Intelligent Identification Software Module (IISM) for the US Navy's Combat Centers

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
http://www.stottlerhenke.com

Abstract—We have developed and continue to enhance our automated intelligent software, which performs the tasks and decision making currently handled by the personnel manning the watch stations in the Combat Direction Center (CDC), the Task Force Combat Center (TFCC), on-board aircraft carriers, and other Navy ships. Integration of information from a variety of sources in a combat station is a complex task; surveillance guards can receive divergent information from on-board radars, sonars, and other sensors, and must assimilate and interpret even conflicting information in a timely manner to relay it up the chain of command. The Intelligent Identification Software Module (IISM) alleviates some of the burden placed on battle commanders by automating certain tasks, such as the management of historical data, disambiguating multiple track targets, assessing the threat level of targets, and even rejecting improbable data. IISM is interfaced to the Advanced Battle Station (ABS). Given tracking data and time stamps from 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 via ABS when necessary, and provides customizable identifications of targets, as well as the threat level of each of these targets. IISM is also capable of correcting errors and recovering snap-shot and history data after unforeseen catastrophes. [1] [2]

On a highly conceptual level, IISM is able to perform these tasks by viewing the target tracking as a classification problem of the threat levels that it assigns to the individual entities present in the situation. It maintains a consistent and reasonably approximate model of several entities' attributes that are only partially perceivable. This, fundamentally, is the task of track handling and analysis currently being handled by human decision-makers. Put simply, IISM 1) determines, to a degree of certainty, the identity of an entity, 2) performs a path analysis of the entities, and 3) infers abstract conclusions regarding the behavior of entities based on their movement over time. Stated another way, both positive and negative evidence is tracked to form multiple, possibly competing hypothesis, and conclusions of these tracks are made through a process of elimination reasoning.

IISM is able to perform this complex task by utilizing certain AI-based solutions. IISM uses SimBionic, a visual authoring tool that outputs C++ code for fast execution. IISM also mimics the intelligent memory provided by current human track-watchers, including all track attributes (position, velocity, ID information, etc.), along with a time stamp for each. IISM also has system independence and will continue to remember the current tactical picture even if tactical decision systems go down, is very robust, has an automated system backup and restore function, and can even be saved to a file server to diminish physical vulnerabilities.

IISM is an AI module that alleviates many of the burdens placed on battle commanders. It is a seamless enhancement to the current Advanced Battle Station, providing enhanced reasoning without the need for users to learn a new system.

TABLE OF CONTENTS

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. The reason for this is the complexity of the weapon systems and associated information, as shown by the high-level organization of it in the 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.

[1] 0-7803-9546-8/06/$20.00© 2006 IEEE
[2] IEEEAC paper #1512, Version 7, Updated Jan. 03, 2006

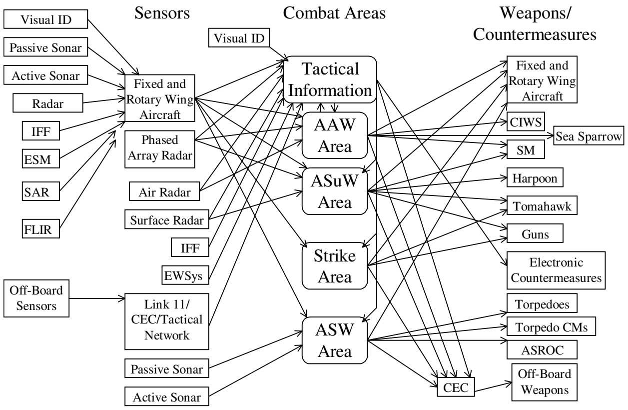

Figure 1. Weapon System High Level Overview

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. 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. 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. We devised an intelligent CDC/TFCC equipment control, monitoring, processing, and fusion system. From the knowledge engineering and the schema we designed and implemented IISM using C++, and SimBionic, a visual AI development tool that can output C++.

