

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

Follow this and additional works at: <http://digitalcommons.unomaha.edu/compscifacpub>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Zhu, Qiuming; Aldridge, Stuart L.; and Resha, Tomas N., "Hierarchical Collective Agent Network (HCAN) for efficient 3 fusion and management of multiple networked sensors" (2007). *Computer Science Faculty Publications*. Paper 30.

<http://digitalcommons.unomaha.edu/compscifacpub/30>

This Article is brought to you for free and open access by the Department of Computer Science at DigitalCommons@UNO. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.



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*

10 Abstract

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

18
19 *Keywords:* Data fusion; Sensor management; Learning and adaptive control; Agent technology; Hierarchical collective agent network
20

21 1. Introduction

22 An increasing number of military systems employ
23 multiple sensors with similar employment characteristics
24 or different incongruent requirements on single or multi-
25 ple platforms to concurrently perform distinct functions.
26 Various missions and operating environments may re-
27 quire dynamic selection of the sensor operating mode,
28 platform attitude, degree of autonomy, and network
29 connections for optimal performance of the overall sys-
30 tem. Several of these functions require feedback from
31 the signal processing algorithms to the sensor manage-
32 ment functions to optimize the allocation of resources
33 between co-located sensors and sensors on other plat-

forms 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 opera-
tor. 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 sen-
sors, 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 environ-
ments outside the design optimization space. A key con-
cept 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.

53 Process Refinement, for network-centric system-of-sys- 109
54 tems configuration, in that the “system” is the mis- 110
55 sion-configured platform and the related “sub- 111
56 systems” are the networked platforms, sensors, and 112
57 communication links. 113

58 There has been significant research into data fusion 114
59 and data integration of multi-sensor outputs over the 115
60 years. However, there has been less research effort put 116
61 into efficient sensor management concepts and algo- 117
62 rithms. A cornerstone (albeit dated) work in sensor fu- 118
63 sion, the seminal work of Multisensor Data Fusion by 119
64 Waltz and Llinas, only touched upon sensor manage- 120
65 ment in general terms [18]. The JDL Data Fusion Group 121
66 formalized the Data Fusion Model and depicted the sen- 122
67 sor management concept in a process refinement level 123
68 that is essentially a feedback path from the other levels 124
69 to allow system control and performance management 125
70 [16]. The JDL Level 4 designation provides a path for 126
71 optimized techniques at the other data fusion levels, 127
72 especially in cases of multiple sensor control. 128

73 Our perception of past research in data fusion and 129
74 sensor management is that most attention was applied 130
75 to JDL Level 1 object assessment (track & ID) and shield 131
76 away from Level 2 situation assessment and Level 3 im- 132
77 pact (threat) assessment. It was thought that as Level 2 133
78 and 3 algorithms were actively pursued and more under- 134
79 stood, then Level 4 Process Refinement would be just a 135
80 final step to each level and therefore needed little re- 136
81 search. We see it differently. The past approach ignores 137
82 the reality that JDL Level 4 is not a natural progression 138
83 from levels 1, 2, and 3. We see Level 4 as the key to the 139
84 modern-day push for a system-of-systems concept. By 140
85 way of a “systems” analogy, consider the human body 141
86 as a “platform” with sensors, analysis, decision-making, 142
87 and motive power to reposition its sensors. Now con- 143
88 sider police detectives at a crime scene keeping all their 144
89 senses open, looking around for clues, smelling, touch- 145
90 ing, quizzing witnesses, gathering data, and using their 146
91 brains to visualize what may have had happened. The 147
92 consummate detective’s goal is to find out what really 148
93 happened, which is often accomplished by means of fus- 149
94 ing all available information and eliminating the impos- 150
95 sibilities from consideration before drawing conclusions. 151
96 This analogy is an engineering example of how a system 152
97 controls separate systems that will bring together the 153
98 individual sensors, effectors, and the coordinated contin- 154
99 ual control of analysis, assessment, decision, and com- 155
100 mand for an integrated purpose (goal and mission). 156

101 The sensor management role involves several tasks 157
102 (more or less depending upon the researcher) [14]. 158
103 Malhotra includes the tasks of generating options (sen- 159
104 sor/task pairings), prioritizing those options, scheduling 160
105 and communicating (cueing), and monitoring health and 161
106 availability [13]. In [11] McIntyre stated that the sensor 162
107 manager was expected to ease the operator workload by 163
108 automating allocation, prioritizing measurements,

aiding in data fusion, and supporting reconfiguration 109
in the event of partial/total loss of a sensor. In general, 110
the sensor manager should perform resource allocation 111
(sensor/task pairing), sensor cueing (scheduling and 112
communication), performance monitoring, and overall 113
system management. However, the state of the research 114
into automated sensor managers is not to the point 115
where a completely automated application can perform 116
the required tasks adequately. While intelligent software 117
agents are perfect for the tasks of resource allocation 118
and scheduling, the overall system management task is 119
another issue. 120

121 An intelligent agent is a software component that 122
123 functions continuously, proactively, and autonomously 124
in a particularly designated environment [10]. Agent sys- 125
tems have many important attributes, such as a reactivity 126
mechanism, an inferential capability, a collaborative 127
behavior, a goal orienting and objective reaching striv- 128
ing, etc. [1,12]. Many intelligent control and decision 129
support systems can be effectively constructed by 130
employing agent-based techniques. For example, Kno- 131
block and Ambite [9] reported an agent-based approach 132
for information gathering from distributed resources. 133
Because of its adaptive features, an agent-based ap- 134
proach is also suitable for the complex and dynamic sys- 135
tem control. desJarins [2] described an agent model of 136
autonomous learning in probabilistic domains. The 137
model incorporates techniques for using the agent’s exist- 138
ing knowledge to guide and constrain the learning pro- 139
cess and for representing, reasoning with, and 140
acquiring probabilistic knowledge. In agent-based soft- 141
ware developed by Geri and Zhu [7], the agent helps 142
users to initiate reasoning queries upon request from 143
users and consequently form better decisions through a 144
learning process. An approach for using agent-based 145
software architecture for combat performance under 146
overwhelming information inflow and uncertainty was 147
introduced in [8,21]. Many of these agent systems employ 148
multi-agents that perform either similar or quite different 149
functionalities, in the concept of “system of systems.” 150

151 Research in multi-agent systems has concentrated on 152
153 domain-independent frameworks, standard protocol 154
definitions, handling of uncertainty, and extensive mod- 155
els of collaboration [17]. Giampapa et al. [6] described a 156
model of autonomous interoperation for agents operat- 157
ing in a multi-agent architecture. The model incorpo- 158
rates techniques for using the agent’s existing 159
knowledge to guide and constrain the interactions. 160
Rodriguez and Poehlman [15] explored the use of multi- 161
ple inference-driven agents cooperating over a network. 162
Research on learning in probabilistic domains has a cer- 163
tain effect for agents representing, reasoning with, and 164
acquiring probabilistic knowledge. It is important to 165
note that information uncertainties can be handled effec- 166
tively by the agent technique applying probabilistic 167
models. A multi-agent system is able to provide an 168

165 assessment for a set of strategies and advice on a coher-
166 ent plan of military action under the constraints of oper-
167 ation efficiency and optimization. However, methods for
168 solid information-theoretic model of agents learning,
169 adaptation, control and collaboration that is critical to
170 sensor management are still lacking.

