion.

(2) Agents at the function levels will apply analytic models and reasoning-integration techniques to make decisions for sensor state control and adjustment.

(3) Agents at the system levels coordinate the sensor management activities of the agents at the lower levels. These agents interface with users as well as receive situation assessment inputs.

The three-level HCAN architecture for multiple networked sensor fusion and management consists of seven different types of software agents and three main data depositories. The seven agent types are:

(1) Sensor Agents (SA), which are directly connected to the networked sensors for receiving target detection data from the sensors and sending sensor control commands. In this sense, the sensor agents also act as the sensor actuators. There are multiple sensor agents in the HCAN, one for each sensor.

(2) Target Analyzer (TA), which is essentially the Sensor Fusion Agent. All sensory data are fed to this agent and is processed for target validation and identification. It sends target data to the User Interface for display (and supporting the user) and send sensor assignment/adjustment requests to the Task Coordinator (cueing).

(3) Task Coordinator (TC), which determines what Sensor Control and management tasks need to be accomplished. It also finds and allocates proper sensors to specific target, or FOV (Field of View) for Sensor-target pairing/tracking coordination.

(4) Sensor Controller (SC), which receives directives and requests from both the System Management Agent (SMA) and the Task Coordinator (TC), generates proper Sensor Control instructions, and sends the instruction to individual Sensor Agent for execution (Sensor status, parameter changes, and cueing).

(5) User Interface (UI), agent which is at the system management level for directly interacting with users. It is responsible for providing users a single picture of the situation awareness for the space covered by the NDS.

(6) System/Mission Management Agent (SMA), which keeps track of the overall mission objectives and ensures that the sensor management/control actions are consistent with the overall system/mission management strategies and priorities.

(7) Performance Monitoring & Adaptation Agent (PMA), which oversees the system activity and performs parametric learning and system adaptation functions that will affect the performance of the agents at all level of the system.

Some of these agent modules are to be described in more detail in this section. All agents in the HCAN architecture use a "*publish-and-subscribe*" model for data communication and agent interactions. There are three data repositories (registers) that are maintained and used by the agents in the HCAN architecture. They are: (1) Sensor Register (SR), (2) Sensor Agent Register (SAR), and (3) Target Register (TR). Each register is administered by an agent for performing data entry/retrieval (responding to requests from other agents), content updating, storage optimization, consistency checking, and database maintenance operations.

The Sensor Register (SR) contains a list of sensor devices, types, characteristics, deploy parameters (Position, Orientation, Scope, etc.), and their assigned Sensor Agents, as shown in Table 1.

In this table, the field "Sensor ID" gives a unique identification for each sensor deployed in the manage-

Table 1
SR data entries

Sensor ID	Type	Characteristics	Deploy parameters (P, v)	Corresponding sensor agent (ID)

ment space. The "Type" field gives a denotation for the nature of the sensor, such as if it is a ground Radar, a Satellite Infrared or optical detector, or others. The "Characteristics" field contains a more detailed description of the sensor device, for example, the detection range of the sensor, line of sight (LOS) or field of view (FOV), etc. The "Deploy parameters (P, v)" records the deployment information about the sensor where P is for the geospatial position and v is for the moving velocity (in case of a satellite sensor v is an angular velocity) of the sensor. These parameters may change through time so they must be updated continuously. The filed "Corresponding Sensor Agent (ID)" records the current software sensor agent assigned to monitor the sensor. Note that this field may also change because the sensor may be assigned to different software agents in a long run of the sensor management process.

The SR creates and keeps a record of mapping from sensor devices in field to Sensor Agent in HCAN. It performs consistency checks via a cross projection to the SAR, and keeps track of the sensors in current deployment, and their assigned Sensor Agents. The SR needs to know and maintains updated information about the capability of each sensor deployed. The register is published by each individual sensor device (through System Management Agent) for registering a sensor in, and is subscribed by functional and system level control and coordinate agents. When answering queries about targets, it needs to go through the associated Sensor Agent to find a list of targets that are currently being tracked by this sensor device.

The Sensor Agent Register (SAR) maintains a list of sensor agents, their assigned sensor devices, and targets under watching and tracking. Table 2 shows the main data entries of this register.

Fields in the Sensor Agent Register (SAR) include the "Agent ID" which gives a unique identification of a software agent in the sensor management system, the "Associated Sensor (ID)" which indicates the physic sensor device that the agent is assigned to, and the "List of Target (ID) Under Tracking" which links the software agent to the target in track. Note that the "List of Target (ID) Under Tracking" needs to be dynamically updated in the sensor management process as time passes, and there could be multiple targets in one sensor's viewing/detecting range.

The SAR builds a mapping from the set of Sensor Agents to the set of sensor devices, and then to a set of Targets. It performs a consistency checking with a cross projection to both Sensor Register and Target Register, and keeps track of the Sensor Agents in current deployment (their assignment to sensors, and current targets in watching/tracking). The content of SAR is published by Sensor Controller for register the agent and sensor connections into the register, and is subscribed by Functional and System level control and coordinate agents. It needs to go through the Sensor Register entry to access the characteristics and deploy parameters of the associated sensors.