2. IISM 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 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.

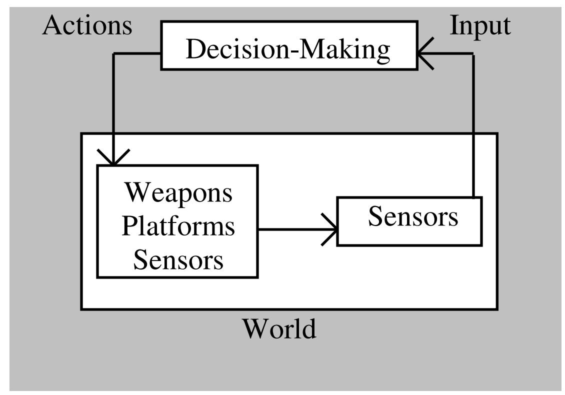

Figure 2. Human Tactical Decision-Making

Perception, Situation, Decision

The *Decision-Making* box in Figure 2 can be broken down into the three tasks of the *perception, situation,* and *decision*, as shown in Figure 3.

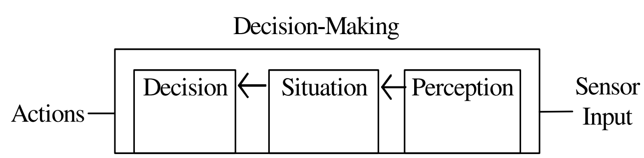

Figure 3. Decision-Making Box

The **perception process** takes sensor input, as well as the history of changes in the tactical picture over time. Since IISM is concerned with the behavior of tracks, IISM monitors the activities of all tracks, analyzing in real time their activities into tactical components. At the lowest level these components include flight segments, simple maneuvers, and weapon and countermeasure deployments. At higher levels, they include known air combat tactics and other indications of threat over time.

The **situation process** includes the tactical picture of all tracks provided by the perception process (including any ID information), behavioral histories of those tracks, and IISM's current assessment of each track along three dimensions:

- Hostility: suspicion of track hostility as evidenced by behavior over time
- Threat: immediacy of possible threat to ownship or protected assets
- Associations: friendliness or hostility toward other platforms

The assessment mechanism for each of these dimensions is described below. It is important to note that for the purposes of this discussion, the term “attribute” is not used with the conventional meaning in the TFCC, as an attribute that can only be declared by a high authority. Here we use the term only to refer to IISM’s private estimation of the tactical implications of a track’s behavior.

The **decision process** serves to assess the hostility, threat, and associations of tracks over time, and also to report significant values of these parameters to the appropriate decision maker. Along with maintaining track histories, IISM also maintains a history of its own assessments for each track, and the reasons for those assessments (e.g., a particular track is considered hostile because it was an unknown that accelerated towards a friendly at a particular time, so that it can later explain its assessments to a human or even to another IISM).

Threat— Threat represents immediacy of possible attack. IISM employs a metric combining the platforms capabilities, so far as they are known, with its current position and velocity, determining the earliest moment at which its weapons envelope will include ownship or protected assets. Tracks of high threat, even if of unknown hostility, will receive higher priority over other tracks.

Associations— Associations of tracks are assessed in order to represent the apparent hostility or friendliness between them. When an association is formed or altered, their hostilities are propagated along this association; in other words, consorting with the enemy makes a track more likely to also be an enemy.

Associations are assessed by interaction such as the proximity of two tracks over time, common points of origin, sustained proximity at high speeds, or hostile acts between them. Rules and Case Based Reasoning (CBR) can be are to detect associations and determine their strengths. These mechanisms are well tested and conservative, because many associations are difficult to determine; for instance, a hostile platform following a commercial craft could together appear as if they were in formation.

The primary mode of action other than performing the assessments described above, is communication. It employs fuzzy logic, tempered by the current tactical situation, to decide at what thresholds of threat it will report to a decision-maker. Such a report can include the expected hostility of an otherwise unknown track, the possible imminence of its attack, and explanations of its own assessments; for example, it may report “Track Y has moderately high likelihood of hostility, low immediate threat, or strong association with hostile track X.” It can provide the same kind of explanation in response to queries, such as “X is considered probably hostile because it was initially detected inbound from enemy country Y and in formation with probably hostile track Z.” If necessary, IISM can provide an event history supporting its assessments.

