Intelligent Track Analysis on Navy Platforms Using Soft Computing

Robert Richards, Richard Stottler, Ben Ball, and Coskun Tasoluk

Stottler Henke Associates, Inc.,
951 Mariner's Island Blvd, Suite 360
San Mateo, CA 94404
{Richards, Stottler, Bball, Tasoluk}@stottlerhenke.com
www.stottlerhenke.com

Abstract. We have developed and continue to enhance automated intelligent software that performs the tasks and decision making which now occurs by the personnel manning watch stations in the Combat Direction Center (CDC) and Task Force Combat Center (TFCC), on-board aircraft carriers and other Navy ships. Integrating information from various sources in a combat station is a complex task; disparate sources of information from radars, sonars, and other sensors are obtained by watch station surveillance guards, who must interpret it and relay it up the chain of command. The *Intelligent Identification Software Module* (IISM) alleviates some of the burden placed on battle commanders by automating tasks like management of historical data, disambiguating multiple track targets, assessing threat levels of targets, and rejecting improbable data. We have knowledge engineered current CDC/TFCC experts and designed IISM using C++ and SimBionic, a visual AI development tool. IISM uses multiple soft computing techniques including Baysian inference and fuzzy reasoning. IISM is interfaced to the Advanced Battle Station (ABS) for use on many US Navy sea vessels.

1. Introduction

The Combat Direction Center (CDC) and Task Force Combat Center (TFCC) on-board aircraft carriers and other ships must be manned with dozens of highly trained technical and tactical personnel [1]. The reason for this is the complexity of the weapon systems and associated information, as shown by the high-level organization of it in Figure 1. The combat areas consist of people; computers; and displays; and the arrows (in the figure) roughly correspond to information flow between combat areas and from sensors, to combat areas and from combat areas to weapons/countermeasures. CDC/TFCC operation is complicated by a large number of sensors, weapons and countermeasures. These operations will only become more complicated as additional sensors, weapons, and even war-fighting areas are added. Furthermore, through the Cooperative Engagement Capability (CEC), each ship can use the sensors and weapons on other ships thus adding additional combat areas, sensors, and weapons.

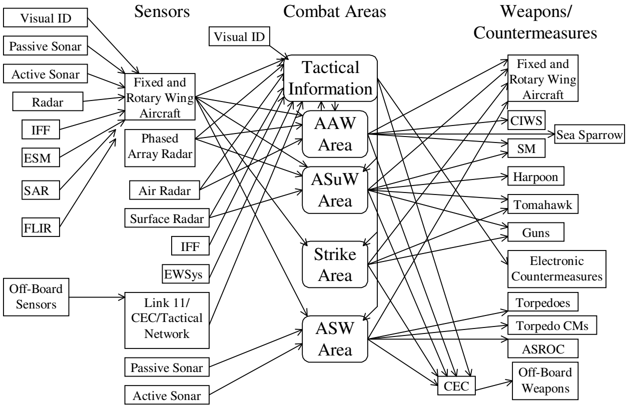

Fig. 1. Weapon System High Level Overview

A naval commander must make complex decisions based on limited or noisy information. In partially observable and adversarial environments it is vital to keep track of an approximate model of the world that simultaneously maintains multiple hypotheses about the world state [2]. These hypotheses facilitate reasonable decisions to take in response to the hostile environment.

To ameliorate the complexity of these systems, Stottler Henke has developed the *Intelligent Identification Software Module* (IISM) that performs the tasks and decision making which now occurs by the human manning that watch station, such as tracking objects that merge and later split up, maintaining history of possible tracks for an object, assessing threat level, rejecting “insane” data, and handling errors.

IISM is interfaced to the Advanced Battle Station (ABS) for use on many US Navy ships. Given tracking data and time stamps from the Advanced Battle Station (ABS), IISM updates the history list of tracking and identification data, rejects nonsense tracks, compares recent history to past patterns of activity, alerts the commander when necessary, and provides customizable identifications of targets as well as the threat level of each target. IISM is also capable of correcting errors and recovering snap-shot and history data after unforeseen catastrophes.