The Target Register (TR) maintains a list of targets under observation, the target parameters (ID, position, velocity, etc.), and their associated Sensors. Table 3 lists the data entries of the TR.

The fields "Target ID" in the above table gives a unique identification of a target being tracked. This "ID" is assigned by the software agent and will not conflict with other target IDs in the management space. The fields "Type" and "Characteristics" describe the physic nature of the target, while the field "Parameters (P, v)" records the current position and moving velocity of the target. The "list of Sensors (IDs) associated" links the target to specific sensors that are tracking this target or are in the tracking (detecting and viewing) range of the target. Note that this list needs to be dynamically updated in the sensor management process when the target enters into or leaves away from the sensor's viewing/detecting range as time passes.

The TR records a mapping between a set of targets and a set of Sensor Agents. It performs a consistency check in a cross projection to Sensor Agent Register, and keeps track of the targets under observation. The TR is published by the Sensor Fusion and Target Analyzer which is responsible for target discovery from sensor data integration. The TR content is subscribed by Functional and System level control and coordination agents. For sensor management function, the TR must know the identity of each target and a list of sensors that are currently tracking that target.

A redundancy does exist between the Sensor Register data entries and the Sensor Agent Register data entries. Each has a field for sensor ID and agent ID (or corre-

Table 2
SAR data entries

Agent ID	Associated sensor (ID)	List of target (ID) under tracking

Table 3
TR Data entries

Target ID	Type	Characteristics	Parameters (P, v)	List of sensors (IDs) associated

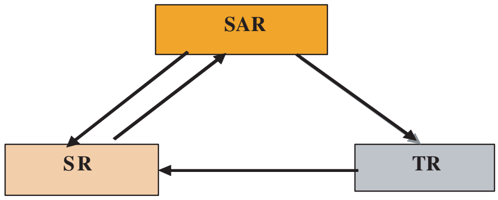

Fig. 6. Inter-relationship of the data repositories in the HCAN system.

sponding sensor agent). The rational to have this redundancy is for both computational efficiency and fault-tolerance considerations, of course, at the cost of memory space and maintenance of the fields. A relationship of the cross projection of the three registers can be illustrated by a diagram in Fig. 6.

There are many sources of uncertainty at different levels of the sensor fusion and management computations in the HCAN. For example, even if a situation-assessor is aware of the presence of certain objects in the operation space, such as the type of contact, intention, reaction rational, etc.; the exact dynamics of the object is still uncertain to the agents tracking the target. The knowledge about the object dynamics is critical in constructing an optimal strategy of sensor management action. Various statistical methodologies and knowledge discovery techniques may be applied in the reasoning module of the HCAN agents. The level of uncertainty forces the reasoning agents to operate with different decision strategies. Some of these agent functionalities are described below.

The Sensor Agents (SA) plays an important role in interfacing between the sensor network and the management system. Data from diverse sensor resources are filtered and preprocessed by the SA to a form that can be effectively used by the sensor control agents. The preprocessing and filtering operators are in charge of clearing up the noises and compensating for the uncertainties contained in the raw data. The interface is standardized such that its application can be ported to all classes of sensors with minimal installation and special interface rendering. A sensor agent can be assigned/allocated to different sensors (i.e, a sensor agent is NOT necessary to be tied to a specific sensor device all its life; it can be dynamically switched to tie with (be assigned to) different sensor devices. Of course, only one sensor device should be tied to one sensor agent at any time. The subscribe-and-publish functions of the SA are defined in Fig. 7.

The Target Analyzer (TA) invokes a *Reasoning & Fusion Engine* (RFE) to perform intelligent reasoning tasks to solve the dynamic re-planning, plan evaluation, and plan selection problems for sensor allocation and deployment in assigned mission states. The TA receives reports from multiple Sensor Agents (SAs), fuses sensor data from the multiple sensor resources and generates one track for each target from multiple sensor reports. It also identifies individual targets—associating targets from multiple sensors and resolving target ID conflicts. After these operations, the TA enters target data into the target register. If a new target is detected, it creates a new entry records the target parameters and its associated sensors in the register. If the target is associated with an existing one, it simply updates the target parameters in record. The TA will also send target data to the User Interface agent for display (and informing user), and send sensor assignment/adjustment requests to Task Coordinator. The subscribe-and-publish functions of the TA are defined in Fig. 8.

The Task Coordinator (TC) agent applies certain control strategies to guide the Sensor Control agent in sensor allocation and deployment planning/re-planning process. A set of goals and sub-goals are set up by the TC agent according to the sensor space situation, mission requirements, sensor operation parameters and function specifications, operator instructions, etc. From these data, the TC will analyze the situation and recommend the optimal course of action to subordinate level agents. From the analysis, the TC agent determines what Sensor Control and management tasks need to be done. It tries to find and allocate proper sensors to specific targets, or FOV for Sensor-target pairing/tracking coordination. It also finds specific position/orienta-

- SA_Subscribe( );
  - Gets information about sensor status and parameter change instruction from Agent Manager (at upper level).
  - Converts the instruction in XML format to Ad Hoc sensor control (status and parameter adjustment) signals.
- SA_Publish( );
  - Takes the raw sensor signal as inputs, detects and extracts basic target information, such as target position, speed, and possible type;
  - Generates XML report of the target, and transmit the report to *Target Analyzer*.
  - When a target is likely getting out of the scope of this sensor device, a notification/request is sent out to the Sensor Agent Manager to arrange for a handoff.