Classifying Threats of Entities

On a high 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, the update algorithm is called on that track. This update algorithm will use 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 Scenario— Let's take a situation where two surface tracks (track 1 & 2) are first detected, they are traveling at a high speed for surface vessels (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 a track 3, that 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. So 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 small snippet of the situation, the situation is very fluid and multiple hypotheses must be tracked and re-evaluated as new information is obtained.

In Figure 4 through Figure 6 below, the dash dot dash lines (—.—.—) are shipping lanes with some random ship traffic. The solid lines represent track paths. The dashed line (— — —) represents a track "jumping" instantaneously to a new position. Circles represent where tracks merge or split. Blue, red, gray, and white represent the ID category of the track. Blue forces are friends, red forces are adversaries, gray entities are military neutral, and white entities are non-military neutral.

1. In Figure 4, track 1 and 2 enter together at high speed (50 knots) [IISM recommends some platform types based on their speed] and the tracks diverge and Track 1 begins towards a direction close to a shipping lane.

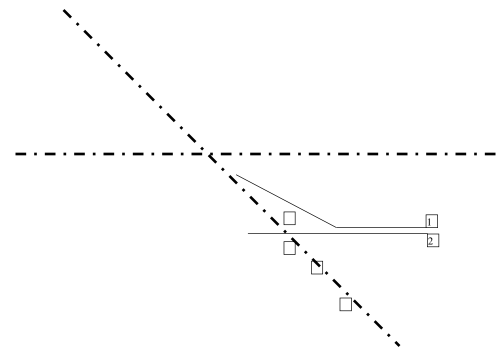

Figure 4. State 1 of Unfolding Scenario

2. In Figure 5, track 2 Merges with Track 3 (previous IDed Grey (Indian Houdong Fast Patrol Boat)). After splitting one heads North and One heads South.

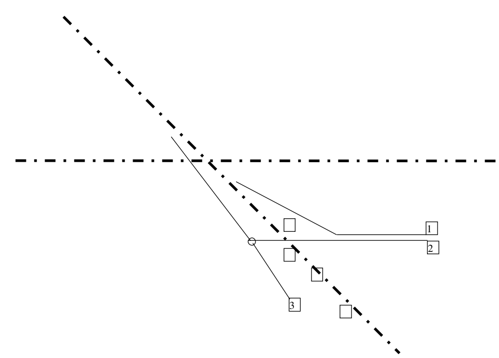

Figure 5. State 2 of Unfolding Scenario

3. In Figure 6, track 4 is a previously IDed Merchant heading West along a shipping lane. Track 4 merges with the southbound track and resulting tracks continue South and West.

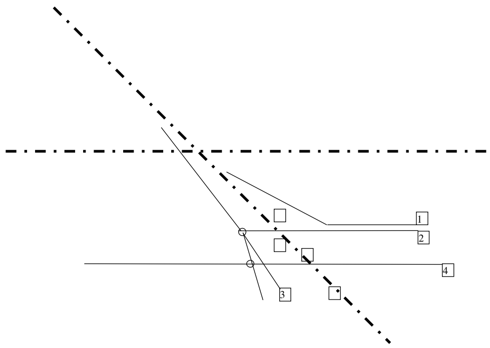

Figure 6. State 3 of Unfolding Scenario

4. Meanwhile, North bound track merges with Track 5 (previously ID'ed fishing boat). Resulting Tracks Head West and Northwest

5. Track 6 is a previously ID'ed **Chinese Houdong (Red) and Track 7 is a Merchant. Tracks 6 and 7 Merge and** Split and each resulting track separately merges/splits with the southernmost west bound track. Northernmost westbound track is IDed as the Fishing boat.