We have knowledge engineered current CDC/TFCC experts and determined that the cognitive processes being utilized were reproducible with Artificial Intelligence techniques [3]. We determined the types of tasks performed and the knowledge required for those tasks. A breadth of positions was important to keep the representation schema truly general. We designed the general CDC/TFCC knowledge representation schema and then an intelligent CDC/TFCC equipment control, monitoring, processing, and fusion system. From knowledge engineering and the schema, we designed and implemented IISM using C++ and SimBionic, a visual AI development tool that can help in the development of fuzzy, Bayesian and other AI techniques.

2. IISM Input / Output Description and Functional Overview

Human tactical decision making in warfare scenarios can be described with the simplified diagram shown in Figure 2. Imperfect information about the current state of the world is gathered by a diverse set of sensors. These sensors can be in several modes, may be off ship, and may be human in nature. The human decision-maker receives the sensor data through a communication or perception processes. Based on that information he makes decisions to take actions that affect the objects in the world over which he has direct control. These might include CDC/TFCC display systems, airborne platforms, weapon systems, communications, and sensors.

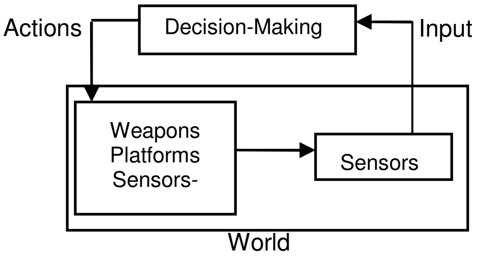

Fig. 2. Human Tactical Decision-Making

On a highly conceptual level, IISM's task can be viewed as a classification problem of the threat level assigned to individual entities, e.g. ships, present in the scenario. Maintaining a consistent and reasonably approximate model of several entities' attributes that are only partially perceivable implies the task of track handling and analysis. The latter is exploited in IISM to: 1) determine the identity of an entity (or some degree of certainty about it), 2) perform path analysis of entities and (3) infer abstract conclusions regarding the behavior of entities on the basis of their movement over time. Stated another way, both positive and negative evidence is tracked to form multiple, possibly competing hypotheses. Conclusions about these hypotheses are inferred for tracks through the process of elimination reasoning.

IISM stores and reasons about incoming track data in a flexible and customizable manner as defined by the control logic defined in SimBionic (see below). During this processing, IISM checks the quality of incoming messages, it updates its history of vessel movements (*tracks*) and IDs and performs threat assessment of units. This functionality is presently performed by trained watch-standing personnel aboard ships. It requires reasoning about whether the perceptions align with the internal model of the world and how *insane* (i.e. misaligned) perceptions are treated.

Insane and noisy data handling

Insane data can arise through an incorrect model or faulty perceptions, and special care must be taken in order to extract hints to potential threats instead of discarding them just like incorrect perceptions are discarded. The IISM reasoning functionality is performed in three subsequent steps in IISM's *Insanity Checker*: (1) *Threat processing* marks a unit as a potential threat in case insane perceptions are indicating this. (2) *Data Neglect Checking* takes account of an erroneous internal model caused by sensor noise and updates the model with the insane update. (3) *Inconsistency with ID checking* keeps track of harmless, but questionable/suspicious pieces of information and thus allows reasoning about temporally dispersed perceptions.

Track Hypothesis Handling

Instead of keeping a flat organization of unit ID hypotheses, IISM uses a hierarchical approach to refine an ID hypothesis as needed, such as in the case of determining the exact type of the enemy's unit. IISM assigns each hypothesis a particular *certainty level* that describes its reliability. When we get new data we use a *Bayesian network* update to keep track of the proper certainties for each track hypothesis. When the certainty for one of multiple hypothesis of a track is changed, or when a new hypothesis for a track arrives, an update algorithm is called on that track. This update algorithm uses the hyperbolic arctangent adjustment algorithm on each certainty to propagate the change made by the additional information. This algorithm runs through every hypothesis that is related to the changed one, updating each certainty according to Bayesian rules. These rules update the certainties based on the prior values and how closely they are related to the other related certainties.