171 Our *Sensor Manager* concept utilizes a sophisticated
172 multi-agent collaborative structure called Hierarchical
173 Collective Agent Network (HCAN). Combined with a
174 feedback mechanism with which to gauge performance
175 and drive system configuration, the HCAN can optimize
176 the management of a networked distributive sensor
177 (NDS) system in question and relative to other systems
178 that would be affected on the platform. HCAN can also
179 consider management at both the sensor level and the
180 higher “system” level of the total platform capability
181 and its mission. We applied the HCAN to sensor fusion
182 and management tasks on a simulated Global Missile
183 Defense (GMD) platform (interceptor, space-based, or
184 airborne) to demonstrate the capability to optimize sen-
185 sor management and/or adaptive processing. In this
186 platform, the agents of HCAN continually monitor
187 the singular and integrated performance of the system’s
188 resources, sensors, communications, and effectors. It
189 recommends the best overall use of sensors resources
190 to perceive and extract the information from the obser-
191 vations, and schedules all sensors and platform re-
192 sources relative to its current mission and prime goal
193 to accomplish the mission.

194 The paper is organized in the following way. Section
195 2 presents the basic HCAN architecture. We discuss its
196 distinct features in the context of comparison with other
197 multi-agent interaction and collaboration system topol-
198 ogies. Section 3 describes our HCAN configuration and
199 its functional modules for the distributed sensor net-
200 work management. Section 4 discusses an implementa-
201 tion of the HCAN in a simulated GMD application.
202 Section 5 contains conclusion remarks.

203 2. Multi-agent cooperation architectures

204 This section explains why we think the HCAN archi-
205 tecture is more appropriate than other multi-agent sys-
206 tem structures for sensor fusion and management.

207 A general understanding of multi-agent systems
208 (MAS) is that (i) each agent has a partial capability to
209 solve a problem, (ii) there is no global system control,
210 (iii) data and knowledge for solving the problem are
211 decentralized, and (iv) the agent computation is asyn-
212 chronous [5]. In MAS, the agents need to work collec-
213 tively so that, as a group, their behavior solves the
214 overall problem without disruption, conflict, and
215 glitches. When a task is assigned, the agents often need
216 to find the other agents to collaborate with. Such a task
217 is easy if they know exactly which agents to contact and

218 at which location. However, a static distribution of
219 agents is very unlikely to exist for most real world appli-
220 cations. For dynamic multi-agent systems, agents need
221 to know how and where to find the other agents [6].
222 Proper structural topology thus plays a critical role in
223 these MAS systems. The topology determines how the
224 agents interact with each other, and how data and
225 knowledge are shared and communicated among the
226 agents. In [20], the authors studied three major MAS
227 topology models according to the criteria: (1) the ways
228 of activation, supervision, and communication between
229 the agents; (2) the dependency of the agents to complete
230 a task; and (3) the ways of sharing data, knowledge, and
231 other resources. These models are shown in Figs. 1–3.
232 For the purpose of ease of comparison, we give a brief
233 description of these models here.

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

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

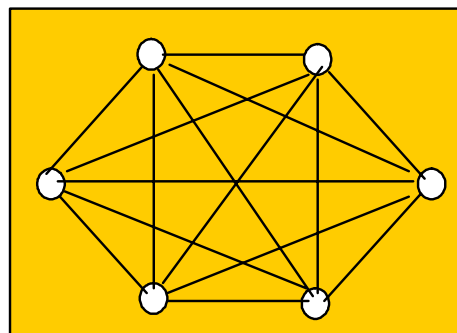


Fig. 1. Web-like topology of agent cooperation.

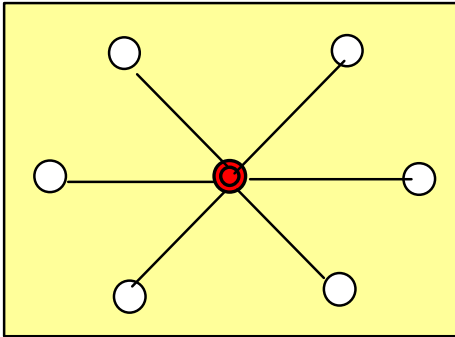


Fig. 2. Star-like topology of agent cooperation.

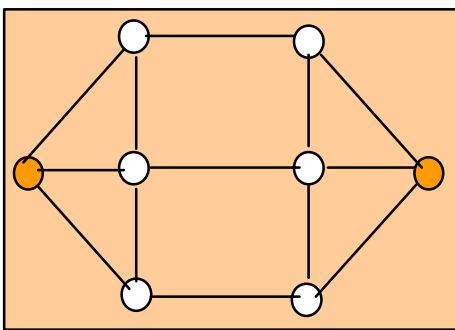


Fig. 3. Grid-like topology of agent cooperation.

259 (3) *Grid-like topology*: In a grid-like topology, each
 260 agent cooperates with a group of agents in its neighbor-
 261 hood (in terms of functional connections) that is a sub-
 262 set of agents in the assembly. That is, each agent has
 263 direct connections with a group of agents in its neigh-
 264 borhood (logically). Each group may be administered
 265 by a supervisor/facilitator. Interaction among agents
 266 not in neighborhood must pass through the neighboring
 267 agents in cascade. This is more like the concept of
 268 “system of systems.” Fig. 3 shows a diagrammatic illus-
 269 tration of this topology.

270 Each of the above topological models of MAS has
 271 advantages and disadvantages. Zhu et al. [20] gave a
 272 qualitative assessment of the above three models in
 273 terms of their capability of facilitating intensive knowl-
 274 edge embedding, accumulation, and incorporation.
 275 They found that the Web-like topology associated with
 276 its indiscriminate behavior of agent activation is often
 277 inefficient, and many times undesirable. In the star-like
 278 topology, though control and coordination limits the
 279 boundary of cooperation the agents can reach, it is
 280 desirable when efficiency of cooperation needs to be en-
 281 sured. The star-like topology is suitable for an environ-
 282 ment and applications where part of the MAS is to act
 283 as a central planner that involves team negotiation and
 284 awareness of what each agent knows, needs, and does.
 285 On the minus side, there is the potential for a facilitator

286 to become a communication bottleneck, or a critical
 287 point of failure.

288 In [20], a fourth topology, named Hierarchical
 289 Collective Agent Network (HCAN) model is also pre-
 290 sented. The HCAN, as shown by diagram in Fig. 4, pos-
 291 sesses the properties that (1) Agents are grouped in
 292 layers, (2) The layers are organized in hierarchy, (3)
 293 Agents in each layer are weakly connected, (4) Agents
 294 between layers are strongly connected, and (5) The con-
 295 trol and coordination of the agent at each layer are car-
 296 ried out through to the agents at the higher level.
 297 Whereas “weakly connected” means that interactions
 298 between the agents are mainly data communications
 299 only, no control function (call or instruct) takes place
 300 there, while “strongly connected” means that agents
 301 on the two ends of the link have both data exchange
 302 activities and control relations (e.g., client and server,
 303 mediator and mediatee, etc.).