6. Track 1 breaks from Lane toward CV [Up hostility, up threat, issue warning]

7. Southbound Track is IDed as a Houdong [IISM positively Deduces that Northwest bound track is Red Chinese and issues threat warning as it is approaching the weapons release distance (though obliquely)], [southbound Houdong deduced to be Indian (Gray) Houdong]

8. Two Blue air tracks intercept two Red air tracks all merge into a "furball" and two tracks emerge [appropriate warning given (including possible high threat)]

9. A very high altitude high speed track labeled an F14 [system responds with a misclassification warning and suggests ballistic missile]

10. Surface track proceeds, we get unique EM [system responds with PossHigh] Later get unique AE [system responds with Prob]. Later get VID [and system responds with CERT]

11. Blue track 9, out of radar coverage, appears to jump 100s of miles. [System responds with a physically impossible message and recommends the position be confirmed.]

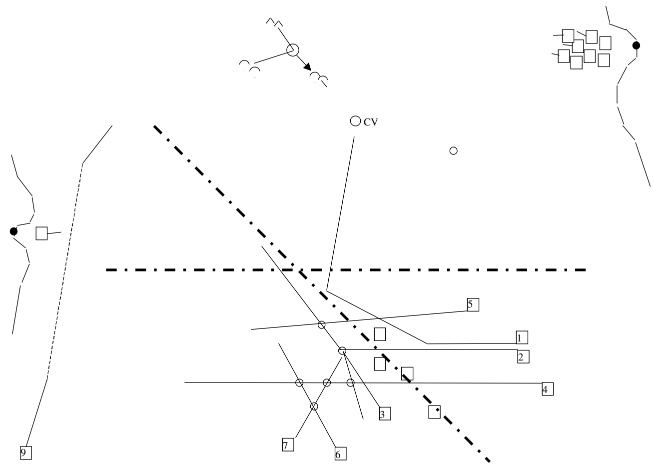

Figure 7. Example Scenario

Figure 8 shows an area around the Persian Gulf and provides an idea of how cluttered the environment being monitored and assessed can be.

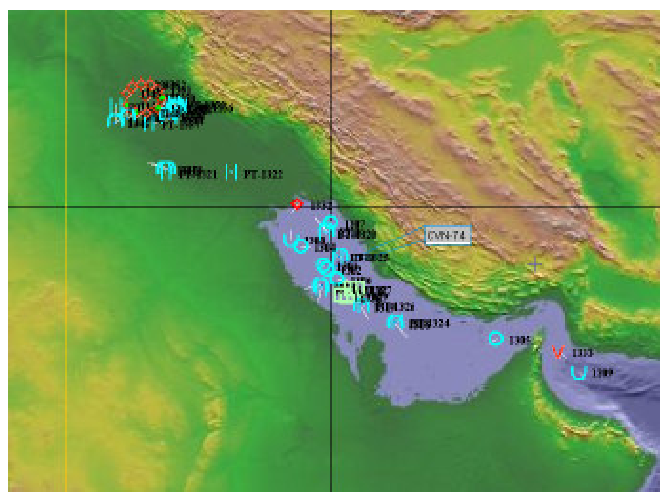

Figure 8. Example of Density of Contacts that Need Monitoring and Assessing

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. The user defines a basic vocabulary of actions and predicates which appear as textual and geometric shapes on the canvas. The actions correspond to states in an FSM. Predicates are used to determine valid transitions between states. The basic model is extended in two major ways. First, algorithms are hierarchical in that they may invoke each other. Second, each algorithm may have a number of specializations indexed through a descriptor hierarchy. These two extensions serve to encapsulate functionality, and to selectively specialize algorithms whenever necessary without arduous re-modification of existing algorithms.

SimBionic employs four programming constructs:

- actions, which define all the different actions the algorithms can perform;
- algorithms (also referred to as behaviors) that string together actions and conditional logic;
- predicates, which set the conditions under which each action and algorithm will happen; and
- 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 9, users can visually create algorithms by drawing actions and invoked 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.

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.