Example Situation

Figure 3, shows an area around the Persian Gulf and provides an idea of how cluttered the environment being monitored and assessed can be.

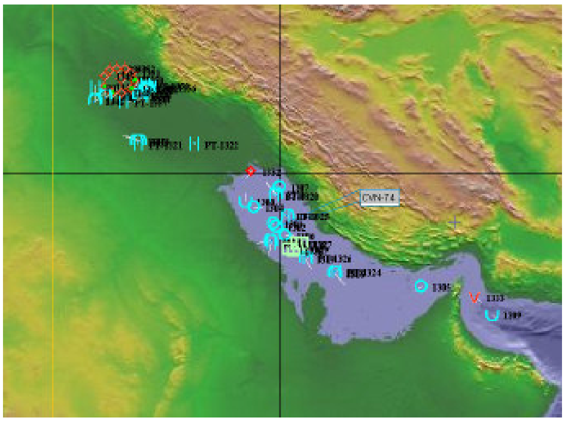

Fig. 3. Example of Density of Contacts that Need Monitoring and Assessing

Let's examine a situation where two surface tracks (track 1 & 2) are first detected, both traveling at a high speed (50 knots). At this point, IISM would already inference a subset of platform types based on their speed. Later these two tracks split up. Track 2 later merges with track 3, which had previously been IDed (identified) as an Iranian Houdong Fast Patrol Boat. These tracks (2 & 3) soon split up; at this point IISM does NOT know which of the tracks (2 or 3) is the formerly identified Iranian Houdong Fast Patrol Boat. Therefore, IISM will keep both sets of past information and use new information to improve its hypothesis on what each boat is. As can been seen even with this simple scenario, the situation is very fluid and multiple hypotheses must be tracked and re-evaluated as new information is obtained.

3. SimBionic

SimBionic is a visual framework that simplifies the authoring of simulated behaviors or algorithms. SimBionic's framework consists of a canvas depicting algorithms as a finite state machine (FSM) graph, a palette of geometric objects and glyphs, and a dictionary of actions and predicates.

SimBionic employs four programming constructs; 1) actions, which define all the different actions the algorithms can perform; 2) algorithms (also referred to as behaviors) that string together actions and conditional logic; 3) predicates, which set the conditions under which each action and algorithm will happen; and 4) connectors, which control the order in which conditions are evaluated, and actions and algorithms take place.

These four constructs allow one to create algorithms that range from simple sequences to complex conditional logic. Via SimBionic's authoring canvas, see Figure 4 (left image), users can visually create algorithms by drawing actions and invoke algorithms (represented as rectangles) and conditions (represented as ovals) to interact in both simple and complex combinations via connectors (represented as arrow-shaped lines with priority numbers). This canvas also allows users to assign arbitrary expressions and comments to these elements.

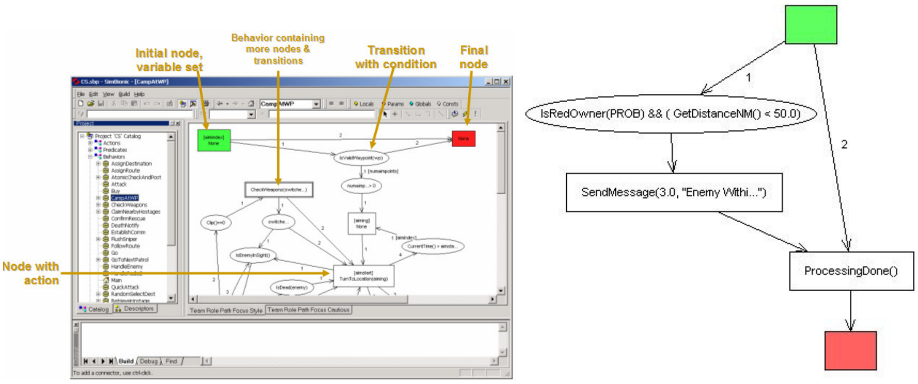
Initial node, variable set
Behavior containing more nodes & transitions
Transition with condition
Final node
Node with action
1
IsRedOwner(PROB) && (GetDistanceNM() < 50.0)
2
SendMessage(3.0, "Enemy Withi...")
ProcessingDone()