Fig. 7. The Subscribe-and-Publish functions of Sensor Agent.

- TA_Subscribe( );
  - Gets target reports (containing target parameters) from Sensor Agent (SA), and
  - Fuses the target reports (in XML format) with the target tracking entries (status and parameters) in Target Register.
    - o Check whether multiple reports reporting the same target (or targets are reported in multiple sensors) – association.
    - o Check whether targets in current reports are already registered in (TR) - correlation.
  - Check whether any target uncertainty/ambiguity exists.
- TA_Publish( );
  - Publish to Target Register:
    - o If a new target, then create a new entry in target register (TR), assign a unique ID, and enter the target status and parameters into the TR.
    - o If an existing target, update target entry in TR according to the new reports.
  - Publish to User Interface Agent:
    - o Report status and parameters of any new target entry for 2D/3D display.
    - o Update status and parameters of targets currently under tracking.
  - Publish to Task Coordinator:
    - o Report instance of target uncertainty/ambiguity.
    - o Send uncertain/ambiguous target's status and parameters, along with associated sensor information.

Fig. 8. The subscribe-and-publish functions of TA agent.

- TC_Subscribe( );
  - Subscribe to Target Register - to deal with handoffs.
  - Subscribe to Target Analyzer and Sensor Agents to receive requests for sensor control actions.
- TC_Publish( );
  - Publish Execution request along with requirement parameters to Sensor Controller.
  - Publish (generate, suggest) sensor control options to Sensor Controller
    - o Adjustment of Sensor parameter (e.g., re-orienting) to keep track of a moving target within a Sensor's rotation range;
    - o Switch of Sensors to handoff target when a target is moving out of the reachable and manageable range of a Sensor;
    - o Putting more Sensors in active service (if there are sensors available) when more targets come and the space becomes crowd.

Fig. 9. The subscribe-and-publish functions of TC agent.

tion parameter requirements for particular sensor to observe a specific target. The tasks determined by the TC are to be executed by the Sensor Controller. The actual (physical) execution of the sensor control is accomplished through the Sensor Agent, and further pass over to the Sensor device. The subscribe-and-publish functions of the TC are defined in Fig. 9.

The Sensor Controller (SC) agent receives directives and requests from both the System Management Agent (SMA) and the Task Coordinator (TC). It generates proper Sensor Control instructions, and sends the instruction to the individual Sensor Agent for execution (Sensor status and parameter changes). The functions performed by the Sensor Controller include assigning, distributing, and dispatching Sensor Agents to individual Sensors in service. The SC finds sensors that fit to specific function and position requirement, issues status and position parameters of the sensors and parameter changes to designated sensors. It will also be in charge of resolving Sensor-target tracking conflict in the handoff process, and optimizing sensor distribution and task assignment. The subscribe-and-publish functions of the SC are defined in Fig. 10.

The other agents of the HCAN function in the following ways. The User Interface (UI) agent connects sensor operators to the HCAN, and subsequently to the sensors. The agent will assist the reasoning and inference agent and the learning adaptation agent by receiving instructions and/or refutations about their sensor control decisions, and adjust (override) the sensor control parameters by applying certain control strategies that are aimed to improve the system performance.

The System Management Agent (SMA) is responsible for the synchronization of the sensor management operations among the agents in the HCAN. It constantly evaluates the available information about the states of the sensors, the locations, environment, and time schedules, and computes the probabilities on each of the objectives. When necessary information is provided by users, the SMA sets up a sensor management policy and a sensor control strategy (e.g., best-first, greedy, heuristic, etc.). It will then prioritize the sensor control

- SC_Subscribe( );
  - Subscribe to System Management Agent for control rules, strategies, priorities, and system constraints
  - Subscribe to Task Coordinator for Sensor control options and parameters
  - Subscribe to Sensor Agent Register for Sensor and Target information
  - Subscribe to Sensor Register - for looking up sensor capabilities
  - Resolve Sensor-target tracking conflict in sensor assignment and handoff processes.
- SC_Publish( );
  - Publish to Specific Sensor Agent for Sensor control action execution.
    - Assigns / distributes / dispatches Sensor Agent to individual Sensor in service
    - Find Sensors that fit to specific status/position requirement.
    - Issues status / position parameters and changed to designated Sensor.
    - Scheduling of the execution of Sensor Control actions.

Fig. 10. The subscribe-and-publish functions of SC agent.

tasks according to these priorities and control strategies—with respect to targets status and other system parameters and set up internal relations and compositions of sensors in the environment.