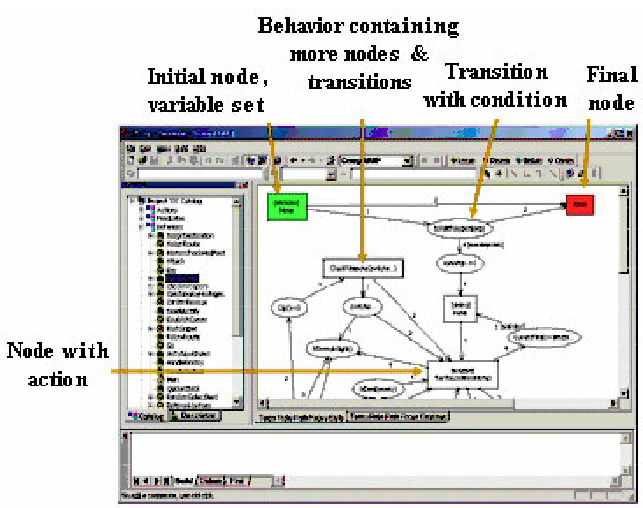

Figure 9. SimBionic Authoring Environment

IISM uses the SimBionic visual AI code generator platform to instantiate intelligent modules that track target paths, assess threat, and identify targets. In SimBionic, actions are connected to other actions or to conditions by connectors that describe the logic of the program. A set of actions and conditions that form a logical whole is called a behaviour. Behaviours that perform common tasks can be reused in different programs.

Take, for example, the behaviour of a driver who must figure out whether to stop at a traffic light. As Figure 10 shows, the program semantics are well-captured by a diagram built in SimBionic. The green traffic light is where program execution begins. The ovals indicate condition nodes and the rectangles represent actions. In this simple example, the SimBionic application generates a simple program consisting of a 3-cased if statement.

A more complicated example involves the Trigger_NearByEnemy behaviour found in IISM, see Figure 11, a schema for interacting with the possible enemy that is labelled RED and within some predefined distance. This behaviour is called when tracking data of the target are consistent with 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.

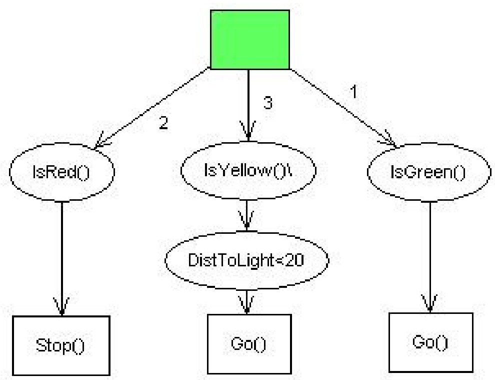

Figure 10. Simple Traffic Light Behavior

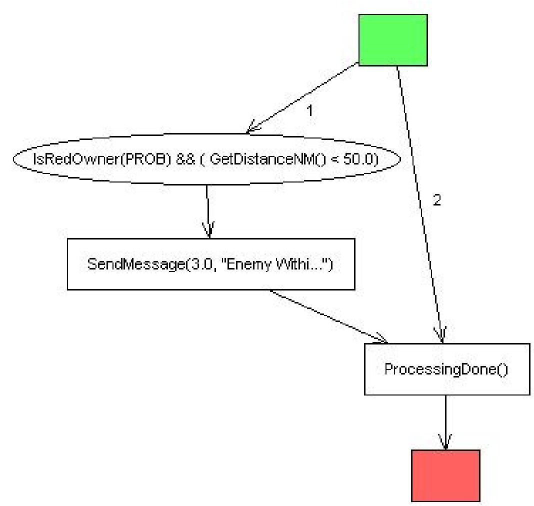

Figure 11. Trigger_NearByEnemy Behavior

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 first are listed 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. As described above, 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. IISM is constantly tested to validate that it meets this requirement. In addition, to handle the cases of hardware failure, IISM constantly backs up its memory to disk and automatically restores it upon start up. IISM can be configured to save to a file server in which case an ax can be taken to it (or less dramatically the plug pulled) and it can be brought up on another machine with no loss of tactical information. IISM saves frequent snap shots of all its data and processing results and keeps a log of every event since the last snap shot. On start-up it automatically checks the most recent snapshot and reconstructs the events since that snap shot was taken.

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.

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 nonenemy 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. 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.