304 The collective nature of the agents in the HCAN par-
 305 adigm overcomes some of the difficulties of the other
 306 agent system topologies. For example, it relieves the
 307 burden of intensive data-exchange between fellow agents
 308 in star-like topology by limiting agent communication to
 309 vertical layers of the assembly only. The collective na-
 310 ture of agent relation in the hierarchical architecture sim-
 311 plifies the functional design of the agent interactions and
 312 enhances the security and efficiency of the information
 313 processing, an advantage over the Web-like and Grid-
 314 like topologies. The HCAN architecture thus strikes a
 315 balance between the centralized control and distributed
 316 computation by allowing distributive agent operation
 317 within layers of the hierarchy and enforcing centralized
 318 control between the layers of the hierarchy, thus creat-
 319 ing a federated agents integration structure.

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

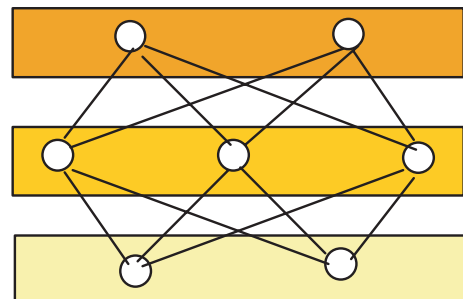


Fig. 4. HCAN topology of agent cooperation.

- 328 (1) *System-centered control*: In this control scheme, 384
 329 the system control agent at a high level knows 385
 330 and determines what actions/sub-actions each 386
 331 agent is to perform at certain time and place. 387
 332 The agent can employ a traditional control invoca- 388
 333 tion scheme. Obviously, this schema is not advan- 389
 334 tageous for maximal utilization of the agent 390
 335 functionality and the autonomous abilities of the 391
 336 agents. 392
- 337 (2) *Agent-centered control*: In this scheme, each agent 393
 338 knows its responsibility, and interactions with 394
 339 other agents when necessary. That is, the agents 395
 340 coordinate the interactions by themselves auto- 396
 341 nomously within its group or scope of cooperation. 397
 342 This is an object-oriented control approach. 398
- 343 (3) *Request-for-service-centered control*: In this 399
 344 scheme, either the central control agent or an indi- 400
 345 vidual agent can issue requests to all other agents 401
 346 specified in the problem domain in situations when 402
 347 cooperation is needed. The invoked agent per- 403
 348 forms the service requested, or issues requests to 404
 349 other agent to cooperatively accomplish the task.
 350 This is a hybrid approach of the above two.

351 In the HCAN architecture for sensor fusion and man-
 352 agement, the agents at the collective level only accept
 353 control from agents at the higher level. Hybrid system
 354 control law for this application in sensor management
 355 requires that directives to multiple platforms must be
 356 synchronized or chaos may occur. Mass effects require-
 357 ments would further exacerbate the problem since very
 358 tight synchronization is required in planning and execut-
 359 ing sensor allocation and re-allocation. While a central-
 360 ized, coordinated operation of the agent group is
 361 essential and needs to be strengthened, it is equally
 362 important to emphasize and retain a high level of agent
 363 autonomy. Thus, the HCAN in the sensor management
 364 application will function somewhat differently versus
 365 those equivalent components in totally centralized or to-
 366 tally autonomous agent control settings (e.g., the other
 367 three topologies discussed above). One distinction is
 368 the communication aspect. The HCAN will engage in
 369 either a one-to-one, direct-line connection schema or
 370 in the entirely open broadcasting approach, and switch
 371 according to the specific situation detected by the sensor
 372 monitoring agents. The HCAN will also switch between
 373 an action–prediction based control strategy and an ac-
 374 tion–response based strategy [17]. In the action–predic-
 375 tion based control strategy, the HCAN makes
 376 predictions of the possible future states of the system
 377 upon sensing the battlespace state changes (via the situ-
 378 ation assessment process) and applying pre-acquired
 379 knowledge in analyzing the collected information, and
 380 convey the predictions to involved agents along with
 381 the state reports. In the action–response based strategy,
 382 the system simply chooses a best reaction alternative

upon sensing the battlespace state changes and conveys 384
 the state report to involved agents. The action–response 385
 strategy would assume more agent autonomous respon- 386
 sibility while the action–prediction strategy provides 387
 more information to the agents, though the predictions 388
 may not be thorough and perfect. The system control 389
 agent of the HCAN decides on which control strategy 390
 to use according to the situation assessment and accord- 391
 ing to its goal of optimizing the overall sensor manage- 392
 ment functions. 393

In the following sections, we describe in more detail 394
 an application of the HCAN architecture for sensor fu- 395
 sion and management. In performing the sensor fusion 396
 and management tasks, the agents assembly will be in 397
 charge of determining registered sensors in field, cuing 398
 applicable sensors to obtain additional information 399
 about objects, take data from various sources and com- 400
 bine them into fused object information, acquire rele- 401
 vant target information, learn better observation and 402
 tracking strategies, and provide real-time decision sup- 403
 port for the sensor control and management operation. 404

3. HCAN for sensor management 405

It is noted that in a typical sensor fusion and control 406
 process, a number of functions need to be performed at 407
 different levels. Three levels of agent functions are iden- 408
 tified in our HCAN implementation of the process. The 409
 first is a sensor data acquisition level. It is at this level 410
 that connections to the various sensor resources are 411
 made. Agent modules are needed to automate the infor- 412
 mation retrieval and integration from heterogeneous 413
 sensor resources. The functions in these modules will 414
 also provide an effective means for extracting useful 415
 information from the sensor resources and perform fil- 416
 tering operations. At the second level, the reasoning 417
 module takes the filtered data from the data acquisition 418
 level, performs various correlation and association func- 419
 tions, and distills the data collections. The outcome of 420
 this level contains information useful toward target 421
 detection, situation awareness, learning and sensor con- 422
 trol, as well as representations of decision supporting 423
 knowledge. Finally, a control and adaptation level is 424
 at the top of the agent hierarchy. The user interface 425
 and visualization module of this level facilitates the task 426
 coordination and performance monitoring functions of 427
 the overall system. 428

The three level architecture of our HCAN system for 429
 sensor network management is illustrated in the block 430
 diagram of Fig. 5. In this architecture, as pointed out 431
 above: 432

- (1) Agents at the lower level interface directly to the 433
 sensor environment and monitor the sensor opera- 434
 tions. These agents collect sensor state parameters 435

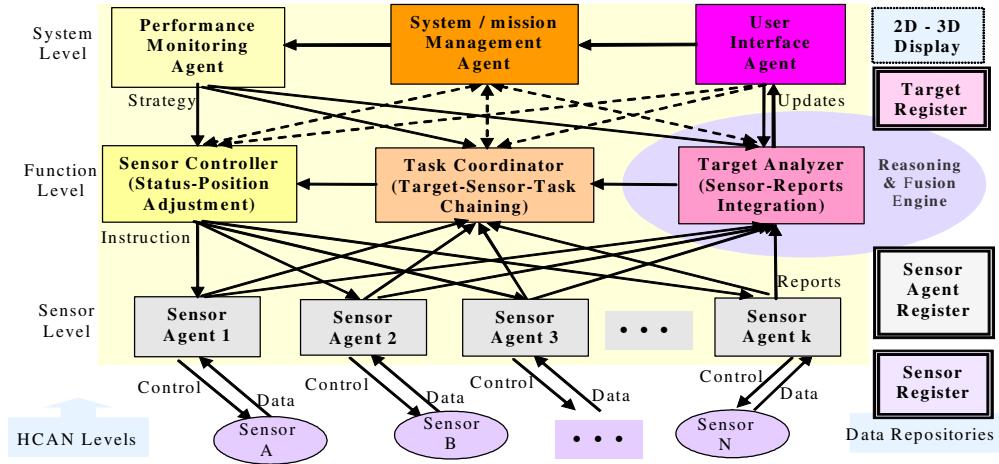


Fig. 5. Overall HCAN system blocks for distributed sensor network management.