The Performance Monitoring & Adaptation Agent (PMA) is responsible for environmental analysis, and providing improvements to the control models and strategies used by the lower level agents (e.g., SC, TC and TA) for sensor management. In its role as a system performance and effectiveness monitor, the PMA is equipped with situation assessment and adaptation functions for system optimization. It also contains functions for supporting sensor reconfiguration in the event of partial/total loss of a sensor in an autonomous operating situation.

Based upon the priorities selected, the sensor state will change under the conditions such that the actions recommended by the agents tend toward optimizing the desired outcome. This optimization spans all possibilities and is computationally intensive. Considering realistic constraints, a heuristic model using a Bayesian and game theoretic approach will provide the real-time action/reaction necessary for multi-sensor operations. In order to drive the sensor configuration to optimality, a mixed strategy of Bayesian network representation and Bayesian Games is applied to the agents in HCAN. The process results from the optimization problem constrained to the set of stochastic kinematical differential equations describing the system behavior of the sensor's maneuver units and other involved components [21].

Among the agent modules in the HCAN structure for sensor management, the Task Coordinator agent and the Sensor Controller agent play the major role for sensor allocation planning/re-planning and optimization of the dynamical sensor deployment and adjustment. A performance monitoring capability and a feedback/optimization mechanism are implemented in the joint processes of these agents for process refinement. A control flow diagram of the process is shown in Fig. 11.

Most autonomous control systems are knowledge-intensive information processing ensembles. The same property is held by the HCAN. The stability and robustness of the sensor control is largely determined by the effectiveness and thoroughness of timely acquisition and utilization of accurate information from the sensors and all of the involved objects in the field. Correspondingly, factors that affect the control stability and robustness of these agents include information imprecision, incompleteness, and inconsistency. Communication among agents and between the central system and the agents thus is a critical aspect. In the HCAN, communication between the agents, between the agents and the sensors, and between the agents and human operators are processed and coordinated by the agents at the higher level of the HCAN. The communication can be carried in the ways of the following:

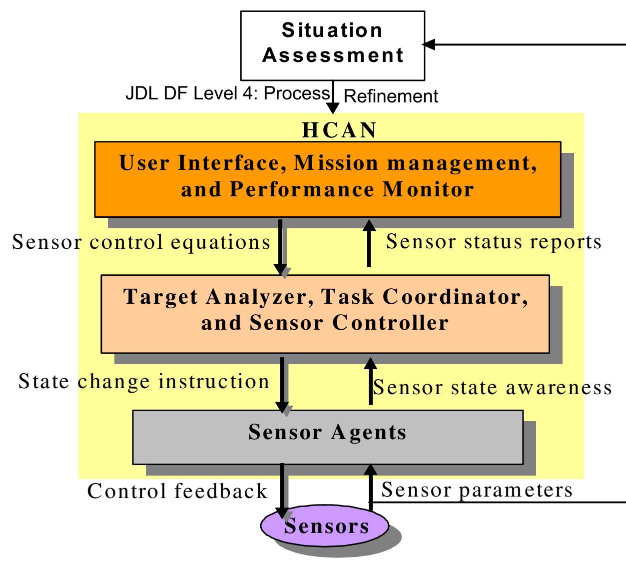

Fig. 11. Feedback control flow of sensor management process.

(1) *Private line communication*: This resembles the traditional way of parameter passing. Only the issuing and receiving agents know the communication has taken place. The advantage

is that it maximally limits the interference of other non-involving agents' activities. The disadvantage is that if the receiving agent is not responsible for, or incapable of, carrying out the requested task, the cooperation among the agents may be broken.

(2) *Blackboard communication*: This is also called party line communication. In this method, every agent has access to a common communication channel. Any task requests are posted to this channel and every agent responses to the call autonomously. If a request meets the pre-assigned duties or pre-specified parameters of an agent, that agent activates. The advantage of this approach is that it maximally guarantees the possibility of accomplishing the required task. The disadvantage is that it sometimes may still interfere other agents' activities, and waste system resources because the agents needs to periodically check and process the requests even they are not present.

(3) *Reserved-channel communication*: This is also called the mailbox method. In this method, a group of agents have an established agreement or protocol that specifies the locations (or frequencies) where communication signals will be transmitted to and accessed by the members of the group. This method is a compromise of the above two methods. Only agents within the group know the special places (or frequencies) where the information is posted. The advantage of this method is that it allows a proper allocation and reservation of system resources. The disadvantage is that it is difficult to identify the coherent group of agents that needs to share and exchange information within themselves exclusively.

In Section 4, we will present an implementation of these methods and approaches in a simulated environment for sensor fusion and management in a GMD application of the HCAN and its agent modules.

4. Experimentation

The ability to integrate and correlate a vast amount of disparate information from heterogeneous sensor and data resources with varying degrees of certainty in real-time is an impediment issue for mission-critical military decision support systems (DSS). For example, military commanders use multiple sensor/data resources and intelligence from reconnaissance and surveillance assets both in and out of a theater to build a whole picture of the battlespace in crucial military operations [8]. The commanders need to know and understand the relationships among the data, such as, what are the physical and functional constituency relations among the objects in a given geographic sector? Are there sequential or temporal dependencies of the objects and what will trigger them? What are the possible consequences of the action and re-actions? That is, decision making based on the situation assessment and impact assessment (SA/IA). These assessments are particularly important for identifying and prioritizing "gaps" between the operation planning and the real-time interactions.