Fig. 4. SimBionic Authoring Environment & Trigger_NearByEnemy Behavior

SimBionic extends the usual notion of finite state machines by making it possible for states to refer to other finite state machines hierarchically, to define modular algorithms that can be combined powerfully. SimBionic software also provides four extensions that increase the power and expressiveness of the basic engine: global and local variables, interrupt transitions, "blackboards" for sharing knowledge among finite state machines, and polymorphic indexing for run-time selection of algorithms.

IISM uses the SimBionic visual AI code generator platform to instantiate intelligent modules that track target paths, assess threat, and identify targets. For example, the Trigger_NearByEnemy behaviour found in IISM, see Figure 4 (right image), is a schema for interacting with the possible enemy labelled RED, within some predefined distance. This behaviour is called when tracking data of the target are consistent with type RED and calculated “distance” from own ship. It invokes an action to contact the target by messaging. Other more complicated behaviours are invoked for identifying targets as friend or foe, for tracking specific targets over time, and for rejecting nonsense/insane data.

4. IISM Detailed Capabilities

IISM has been implemented using C++, and SimBionic. SimBionic can output its behaviors as C++ code for fast execution. IISM utilizes this facility to create a fast executing AI-based solution. Not all of the major capabilities or requirements utilize SimBionic, so listed first are those major capabilities or requirements that do not exploit SimBionic, and then those that do are described.

Intelligent Tactical Memory
One of the important functions that humans currently provide in the CDC/TFCC is that of intelligent memory and IISM mimics this capability. This memory includes all track attributes (position, velocity, ID information, etc.) along with a time stamp for each. Current ship systems do not keep, in a readily recalled format, the trajectory and ID history of each track. IISM fulfills this purpose.

System Independence
If tactical decision systems go down, IISM will continue to remember (and update from other sources if possible) the current tactical picture. This memory function is important for rebuilding the tactical picture. IISM is set up to take inputs from multiple sources.

IISM Reliability
IISM is required to be very robust, never crashing and able to run around the clock without requiring reboots. To handle the cases of hardware failure, IISM constantly backs up its memory to disk and automatically restore it upon start up.

Human Computer Interaction (HCI)
Most of the HCI occurs through the Advanced Battle Station (ABS). This way watch station personnel do not need to learn anything new, the information will appear in the same manner as if the current human decision makers had provided the information.

4.1 SimBionic Supported Capabilities

SimBionic is used to support IISM’s core capabilities of automating the task of intelligent track analysis. The track’s position and velocity with historical information, if any, regarding position, velocity, proximity and other interactions with other platforms is analyzed by IISM to estimate the probability of hostile intentions of and assess the threat posed by the track. Whenever a track significantly changes its velocity, analysis is made to determine if the maneuver warrants a change in the current ID estimate. Considerations include existing ship and air lanes, motion toward or away from blue forces or the assets that they are protecting, whether tracks appear to be cooperating, and attacks. For example, consider two tracks proceeding together at high speed. One breaks off and mingles with local fishing traffic. Later the other attacks. IISM will warn the watch stander about the other track. If the attack track has merged with other tracks, IISM will notify the user of which ones are possible enemy. IISM can reason from process of elimination as the non-enemy tracks are IDed to identify the remaining possibilities.