436 and receive control feedback for sensor state
 437 adjustment. These agents act in a distributive
 438 fashion.
 439 (2) Agents at the function levels will apply analytic
 440 models and reasoning-integration techniques to
 441 make decisions for sensor state control and
 442 adjustment.
 443 (3) Agents at the system levels coordinate the sensor
 444 management activities of the agents at the lower
 445 levels. These agents interface with users as well
 446 as receive situation assessment inputs.

447
 448 The three-level HCAN architecture for multiple net-
 449 worked sensor fusion and management consists of seven
 450 different types of software agents and three main data
 451 depositories. The seven agent types are:

- 452 (1) Sensor Agents (SA), which are directly connected
 453 to the networked sensors for receiving target detection
 454 data from the sensors and sending sensor control
 455 commands. In this sense, the sensor agents
 456 also act as the sensor actuators. There are multiple
 457 sensor agents in the HCAN, one for each sensor.
- 458 (2) Target Analyzer (TA), which is essentially the Sensor
 459 Fusion Agent. All sensory data are fed to this
 460 agent and is processed for target validation and
 461 identification. It sends target data to the User
 462 Interface for display (and supporting the user)
 463 and send sensor assignment/adjustment requests
 464 to the Task Coordinator (cueing).
- 465 (3) Task Coordinator (TC), which determines what
 466 Sensor Control and management tasks need to
 467 be accomplished. It also finds and allocates proper
 468 sensors to specific target, or FOV (Field of View)
 469 for Sensor-target pairing/tracking coordination.
- 470 (4) Sensor Controller (SC), which receives directives
 471 and requests from both the System Management
 472 Agent (SMA) and the Task Coordinator (TC),

- 473 generates proper Sensor Control instructions,
 474 and sends the instruction to individual Sensor
 475 Agent for execution (Sensor status, parameter
 476 changes, and cueing).
- 477 (5) User Interface (UI), agent which is at the system
 478 management level for directly interacting with
 479 users. It is responsible for providing users a single
 480 picture of the situation awareness for the space
 481 covered by the NDS.
- 482 (6) System/Mission Management Agent (SMA),
 483 which keeps track of the overall mission objectives
 484 and ensures that the sensor management/control
 485 actions are consistent with the overall system/mis-
 486 sion management strategies and priorities.
- 487 (7) Performance Monitoring & Adaptation Agent
 488 (PMA), which oversees the system activity and
 489 performs parametric learning and system adapta-
 490 tion functions that will affect the performance of
 491 the agents at all level of the system.

492
 493 Some of these agent modules are to be described in
 494 more detail in this section. All agents in the HCAN
 495 architecture use a “publish-and-subscribe” model for
 496 data communication and agent interactions. There are
 497 three data repositories (registers) that are maintained
 498 and used by the agents in the HCAN architecture. They
 499 are: (1) Sensor Register (SR), (2) Sensor Agent Register
 500 (SAR), and (3) Target Register (TR). Each register is
 501 administered by an agent for performing data entry/re-
 502 trieval (responding to requests from other agents), con-
 503 tent updating, storage optimization, consistency
 504 checking, and database maintenance operations.

505 The Sensor Register (SR) contains a list of sensor de-
 506 vices, types, characteristics, deploy parameters (Posi-
 507 tion, Orientation, Scope, etc.), and their assigned
 508 Sensor Agents, as shown in Table 1.

509 In this table, the field “Sensor ID” gives a unique
 510 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)
-----------	------	-----------------	------------------------------	---------------------------------

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

528 The SR creates and keeps a record of mapping from
 529 sensor devices in field to Sensor Agent in HCAN. It per-
 530 forms consistency checks via a cross projection to the
 531 SAR, and keeps track of the sensors in current deploy-
 532 ment, and their assigned Sensor Agents. The SR needs
 533 to know and maintains updated information about the
 534 capability of each sensor deployed. The register is pub-
 535 lished by each individual sensor device (through System
 536 Management Agent) for registering a sensor in, and is
 537 subscribed by functional and system level control and
 538 coordinate agents. When answering queries about tar-
 539 gets, it needs to go through the associated Sensor Agent
 540 to find a list of targets that are currently being tracked
 541 by this sensor device.

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

546 Fields in the Sensor Agent Register (SAR) include the
 547 “Agent ID” which gives a unique identification of a
 548 software agent in the sensor management system, the
 549 “Associated Sensor (ID)” which indicates the physic
 550 sensor device that the agent is assigned to, and the “List
 551 of Target (ID) Under Tracking” which links the soft-
 552 ware agent to the target in track. Note that the “List

Table 2
SAR data entries

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

of Target (ID) Under Tracking” needs to be dynami- 553
 cally updated in the sensor management process as time 554
 passes, and there could be multiple targets in one sen- 555
 sor’s viewing/detecting range. 556

557 The SAR builds a mapping from the set of Sensor
 558 Agents to the set of sensor devices, and then to a set
 559 of Targets. It performs a consistency checking with a
 560 cross projection to both Sensor Register and Target
 561 Register, and keeps track of the Sensor Agents in cur-
 562 rent deployment (their assignment to sensors, and cur-
 563 rent targets in watching/tracking). The content of SAR
 564 is published by Sensor Controller for register the agent
 565 and sensor connections into the register, and is sub-
 566 scribed by Functional and System level control and
 567 coordinate agents. It needs to go through the Sensor
 568 Register entry to access the characteristics and deploy
 569 parameters of the associated sensors.

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

574 The fields “Target ID” in the above table gives a
 575 unique identification of a target being tracked. This “ID”
 576 is assigned by the software agent and will not conflict
 577 with other target IDs in the management space. The
 578 fields “Type” and “Characteristics” describe the physic
 579 nature of the target, while the field “Parameters (P, v)”
 580 records the current position and moving velocity of
 581 the target. The “list of Sensors (IDs) associated” links
 582 the target to specific sensors that are tracking this target
 583 or are in the tracking (detecting and viewing) range of
 584 the target. Note that this list needs to be dynamically up-
 585 dated in the sensor management process when the target
 586 enters into or leaves away from the sensor’s viewing/
 587 detecting range as time passes.

588 The TR records a mapping between a set of targets
 589 and a set of Sensor Agents. It performs a consistency
 590 check in a cross projection to Sensor Agent Register,
 591 and keeps track of the targets under observation. The
 592 TR is published by the Sensor Fusion and Target Ana-
 593 lyzer which is responsible for target discovery from sen-
 594 sor data integration. The TR content is subscribed by
 595 Functional and System level control and coordination
 596 agents. For sensor management function, the TR must
 597 know the identity of each target and a list of sensors that
 598 are currently tracking that target.