In a mission-critical theater/situation demanding decision support, timely and accurate data fusion is a force multiplier. The lower-level data fusion from single or multiple sensor resources has become relatively well understood, resulting in accurate positional tracks and identification of physical objects. However, the processes for higher levels of data fusion, namely the level 2—situation assessment, and level 3—impact assessment (SA/IA), still requires the study and development of mathematically rigorous techniques and computational schemes. More in this realm is the level 4—process refinement which involves active control and management of the sensor resources. The kind of robust, integrated fusion architectures for handling increasing diversity of input sources are especially important in contemporary decision support missions. A well crafted software agent system integrating knowledge acquisition tools and proper decision support models can assist military operation planners in their tactical decision-making situations in many different ways, particularly with respect to quickly identifying responses and counter-responses to enemy action or inaction, providing a more current and more comprehensive picture to the field units.

We apply the above HCAN model to sensor management on a simulated GMD platform (interceptor, space-based, or airborne) to demonstrate the capability in sensor management and adaptive data processing. To accomplish the mission and schedules of all sensors and platform resources relative to its current mission and prime goal, we first conducted a system model analysis. The intent of this analysis is to hide the system dependent details and to abstract sensor information so as to form a basis for a formal specification of the sensor platform capabilities and their configurations. Care was taken to characterize the types of information provided by disparate systems in such a way as to make them compatible without making them sterile. This characterization is structured such that it's possible to determine complementary sensor characteristics and to allow the system to determine a sensor that can provide additional data leading to more accurate information, as opposed to duplicate data. The form of the characterization lends itself to rapid traversal to assist in the cueing process. For example, a tree structure or directed acyclic graph (DAG) based on sensor spectrum is more desirable than a straight list due to their speed in traversal.

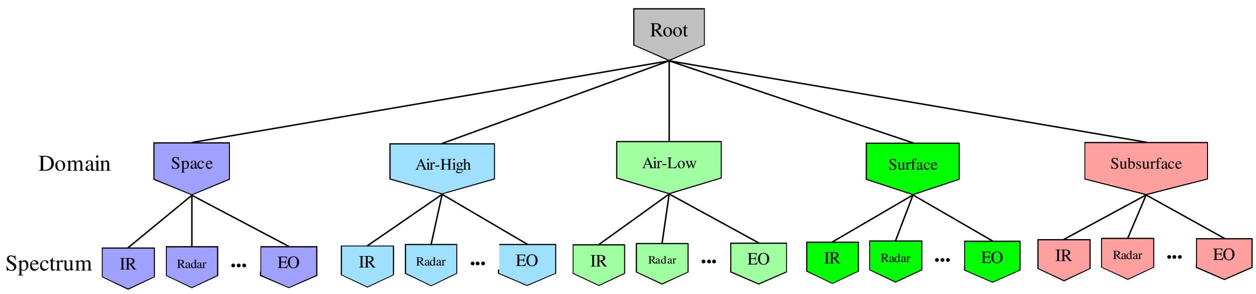

Fig. 12. Sensor deployment and classification diagram.

For the purposes of system specification, we chose to limit the sensor capabilities characterization to two levels of abstraction. Fig. 12 depicts a sample characterization. First, we divided a sensor operating environment into five realms: space, air-high, air-low, surface, and subsurface. The subsurface realm consists of the subterranean and underwater areas. A sensor is associated with a realm based on its sensing capability. For example, while a DSP satellite exists in the space realm, its sensing capability is targeted at the air realm. Many sensors will be associated with more than one realm (e.g., THAAD sensor).

The sensor monitoring agents in HCAN need to promptly sense and detect state changes of the sensor space, including the altering of tactical mission objectives, the switching of targets, the loss or gain of tactical forces and other assets in both adversary and own units, the relocating of the battlespace, etc. The main duty of the HCAN agents thus is to timely collect and promptly feedback the spatial situation and field sensor information to functional agents involved. In addition, the agents are also responsible for on-site analyses of the collected data and extraction of information that is useful for the control agent to coordinate the actions of the distributed agents or agent groups in the HCAN.

In additional to sensor control capabilities, there are also constraints associated with sensor detecting capabilities. Two of the most obvious are the line-of-sight (LOS) and range constraints associated with many sensors. But, there are more subtle constraints that must also be taken into consideration in the sensor management control mechanisms. Sensor platforms themselves may have resource management constraints (power, attitude, interference, orbit, time-on-station, etc.) associated with the platform itself. These constraints are also entered into the management schema. In a similar vein, constraints that occur as a result of a single platform having multiple sensors must also be considered (interference, resource limitations, etc.).

The result of this analysis is a specification for sensor configuration that incorporates capabilities and constraints of the sensor and its platform. The specification provides input to the next task, the development of the HCAN agent structure. Additionally, this task will lead to the development of a virtual multi-sensor platform mapped collectively in the HCAN.