For example, the Track Id Processing Behavior (TIPB) is a hierarchal decision tree to classify the track into one of the ID categories (BLUE, RED, GRAY, WHITE) with a given certainty level by analyzing current information as well as historical information of the track, see Figure 5 (left image). TIPB has 3 top-level behaviors: Surface Track Behavior for analyzing surface tracks, Air Track Behavior for analyzing air tracks and Undersea Behavior for analyzing undersea tracks. When IISM receives new updates for the track it runs through TIPB.

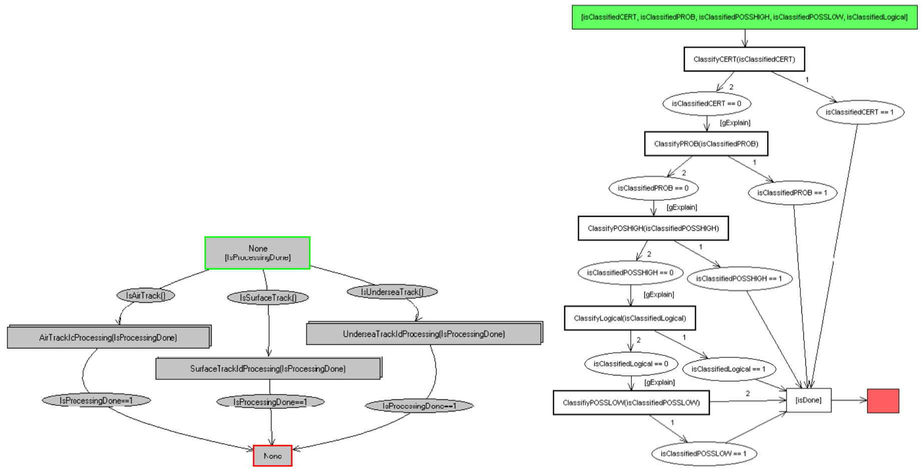
None
[IsProcessingDone]
IsAirTrack()
IsSurfaceTrack()
IsUnderseaTrack()
AirTrackIcProcessing(IsProcessingDone)
SurfaceTrackIdProcessing(IsProcessingDone)
UnderseaTrackIdProcessing(IsProcessingDone)
IsProcessingDone==1
IsProcessingDone==1
IsProcessingDone==1
None
[isClassifiedCERT, isClassifiedPROB, isClassifiedPOSSHIGH, isClassifiedPOSSLOW, isClassifiedLogical]
ClassifyCERT(isClassifiedCERT)
2
1
isClassifiedCERT == 0
isClassifiedCERT == 1
[gExplain]
ClassifyPROB(isClassifiedPROB)
2
1
isClassifiedPROB == 0
isClassifiedPROB == 1
[gExplain]
ClassifyPOSHIGH(isClassifiedPOSSHIGH)
2
1
isClassifiedPOSSHIGH == 0
isClassifiedPOSSHIGH == 1
[gExplain]
ClassifyLogical(isClassifiedLogical)
2
1
isClassifiedLogical == 0
isClassifiedLogical == 1
[gExplain]
ClassifiyPOSSLOW(isClassifiedPOSSLOW)
2
1
[isDone]
isClassifiedPOSSLOW == 1

Fig. 5. Track Id Processing Behavior& Surface Track Id Processing

Now looking at the Surface Track Behavior, see Figure 5 (right image), it consists of five behaviors:

- ClassifyCERT
- ClassifyPROB
- ClassifyPOSSHIGH
- ClassifyLogical
- ClassifyPOSSLOW

The analysis of the information starts with ClassifyCERT and goes through ClassifyPOSSLOW if the track cannot be classified by any of the behaviors.

The following details some of the reasoning techniques used to perform the intelligent track analysis.