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

Table 3
TR Data entries

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

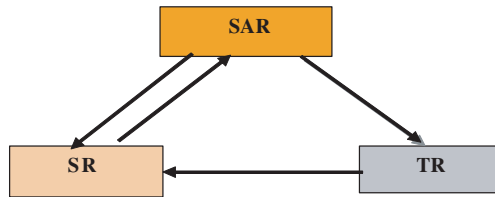


Fig. 6. Inter-relationship of the data repositories in the HCAN system.

602 sponding sensor agent). The rationale to have this redun-
 603 dancy is for both computational efficiency and fault-tol-
 604 erance considerations, of course, at the cost of memory
 605 space and maintenance of the fields. A relationship of
 606 the cross projection of the three registers can be illus-
 607 trated by a diagram in Fig. 6.

608 There are many sources of uncertainty at different
 609 levels of the sensor fusion and management computa-
 610 tions in the HCAN. For example, even if a situation-
 611 assessor is aware of the presence of certain objects in
 612 the operation space, such as the type of contact, inten-
 613 tion, reaction rationale, etc.; the exact dynamics of the
 614 object is still uncertain to the agents tracking the target.
 615 The knowledge about the object dynamics is critical in
 616 constructing an optimal strategy of sensor management
 617 action. Various statistical methodologies and knowledge
 618 discovery techniques may be applied in the reasoning
 619 module of the HCAN agents. The level of uncertainty
 620 forces the reasoning agents to operate with different
 621 decision strategies. Some of these agent functionalities
 622 are described below.

623 The Sensor Agents (SA) plays an important role in
 624 interfacing between the sensor network and the manage-
 625 ment system. Data from diverse sensor resources are fil-
 626 tered and preprocessed by the SA to a form that can be
 627 effectively used by the sensor control agents. The pre-
 628 processing and filtering operators are in charge of clear-
 629 ing up the noises and compensating for the uncertainties
 630 contained in the raw data. The interface is standardized
 631 such that its application can be ported to all classes of
 632 sensors with minimal installation and special interface
 633 rendering. A sensor agent can be assigned/allocated to

634 different sensors (i.e., a sensor agent is NOT necessary
 635 to be tied to a specific sensor device all its life; it can
 636 be dynamically switched to tie with (be assigned to) dif-
 637 ferent sensor devices. Of course, only one sensor device
 638 should be tied to one sensor agent at any time. The sub-
 639 scribe-and-publish functions of the SA are defined in
 640 Fig. 7.

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

661 The Task Coordinator (TC) agent applies certain
 662 control strategies to guide the Sensor Control agent in
 663 sensor allocation and deployment planning/re-planning
 664 process. A set of goals and sub-goals are set up by the
 665 TC agent according to the sensor space situation, mis-
 666 sion requirements, sensor operation parameters and
 667 function specifications, operator instructions, etc. From
 668 these data, the TC will analyze the situation and recom-
 669 mend the optimal course of action to subordinate level
 670 agents. From the analysis, the TC agent determines
 671 what Sensor Control and management tasks need to
 672 be done. It tries to find and allocate proper sensors to
 673 specific targets, or FOV for Sensor-target pairing/track-
 674 ing 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.
 - Check whether multiple reports reporting the same target (or targets are reported in multiple sensors) – association.
 - 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:
 - 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.
 - If an existing target, update target entry in TR according to the new reports.
 - Publish to User Interface Agent:
 - Report status and parameters of any new target entry for 2D/3D display.
 - Update status and parameters of targets currently under tracking.
 - Publish to Task Coordinator:
 - Report instance of target uncertainty/ambiguity.
 - 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
 - Adjustment of Sensor parameter (e.g., re-orienting) to keep track of a moving target within a Sensor's rotation range;
 - Switch of Sensors to handoff target when a target is moving out of the reachable and manageable range of a Sensor;
 - 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.

675 tion parameter requirements for particular sensor to ob- 696
 676 serve a specific target. The tasks determined by the TC 697
 677 are to be executed by the Sensor Controller. The actual 698
 678 (physical) execution of the sensor control is accom- 699
 679 plished through the Sensor Agent, and further pass over 700
 680 to the Sensor device. The subscribe-and-publish func- 701
 681 tions of the TC are defined in Fig. 9. 702

682 The Sensor Controller (SC) agent receives directives 703
 683 and requests from both the System Management Agent 704
 684 (SMA) and the Task Coordinator (TC). It generates 705
 685 proper Sensor Control instructions, and sends the 706
 686 instruction to the individual Sensor Agent for execution 707
 687 (Sensor status and parameter changes). The functions 708
 688 performed by the Sensor Controller include assigning, 709
 689 distributing, and dispatching Sensor Agents to individ- 710
 690 ual Sensors in service. The SC finds sensors that fit to 711
 691 specific function and position requirement, issues status 712
 692 and position parameters of the sensors and parameter 713
 693 changes to designated sensors. It will also be in charge 714
 694 of resolving Sensor-target tracking conflict in the hand- 715
 695 off process, and optimizing sensor distribution and task 716

assignment. The subscribe-and-publish functions of the 696
 SC are defined in Fig. 10. 697

708 The other agents of the HCAN function in the fol- 698
 709 lowing ways. The User Interface (UI) agent connects 699
 710 sensor operators to the HCAN, and subsequently to 700
 711 the sensors. The agent will assist the reasoning and infer- 701
 712 ence agent and the learning adaptation agent by receiv- 702
 713 ing instructions and/or refutations about their sensor 703
 714 control decisions, and adjust (override) the sensor con- 704
 715 trol parameters by applying certain control strategies 705
 716 that are aimed to improve the system performance. 706

707 The System Management Agent (SMA) is responsible 707
 708 for the synchronization of the sensor management oper- 708
 709 ations among the agents in the HCAN. It constantly 709
 710 evaluates the available information about the states of 710
 711 the sensors, the locations, environment, and time sched- 711
 712 ules, and computes the probabilities on each of the 712
 713 objectives. When necessary information is provided by 713
 714 users, the SMA sets up a sensor management policy 714
 715 and a sensor control strategy (e.g., best-first, greedy, 715
 716 heuristic, etc.). It will then prioritize the sensor control 716

- 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.

717 tasks according to these priorities and control strate-
 718 gies—with respect to targets status and other system
 719 parameters and set up internal relations and composi-
 720 tions of sensors in the environment.

721 The Performance Monitoring & Adaptation Agent
 722 (PMA) is responsible for environmental analysis, and
 723 providing improvements to the control models and
 724 strategies used by the lower level agents (e.g., SC, TC
 725 and TA) for sensor management. In its role as a system
 726 performance and effectiveness monitor, the PMA is
 727 equipped with situation assessment and adaptation
 728 functions for system optimization. It also contains func-
 729 tions for supporting sensor reconfiguration in the event
 730 of partial/total loss of a sensor in an autonomous oper-
 731 ating situation.

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

747 Among the agent modules in the HCAN structure for
 748 sensor management, the Task Coordinator agent and
 749 the Sensor Controller agent play the major role for sen-
 750 sor allocation planning/re-planning and optimization of
 751 the dynamical sensor deployment and adjustment. A
 752 performance monitoring capability and a feedback/opti-
 753 mization mechanism are implemented in the joint pro-
 754 cesses of these agents for process refinement. A control
 755 flow diagram of the process is shown in Fig. 11.