Our implementation of the *HCAN Sensor Manager* is facilitated by using AEDGE®, a publish-subscribe agent architecture. The AEDGE® support active entities (agents of different types, simulation objects as well as functional objects) communicating over a software bus, cooperating and so on. Class and object hierarchies (inheritance) are employed. The agent modules are implemented in Java™, with Java AWT and Java3D for interfaces and JFC for common object specifications. We bounded the experimentations through the following networked sensor parameters:

- Number of sensors: 10 – 15 sensors (with 15 as the maximum).
- Sensor Platforms: all domains possible—Airborne, satellite, surface, and subsurface.
- Platform characteristics: mobile and fixed—support multiple sensors with issues related to range, attitude, placement, etc.
- Sensor types: Multiple (in order to show the utility of complementary spectrums i.e., radar pass off to IR pass off to EO pass to second EO)—Radar, Synthetic Aperture Radar (SAR), Infrared (IR), optical (EO), electronic support measures (ELINT).
- Sensor characteristics: Detection range specified by LOS (Line of Sight) and FOV (Field of View).
- Sensor deployment parameters: Location (3D coordinates, Ground, Mid-Air, Air, Upper-Air), velocity, and terrain.

For targets to be detected and monitored by the networked multiple sensors, we set the following parameters solely for the purpose of demonstrating the system feasibility.

- Number of targets: 10 max at a time.
- Target types: Missile, Aircraft, Land Vehicles, etc.

The capability can be significantly improved with a proportional increase in the quantity and complexity of the target parameters. In the simulation environment, we set up the situation in the following computational steps:

1. Defining an operation space (an AOI, that is the total area of interest—space where the sensors are to operate jointly, the same space the targets are to travel through—not the AOI of each sensor), which is a 3D box including space, air, land, and sea areas.
2. Designing targets moving across the operation space, in sequence, individually or in groups. Multiple targets occur, where each is controlled by a dynamic equation with its own parameters (position, velocity, trajectory, etc.) entering the monitoring space independently.
3. Visualizations of the operation space, sensor locations, target movements, sensor cueing, handoff, etc., are to be handled through the GUI development of the system. This piece was mostly derived by leveraging our previous work, the Sensor AEDGE application.
4. The HCAN mechanism performs the following actions upon the simulated inputs from the multiple networked sensors.
   (1) When system operation starts, a number of Sensors (A, B, C,...) and corresponding Sensor Agents and Platform Agents are deployed in place, registered in the Sensor Register and Sensor Agent Register, and shown on scenario display.
   (2) As new sensors enter the fray (their swath enters the AOI), new Platform and Sensor Agents are instantiated for each.
   (3) Each Sensor Agent monitors its assigned targets for events that will impact its ability to continue its monitoring function. These events are future loss of LOS due to terrain or the target leaving the sensor's range or FOV. The agent also monitors its FOV to see if any new targets are approaching the Area of Interest (AOI). The agent will trigger a user alert in this case.
   (4) The targets start to appear (also shown in display).
   (5) When a target enters into the FOV of Sensor A, it is picked up by the Sensor Agent in connection with the Sensor A.
   (6) The Sensor Agent sends an event about the specific target (target type, location, motion characteristics, Field of View (FOV), cross-section, range, etc.) to the Target Analyzer—a sensor-data fusion agent, and the Target Register.
   (7) The Target Analyzer aggregates target reports from multiple Sensors, identifies the target and its track, enters the consolidated target data into the Target Register, and sends the target data to User Interface for display. If uncertainty and ambiguity arises, send an event (request) to the Task Coordinator for sensor cuing, allocation, adjustment, or other proper actions.
   (8) When a target is projected to leave sensor A's FOV (due to range, loss of line-of-sight (LOS), communication failure, etc.), an event is sent out by the Sensor Agent of Sensor A to the Task Coordinator to arrange for a handoff.
   (9) The Target Analyzer also takes known targets and attempts to identify complementary sensors (sensors in a different spectrum) with appropriate range and FOV so they can glean additional information about the target.
   (10) In the case of a handoff (passing the target from like sensor to like sensor), the Sensor Controller checks with both the Sensor Register and Agent Register to identify an available Platform and a Sensor Agent to take over (handover) the task (target watching/tracking).

Coupling the results of our research with previous experience, we structured an environment to allow the determination of complementary sensor characteristics and allow the system to compare and select the appropriate sensor to provide additional data leading to more accurate information, as opposed to duplicate data. This form of the characterization lends itself to rapid traversal to assist in the cueing process. Fig. 13 depicts some screen captures of the implementation. The situation involves an AOI with surface and airborne ISR assets. The surface assets are an AEGIS cruiser (radar) and two Rapier sites (optical camera). The airborne assets consist of an E-2C Hawkeye (radar), an E-3B AWACS (radar), and an RC-135V/W RIVET JOINT (ELINT) aircraft. While not necessarily a realistic situation, the goal was to have ISR assets from different spectrums in order to validate the *HCAN Sensor Manager*'s ability to assign complementary sensor assets for continual tracking of targets.