Track History Maintenance

Memory is also used to correlate previous tracks with new track information. A complete track history is kept, which allows IISM (or a human operator) to quickly determine if the track's ID is ambiguous because of a track merge or ID swap. Several mistakes, during naval exercises, caused by merges and swaps resulted in the targeting of several neutral, and even blue, platforms. Such mistakes during exercises cause commanders to limit their own options during future exercises or real missions. They are much less likely to use a weapon like the Harpoon, since they lack faith in their own ID picture. Although these problems are rare during random or benign scenarios (tracks don't normally pass that close to each other), a real adversary will go out of his way to try to create them. E.g., a terrorist attacking platforms under US protection would try to mingle, possibly several different times, with commercial platforms, such as fishing boats and merchant traffic. IISM has algorithms implemented with SimBionic that will handle the most complex set of merge/split scenarios (e.g. platforms merging with several different platforms and each other at separate times) logically and correctly. These algorithms already outperform humans in their ability to determine the possible IDs of tracks involved in several merges.

Historical Comparison

A track's history is kept in varying levels of detail, depending upon its age. IISM will remember all tactical data (to different levels of detail, minutes, hours, days, months, or even years before) and compare the current data, events, and situation to the recent or distant past. IISM will retrieve tracks similar to the current one and make recommendations accordingly.

Multiple Competing Hypotheses for ID

IISM keeps simultaneous competing hypothesis for each track as to the type/hull of the platform and its country of ownership. It will track both positive and negative evidence and reach both positive and negative conclusions. IISM explicitly keeps track of all possible hypotheses and the associated likelihoods for each track. Initially, a track can be anything, but incoming evidence impacts the certainties of each hypothesis. Positive ID information, such as a good visual ID, eliminates the competing hypotheses until the track is involved with a merge, at which time the resulting tracks each contain all the hypotheses of both tracks that merged.

Hierarchy of possible ID values

For both dimensions of ID information, IISM will include a hierarchy (from general to specific) of possible ID values. E.g.:

- Blue – UK, Combatant – frigate – FFG-7 – Specific platform; or
- White – Merchant, Cargo Carrier – Ship Class – Specific Hull

ID is often hierarchical with the goal of determining the most precise value that is worthwhile. Thus while an ID of White Merchant might be adequate, a Red Combatant may need to be IDed more precisely, perhaps as Chinese Houdong Fast Patrol Boat. These hierarchical symbols interact with the competing hypotheses described above. Thus, if the only competing hypotheses for a track are Gray Destroyer and Red fast patrol boat, and information is received that it has a speed greater than is possible for a destroyer, then IISM will conclude it is red.

Sanity Checking

When new data is received, before the track information is updated, the new data is compared to the recent history to make sure it makes sense and is at least physically possible. Any inconsistencies are reported, and to the degree practical, automatically resolved. This sanity checking function occurs for red, blue, gray, and white forces. IISM compares the current position/velocity to the last reported position for that track and determines if it is physically possible, given the platform type. If not, it determines if it is most likely a spurious data point, that the assigned track type is wrong, that a completely different platform as been assigned the same track number, or that the reported position of a friendly track is incorrect. It then recommends the appropriate action.

Fuzzy Reasoning

Classify Logical of the surface track behavior is an example of the fuzzy reasoning used by IISM. It will analyze the trajectory of the track to try and classify what kind of platform it is. Please refer to Figure 6