756 Most autonomous control systems are knowledge-
 757 intensive information processing ensembles. The same

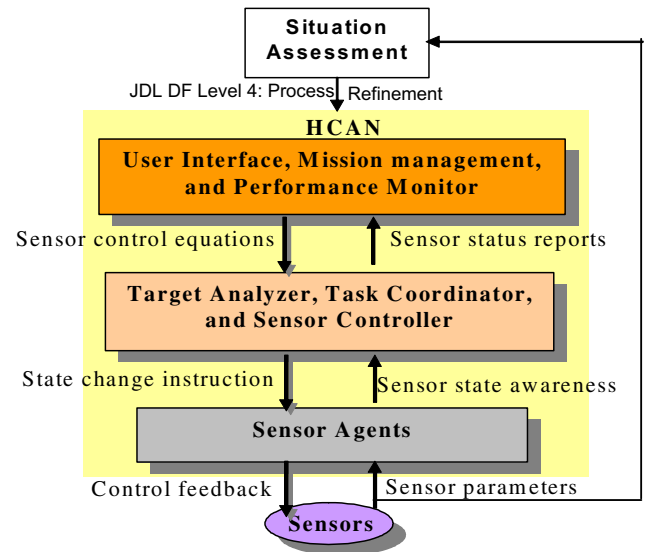


Fig. 11. Feedback control flow of sensor management process.

property is held by the HCAN. The stability and robust-
 ness 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. Correspond-
 ingly, factors that affect the control stability and robust-
 ness 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, communi-
 cation 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 high-
 er level of the HCAN. The communication can be car-
 ried in the ways of the following:

- (1) *Private line communication*: This resembles the tra-
 ditional way of parameter passing. Only the issu-
 ing and receiving agents know the
 communication has taken place. The advantage

758
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 770
 771
 772
 773
 774
 775
 776

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

783 (2) *Blackboard communication*: This is also called
784 party line communication. In this method, every
785 agent has access to a common communication
786 channel. Any task requests are posted to this chan-
787 nel and every agent responds to the call auton-
788 omously. If a request meets the pre-assigned duties
789 or pre-specified parameters of an agent, that agent
790 activates. The advantage of this approach is that it
791 maximally guarantees the possibility of accom-
792 plishing the required task. The disadvantage is that
793 it sometimes may still interfere other agents' activ-
794 ities, and waste system resources because the
795 agents needs to periodically check and process
796 the requests even they are not present.

797 (3) *Reserved-channel communication*: This is also
798 called the mailbox method. In this method, a
799 group of agents have an established agreement or
800 protocol that specifies the locations (or frequen-
801 cies) where communication signals will be trans-
802 mitted to and accessed by the members of the
803 group. This method is a compromise of the above
804 two methods. Only agents within the group know
805 the special places (or frequencies) where the infor-
806 mation is posted. The advantage of this method is
807 that it allows a proper allocation and reservation
808 of system resources. The disadvantage is that it is
809 difficult to identify the coherent group of agents
810 that needs to share and exchange information
811 within themselves exclusively.

812
813 In Section 4, we will present an implementation of
814 these methods and approaches in a simulated environ-
815 ment for sensor fusion and management in a GMD
816 application of the HCAN and its agent modules.

817 4. Experimentation

818 The ability to integrate and correlate a vast amount
819 of disparate information from heterogeneous sensor
820 and data resources with varying degrees of certainty in
821 real-time is an impediment issue for mission-critical mil-
822 itary decision support systems (DSS). For example, mil-
823 itary commanders use multiple sensor/data resources
824 and intelligence from reconnaissance and surveillance
825 assets both in and out of a theater to build a whole pic-
826 ture of the battlespace in crucial military operations [8].
827 The commanders need to know and understand the rela-
828 tionships among the data, such as, what are the physical
829 and functional constituency relations among the objects

in a given geographic sector? Are there sequential or 830
temporal dependencies of the objects and what will trig- 831
ger them? What are the possible consequences of the ac- 832
tion and re-actions? That is, decision making based on 833
the situation assessment and impact assessment (SA/ 834
IA). These assessments are particularly important for 835
identifying and prioritizing "gaps" between the opera- 836
tion planning and the real-time interactions. 837

In a mission-critical theater/situation demanding 838
decision support, timely and accurate data fusion is a 839
force multiplier. The lower-level data fusion from single 840
or multiple sensor resources has become relatively well 841
understood, resulting in accurate positional tracks and 842
identification of physical objects. However, the pro- 843
cesses for higher levels of data fusion, namely the level 844
2—situation assessment, and level 3—impact assessment 845
(SA/IA), still requires the study and development of 846
mathematically rigorous techniques and computational 847
schemes. More in this realm is the level 4—process 848
refinement which involves active control and manage- 849
ment of the sensor resources. The kind of robust, inte- 850
grated fusion architectures for handling increasing 851
diversity of input sources are especially important in 852
contemporary decision support missions. A well crafted 853
software agent system integrating knowledge acquisition 854
tools and proper decision support models can assist mil- 855
itary operation planners in their tactical decision-mak- 856
ing situations in many different ways, particularly with 857
respect to quickly identifying responses and counter-re- 858
sponses to enemy action or inaction, providing a more 859
current and more comprehensive picture to the field 860
units. 861

We apply the above HCAN model to sensor manage- 862
ment on a simulated GMD platform (interceptor, space- 863
based, or airborne) to demonstrate the capability in sen- 864
sor management and adaptive data processing. To 865
accomplish the mission and schedules of all sensors 866
and platform resources relative to its current mission 867
and prime goal, we first conducted a system model anal- 868
ysis. The intent of this analysis is to hide the system 869
dependent details and to abstract sensor information 870
so as to form a basis for a formal specification of the 871
sensor platform capabilities and their configurations. 872
Care was taken to characterize the types of information 873
provided by disparate systems in such a way as to make 874
them compatible without making them sterile. This 875
characterization is structured such that it's possible to 876
determine complementary sensor characteristics and to 877
allow the system to determine a sensor that can provide 878
additional data leading to more accurate information, as 879
opposed to duplicate data. The form of the characteriza- 880
tion lends itself to rapid traversal to assist in the cueing 881
process. For example, a tree structure or directed acyclic 882
graph (DAG) based on sensor spectrum is more desir- 883
able than a straight list due to their speed in traversal. 884

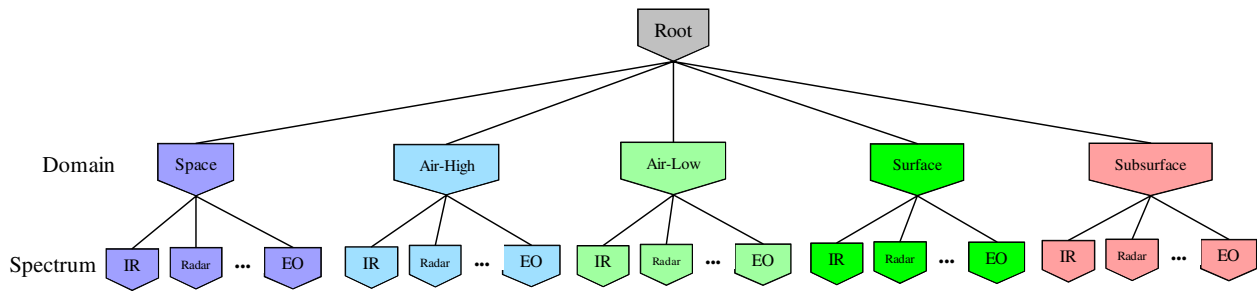


Fig. 12. Sensor deployment and classification diagram.