Basically, the HCAN system in our GMD simulation for sensor fusion and management has the following functionalities.

(1) a flexible software architecture for accommodating system augmentation and evolutions;
(2) a powerful representation schema for accommodating heterogeneous forms of information;
(3) a diverse interface for various input resources, output formats, and human interactions;
(4) an ability of reasoning on incomplete and inconsistent information, and extracting useful knowledge from the data of heterogeneous resources;

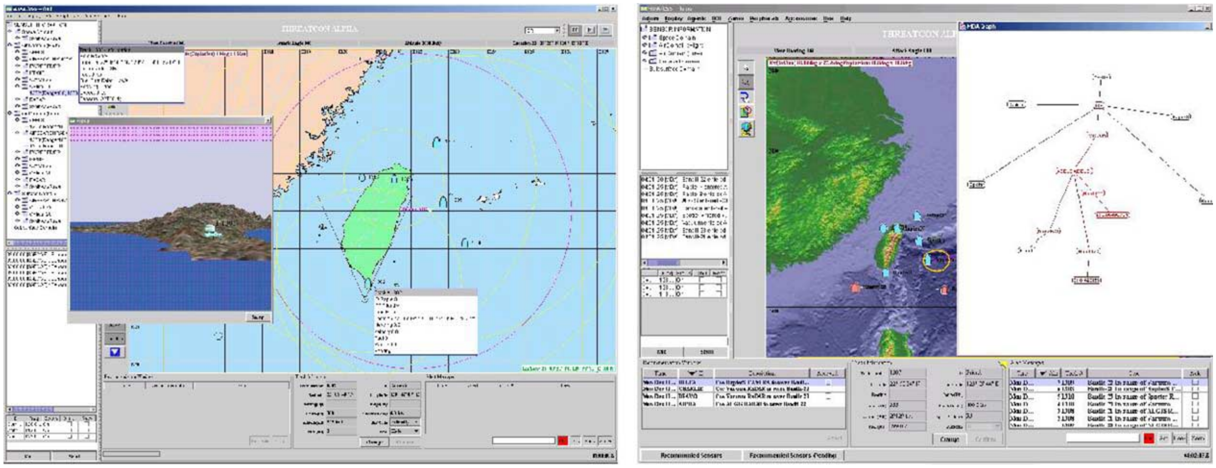

Fig. 13. Screen captures of the HCAN implementation of sensor management.

(5) an ability of incorporating real-time dynamics of the information resources into the system anytime during the operation, and promptly adjusting the reasoning mechanisms;
(6) an ability of summarizing and refining knowledge extracted, and distinguishing mission and time critical knowledge from insignificant and redundant ones;
(7) a capability of supplying meaningful and accurate explanations, both qualitatively and quantitatively, of the automated system actions; and
(8) a capability of providing adequate control and scrutinizing of the system operations under the environmental constrains of the given situation.

The expected performance improvements from employing the HCAN architecture for sensor management include the following:

- *Efficiency*: The system makes maximum use of onboard platform control and decision-making capabilities of the HCAN. The resulting software minimizes human intervention and enhances the self-sustainability of the multi-sensors autonomous operations.
- *Robustness*: The system is equipped with a self-diagnosis and certain self-repair, reconfiguration, and alternatives/backup capabilities through the embedded PMA modules and functionalities. The resulting software allows the multi-sensors' sustained and reliable operations even under partial impairment of the system.
- *Flexibility*: The system is empowered with high level of scalability and field adaptation ability. The HCAN-based control system re-organizes itself in different levels involving different numbers of components. It facilitates the control of multiple sensors to self-configure and operate either individually, in a group, or as a swarm and to interoperate in both manned and unmanned platforms.

5. Conclusion

The field of data fusion and sensor management can benefit significantly by focusing the major concerns on employment of agent-based technologies. Given the characteristics of most sensor fusion and management situations, it seems that one natural way to provide timely and critical support to the functions is to have a collection of distributed, autonomous problem solving intelligent agents working together on different aspects of the processes [4]. This research addresses the problems of how to make effective use of real-time information acquired from multiple and heterogeneous sensor and data resources, and reasoning on the gathered information for situation assessment and impact assessment through a hierarchical collective agents assembly organized in a network structure (HCAN). The system is to provide a refinement process (Fusion level 4) for time-critical missions in military operations, as well.

The hierarchically networked agent architecture of HCAN has three distinct features as compared to other multi-agent structures. These features are: (1) the agents in the HCAN assembly are organized with layered supervision rather than equal citizen type objects (though may function differently) [3]; (2) relations between agents in the HCAN assembly are collective in nature, resulting a soft-coupling between agents at the same layer of the network rather than hard-coupling (closely tied interactions) [6]; and (3) a goal-driven control scheme is employed to coordinate a top–down and bottom–up two-way iterative process for the agent-activation and interactions, rather than the conventionally adopted one way control approach [19]. The collective nature of the HCAN architecture allows for flexible addition or modification of the agents in the system because no complex de-coupling operations from the other agents at the same level (neighboring agents) are needed for the agents added or deleted. More importantly, the HCAN renders itself to a fault tolerant computing architecture, which is especially critical to sensor management operations. Since no tight coupling or coordination takes place among the agents at the collective agent level, every agent acts by their own under the supervision of the control agents at an upper level of the hierarchy. Thus, the agents at the collective level can be assigned to perform either different tasks or the same task at the same time, allowing for fault detection and functional back up.