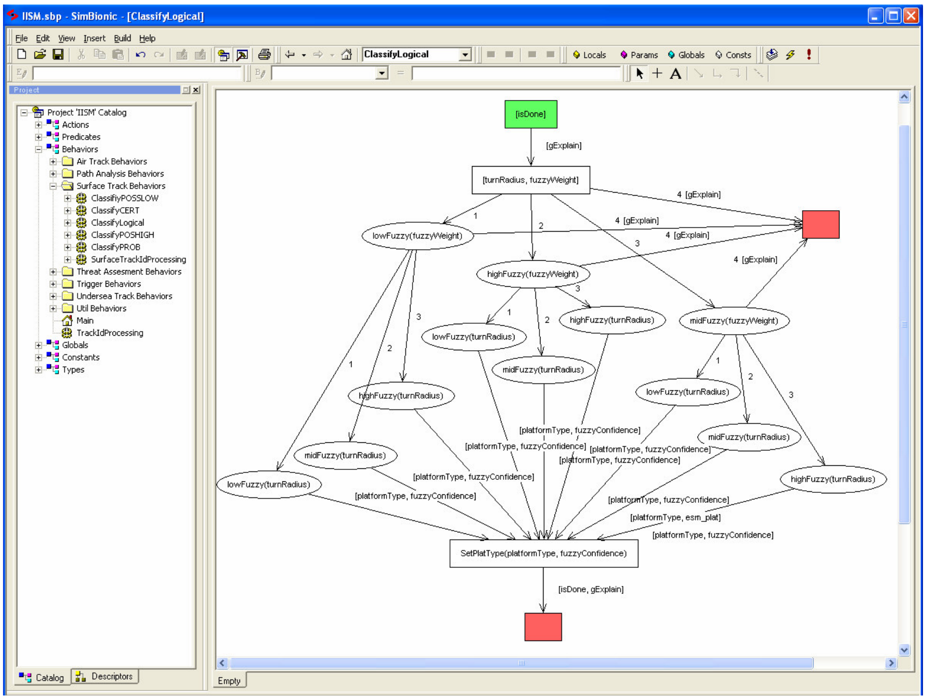

Fig. 6. Classify Logical Fuzzy Reasoning Behavior

In this behavior, first the turnRadius and Weight of the track is estimated based on the history of the trajectory. Next these numbers are converted into one of three fuzzy values, representing heavy, light, or middle weights, and small, middle, and large turn radii. The reason we use fuzzy values for the calculations is because this algorithm now becomes much more robust in the presence of noise or other negative factors. Finally, the platform type is recommended with various fuzzyConfidence levels depending on the fuzzy values. For example, if we have a low weight and high turn radius, we are PROB small light platform, and similarly if we are high weight and low turn radius we are POSHIGH large platform. The reason the large is only poshigh while the small is prob is because a large ship cannot move quickly, but a small ship can, thus we are more confident a ship is small when it moves quickly than that a ship is large when it moves slowly. This kind of intuitive reasoning is only possible via fuzzy reasoning.

Process of Elimination Reasoning

IISM employs logic and the process of elimination in making ID decisions. For example, IISM may know one combatant is out in a particular area where several other tracks are present. Even though every track seems to have low probability of being a combatant based on their behavior, a higher probability bias is used since one of them must be the combatant. The process of elimination is used to determine the most likely tracks to investigate first.

5. Conclusion

IISM is an AI module that alleviates the burdens placed on battle commanders by tracking sometimes ambiguous target signals, storing and handling past target data, assessing threat levels of targets, filtering out insane data, as well as robustly recovering from crashes and errors. IISM's rule-based logic is used to compute track IDs, estimate threats, and notify users of alert conditions; its probabilistic hypothetical reasoning system keeps track of multiple track hypotheses based on the fusion of evidence from multiple sources, and uses statistical algorithms to find correlations between track movements. IISM is a seamless enhancement to the current Advanced Battle Station, providing enhanced reasoning without the need for any user to learn a new system. By applying multiple soft computing techniques including Bayesian inference and fuzzy reasoning, as well as other AI techniques, including polymorphic finite-state-machines, IISM is performing as well as or better than Navy personnel.

References:

1. Navy Warfare Development Command, "Sea-Based Theater Air and Missile Defense: A 21st-Century Warfighting Concept", http://www.ndcweb.navy.mil.
2. Hutchins, S. G., Technical Report 1718, Principles for Intelligent Decision Aiding.
3. Salas, E., Cannon-Bowers, J. A., & Johnston, J. H., "How can you turn a team of experts into an expert team?: Emerging training strategies". In C. Zsambok & G. Klein (Eds.), Naturalistic decision making (pp359-370).