885 For the purposes of system specification, we chose to
 886 limit the sensor capabilities characterization to two lev-
 887 els of abstraction. Fig. 12 depicts a sample characteriza-
 888 tion. First, we divided a sensor operating environment
 889 into five realms: space, air-high, air-low, surface, and
 890 subsurface. The subsurface realm consists of the subter-
 891 ranean and underwater areas. A sensor is associated
 892 with a realm based on its sensing capability. For exam-
 893 ple, while a DSP satellite exists in the space realm, its
 894 sensing capability is targeted at the air realm. Many sen-
 895 sors will be associated with more than one realm (e.g.,
 896 THAAD sensor).

897 The sensor monitoring agents in HCAN need to
 898 promptly sense and detect state changes of the sensor
 899 space, including the altering of tactical mission objec-
 900 tives, the switching of targets, the loss or gain of tactical
 901 forces and other assets in both adversary and own units,
 902 the relocating of the battlespace, etc. The main duty of
 903 the HCAN agents thus is to timely collect and promptly
 904 feedback the spatial situation and field sensor informa-
 905 tion to functional agents involved. In addition, the
 906 agents are also responsible for on-site analyses of the
 907 collected data and extraction of information that is use-
 908 ful for the control agent to coordinate the actions of the
 909 distributed agents or agent groups in the HCAN.

910 In addition to sensor control capabilities, there are
 911 also constraints associated with sensor detecting capa-
 912 bilities. Two of the most obvious are the line-of-sight
 913 (LOS) and range constraints associated with many sen-
 914 sors. But, there are more subtle constraints that must
 915 also be taken into consideration in the sensor manage-
 916 ment control mechanisms. Sensor platforms themselves
 917 may have resource management constraints (power, atti-
 918 tude, interference, orbit, time-on-station, etc.) associ-
 919 ated with the platform itself. These constraints are also
 920 entered into the management schema. In a similar vein,
 921 constraints that occur as a result of a single platform
 922 having multiple sensors must also be considered (inter-
 923 ference, resource limitations, etc.).

924 The result of this analysis is a specification for sensor
 925 configuration that incorporates capabilities and con-
 926 straints of the sensor and its platform. The specification
 927 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 follow-
 ing networked sensor parameters:

- Number of sensors: 10 – 15 sensors (with 15 as the maximum). 942
- Sensor Platforms: all domains possible—Airborne, satellite, surface, and subsurface. 943
- Platform characteristics: mobile and fixed—support multiple sensors with issues related to range, attitude, placement, etc. 944
- 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). 945
- Sensor characteristics: Detection range specified by LOS (Line of Sight) and FOV (Field of View). 946
- Sensor deployment parameters: Location (3D coordinates, Ground, Mid-Air, Air, Upper-Air), velocity, and terrain. 947

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. 948

- Number of targets: 10 max at a time. 949
- Target types: Missile, Aircraft, Land Vehicles, etc. 950

The capability can be significantly improved with a proportional increase in the quantity and complexity 951

969 of the target parameters. In the simulation environment,
970 we set up the situation in the following computational
971 steps:

- 972 1. Defining an operation space (an AOI, that is the total
973 area of interest—space where the sensors are to oper-
974 ate jointly, the same space the targets are to travel
975 through—not the AOI of each sensor), which is a
976 3D box including space, air, land, and sea areas.
- 977 2. Designing targets moving across the operation space,
978 in sequence, individually or in groups. Multiple tar-
979 gets occur, where each is controlled by a dynamic
980 equation with its own parameters (position, velocity,
981 trajectory, etc.) entering the monitoring space
982 independently.
- 983 3. Visualizations of the operation space, sensor loca-
984 tions, target movements, sensor cueing, handoff,
985 etc., are to be handled through the GUI development
986 of the system. This piece was mostly derived by lever-
987 aging our previous work, the Sensor AEDGE
988 application.
- 989 4. The HCAN mechanism performs the following
990 actions upon the simulated inputs from the multiple
991 networked sensors.
 - (1) When system operation starts, a number of Sen-
sors (A, B, C, ...) and corresponding Sensor
Agents and Platform Agents are deployed in
place, registered in the Sensor Register and Sen-
sor 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 mon-
itors 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 connec-
tion with the Sensor A.
 - (6) The Sensor Agent sends an event about the spe-
cific target (target type, location, motion char-
acteristics, 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 uncer-

tainty and ambiguity arises, send an event
(request) to the Task Coordinator for sensor
cueing, 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 appropri-
ate 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).

1045

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 appro-
priate sensor to provide additional data leading to more
accurate information, as opposed to duplicate data. This
form of the characterization lends itself to rapid tra-
versal to assist in the cueing process. Fig. 13 depicts
some screen captures of the implementation. The situa-
tion involves an AOI with surface and airborne ISR as-
sets. The surface assets are an AEGIS cruiser (radar)
and two Rapiers sites (optical camera). The airborne as-
sets 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 situa-
tion, the goal was to have ISR assets from different spec-
trums in order to validate the *HCAN Sensor Manager's*
ability to assign complementary sensor assets for contin-
ual tracking of targets.

1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068

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 accommo-
dating heterogeneous forms of information;
- (3) a diverse interface for various input resources, out-
put formats, and human interactions;
- (4) an ability of reasoning on incomplete and inconsis-
tent information, and extracting useful knowledge
from the data of heterogeneous resources;

1069
1070
1071
1072
1073
1074
1075
1076
1077

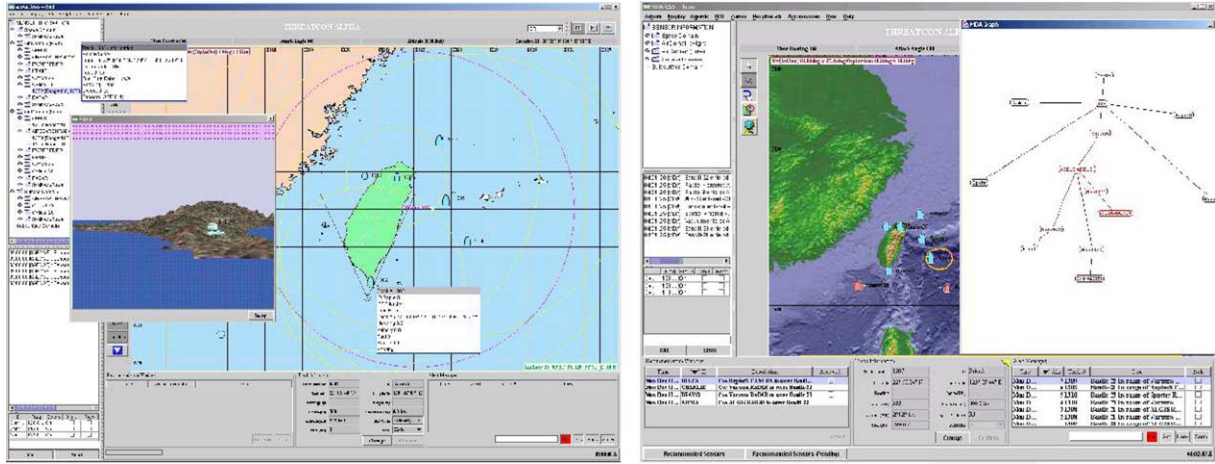


Fig. 13. Screen captures of the HCAN implementation of sensor management.