The HCAN is flexible in terms of the ability in which communities of agents can be assembled, and the adaptation with which services can be added at runtime and brought into use without requiring changes to the other parts of the agent assembly. A unified set of concepts, declarations, and interfaces that are consistently configured across all services in the framework and the role played by the agents at different levels are defined. The HCAN architecture strikes a balance between the centralized control and distributed computation by allowing distributive agent operations within layers of the hierarchy and enforcing centralized control between the layers of the hierarchy, thus eases the coordination and control burden needed to manage interactions between agents. The worth of this concept lies in its applicability to many operational situations. From a single integrated air picture (SIAP) to an integrated intelligence preparation of the battlefield (IPB) application, the HCAN Sensor Manager concept can be applied without reengineering the core architecture. The intelligent agents that provide the decision support assistance can be tailored to the situational awareness and decision needs of the designated users. Additionally, users with different needs can have different decision support clients while using the same core data and architecture. We don't force a common picture; we provide a tailored picture based on a common situation.

References

[1] J. Bradshaw (Ed.), Software Agents, AAAI Press, 1997.

[2] M. desJarins, Representing and reasoning with probabilistic knowledge: a Bayesian approach, Uncertainty in Artificial Intelligence (1993) 227–234.

[3] K. Decker, K. Sycara, Intelligent adaptive information agents, Journal of Intelligent Information Systems 9 (1997) 239–260.

[4] Z. Elouedi, K. Mellouli, P. Smets, Assessing sensor reliability for multisensor data fusion within the transferable belief model, IEEE Transactions on Systems Man, and Cybernetics—Part B: Cybernetics 34 (1) (2004) 782–787.

[5] R. Flores-Mendez, Towards a standardization of multi-agent system frameworks. Available from: http://www.acm.org/crossroads/crew/roberto_flores-mendez.html.

[6] Giampapa, Paoluc, Sycara, Agent interoperation across multi agent system boundaries, in: Proceedings of 4th International Conference on Autonomous Agents, Barcelona, Sp, June 2000.

[7] S. Geri, Q. Zhu, dbAgent: an intelligent web agent for database mining, International Conference on Computer and Informatics (CS&I'98) (1998) 466–470.

[8] J. Hicks, A. Stoyen, Q. Zhu, Intelligent agent-based software architecture for combat performance under overwhelming information inflow and uncertainty, Seventh IEEE International Conference on Engineering of Complex Computer Systems (2001) 200–210.

[9] C. Knoblock, J. Ambite, Agents for information gathering, in: J.M. Bradshaw (Ed.), Chap. in Software Agents, 1997, pp. 347–373.

[10] M. Lejter, T.A. Dean, Framework for the development of multiagent architectures, IEEE Expert (1996) 47–59.

[11] G. McIntyre, A comprehensive approach to sensor management and scheduling, Dissertation, George Mason University, Fall 1998.

[12] P. Maes, Agents that reduce work and information overload, Communications of the ACM 37 (7) (1994) 30–40.

[13] S. Musick, R. Malhotra, Chasing the elusive sensor manager, Proceedings of the IEEE 1994 NAECON, 606–613, 23–27 May 1994.

[14] D.E. Penny, The automatic management of multi-sensor systems, in: Proceedings of the International Conference on Multi-source Multi-sensors Information Fusion, CSREA, 1998, pp. 748–756.

[15] B. Rodriguez, W. Poehlman, A system for distributed inferencing, in: Proceedings 1996 Rochester Forth Conference Open Systems, 1997.

[16] A. Steinberg, C. Bowman, F. White, Revisions to the JDL data fusion model, in: Proceedings of the 3rd NATO/IRIS Conference, Que., Canada, 1998.

[17] K. Sycara, D. Zeng, Multi-agent integration of information gathering and decision support, Proceedings of the 12th European Conference on Artificial Intelligence 1 (1996).

[18] E. Waltz, J. Llinas, Multisensor Data Fusion, Artech House Inc., 1990.

[19] Q. Zhu, Z. Chen, Goal-driven reasoning for knowledge discovery using belief networks, Proceedings of International Conference on Artificial Intelligence for Engineering (1998) 231–235.

[20] Q. Zhu, J.D. Hicks, P.V. Petrov, A.D. Stoyen, The topologies of knowledge intensive multi-agents cooperation, in: Proceedings of the IEEE Knowledge Intensive Multi-Agent Systems Conference, Cambridge MA, USA, October 2003, pp. 741–746.

[21] Q. Zhu, J.D. Hicks, et al., Bayesian-game modeling of C2 decision making in submarine battle-space situation awareness, in: The 2004 International Command and Control Research and Technology Symposium (CCRT2004), San Diago CA. CD#160, June 2004.