- 1078 (5) an ability of incorporating real-time dynamics of
 1079 the information resources into the system anytime
 1080 during the operation, and promptly adjusting the
 1081 reasoning mechanisms;
 1082 (6) an ability of summarizing and refining knowledge
 1083 extracted, and distinguishing mission and time
 1084 critical knowledge from insignificant and redund-
 1085 ant ones;
 1086 (7) a capability of supplying meaningful and accurate
 1087 explanations, both qualitatively and quantita-
 1088 tively, of the automated system actions; and
 1089 (8) a capability of providing adequate control and
 1090 scrutinizing of the system operations under the
 1091 environmental constrains of the given situation.

1092
 1093 The expected performance improvements from
 1094 employing the HCAN architecture for sensor manage-
 1095 ment include the following:

- 1096 • *Efficiency*: The system makes maximum use of
 1097 onboard platform control and decision-making capa-
 1098 bilities of the HCAN. The resulting software mini-
 1099 mizes human intervention and enhances the self-
 1100 sustainability of the multi-sensors autonomous
 1101 operations.
- 1102 • *Robustness*: The system is equipped with a self-diag-
 1103 nosis and certain self-repair, reconfiguration, and
 1104 alternatives/backup capabilities through the embed-
 1105 ded PMA modules and functionalities. The resulting
 1106 software allows the multi-sensors' sustained and reli-
 1107 able operations even under partial impairment of the
 1108 system.
- 1109 • *Flexibility*: The system is empowered with high level
 1110 of scalability and field adaptation ability. The
 1111 HCAN-based control system re-organizes itself in
 1112 different levels involving different numbers of compo-
 1113 nents. 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 prob-
 lems of how to make effective use of real-time informa-
 tion acquired from multiple and heterogeneous sensor
 and data resources, and reasoning on the gathered infor-
 mation for situation assessment and impact assessment
 through a hierarchical collective agents assembly orga-
 nized 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 be-
 tween 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 con-
 trol scheme is employed to coordinate a top-down and
 bottom-up two-way iterative process for the agent-acti-
 vation and interactions, rather than the conventionally

1114
 1115
 1116
 1117

1118

1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133
 1134
 1135

1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148

1149 adopted one way control approach [19]. The collective
 1150 nature of the HCAN architecture allows for flexible
 1151 addition or modification of the agents in the system be-
 1152 cause no complex de-coupling operations from the other
 1153 agents at the same level (neighboring agents) are needed
 1154 for the agents added or deleted. More importantly, the
 1155 HCAN renders itself to a fault tolerant computing
 1156 architecture, which is especially critical to sensor man-
 1157 agement operations. Since no tight coupling or coordi-
 1158 nation takes place among the agents at the collective
 1159 agent level, every agent acts by their own under the
 1160 supervision of the control agents at an upper level of
 1161 the hierarchy. Thus, the agents at the collective level
 1162 can be assigned to perform either different tasks or the
 1163 same task at the same time, allowing for fault detection
 1164 and functional back up.

1165 The HCAN is flexible in terms of the ability in which
 1166 communities of agents can be assembled, and the adap-
 1167 tation with which services can be added at runtime and
 1168 brought into use without requiring changes to the other
 1169 parts of the agent assembly. A unified set of concepts,
 1170 declarations, and interfaces that are consistently config-
 1171 ured across all services in the framework and the role
 1172 played by the agents at different levels are defined. The
 1173 HCAN architecture strikes a balance between the cen-
 1174 tralized control and distributed computation by allow-
 1175 ing distributive agent operations within layers of the
 1176 hierarchy and enforcing centralized control between
 1177 the layers of the hierarchy, thus eases the coordination
 1178 and control burden needed to manage interactions be-
 1179 tween agents. The worth of this concept lies in its appli-
 1180 cability to many operational situations. From a single
 1181 integrated air picture (SIAP) to an integrated intelli-
 1182 gence preparation of the battlefield (IPB) application,
 1183 the HCAN Sensor Manager concept can be applied
 1184 without reengineering the core architecture. The intelli-
 1185 gent agents that provide the decision support assistance
 1186 can be tailored to the situational awareness and decision
 1187 needs of the designated users. Additionally, users with
 1188 different needs can have different decision support cli-
 1189 ents while using the same core data and architecture.
 1190 We don't force a common picture; we provide a tailored
 1191 picture based on a common situation.

1192 References

1193 [1] J. Bradshaw (Ed.), *Software Agents*, AAAI Press, 1997.
 1194 [2] M. desJarins, Representing and reasoning with probabilistic
 1195 knowledge: a Bayesian approach, *Uncertainty in Artificial Intel-*
 1196 *ligence* (1993) 227–234.

[3] K. Decker, K. Sycara, Intelligent adaptive information agents, *Journal of Intelligent Information Systems* 9 (1997) 239–260. 1197
 [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. 1198
 [5] R. Flores-Mendez, Towards a standardization of multi-agent system frameworks. Available from: http://www.acm.org/crossroads/crew/roberto_flores-mendez.html. 1199
 [6] Giampapa, Paoluc, Sycara, Agent interoperation across multi agent system boundaries, in: *Proceedings of 4th International Conference on Autonomous Agents*, Barcelona, Sp, June 2000. 1200
 [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. 1201
 [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. 1202
 [9] C. Knoblock, J. Ambite, Agents for information gathering, in: J.M. Bradshaw (Ed.), *Chap. in Software Agents*, 1997, pp. 347–373. 1203
 [10] M. Lejter, T.A. Dean, Framework for the development of multiagent architectures, *IEEE Expert* (1996) 47–59. 1204
 [11] G. McIntyre, A comprehensive approach to sensor management and scheduling, *Dissertation*, George Mason University, Fall 1998. 1205
 [12] P. Maes, Agents that reduce work and information overload, *Communications of the ACM* 37 (7) (1994) 30–40. 1206
 [13] S. Musick, R. Malhotra, Chasing the elusive sensor manager, *Proceedings of the IEEE 1994 NAECON*, 606–613, 23–27 May 1994. 1207
 [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. 1208
 [15] B. Rodriguez, W. Pohlman, A system for distributed inferencing, in: *Proceedings 1996 Rochester Forth Conference Open Systems*, 1997. 1209
 [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. 1210
 [17] K. Sycara, D. Zeng, Multi-agent integration of information gathering and decision support, *Proceedings of the 12th European Conference on Artificial Intelligence I* (1996). 1211
 [18] E. Waltz, J. Llinas, *Multisensor Data Fusion*, Artech House Inc., 1990. 1212
 [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. 1213
 [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. 1214
 [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. 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256