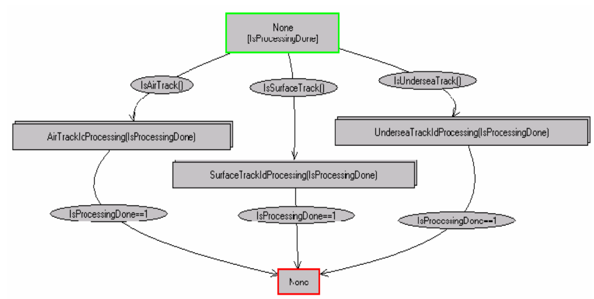

Figure 12. Track Id Processing Behavior

Now looking at the Surface Track Behavior, see Figure 13, 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.

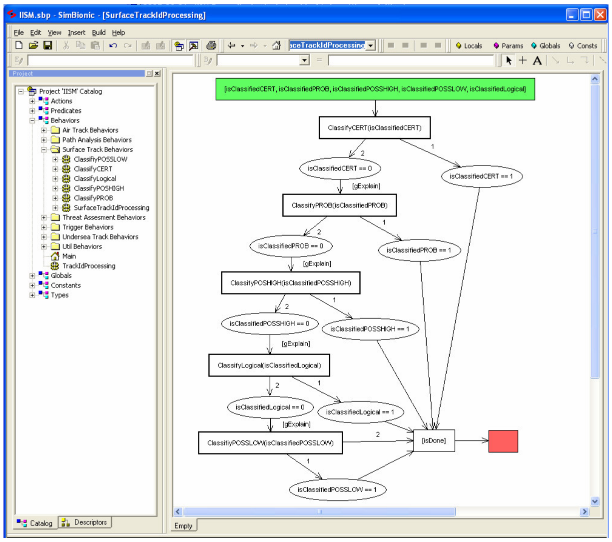

Figure 13. Surface Track Id Processing.

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 their 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 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 has 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— The Classify Logical Behavior of the Surface Track behavior is an example of the statistical 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 14 below:

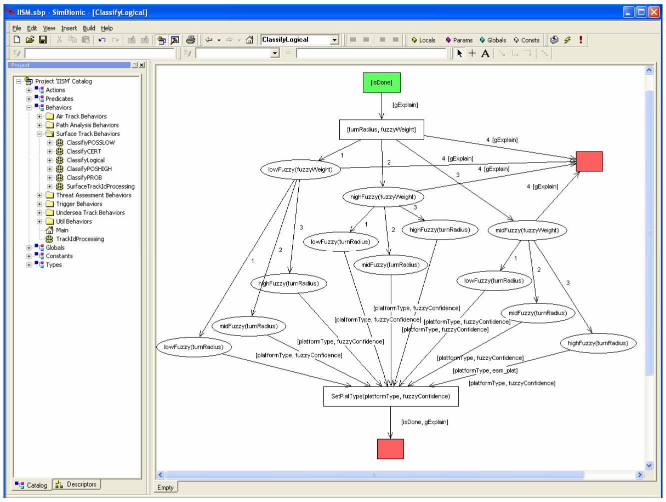

Figure 14. Classify Logical Behavior

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 statistical values, representing heavy, light, or middle weights, and small, middle, and large turn radii. The reason we use statistical 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 plaftform, 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 sure 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 with the statistical reasoning that is done here. With statistical reasoning we get more realistic results that are more correct more of the time.

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, and filtering out insane data. IISM uses SimBionic, a visual authoring tool that outputs C++ code for fast execution. IISM also mimics the intelligent memory provided by current human track-watchers, including all track attributes (position, velocity, ID information, etc.), along with a time stamp for each. IISM has system independence and will continue to remember the current tactical picture even if tactical decision systems go down, is very robust, has an automated system backup and restore function, and can even be saved to a file server to diminish physical vulnerabilities. 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.

REFERENCES

[1] Hutchins, S. G., Technical Report 1718, Principles for Intelligent Decision Aiding.

[2] Maher, T., "Tactical Action Officer (TAO) Intelligent Tutoring System (ITS) Version 1.0 Student's User Manual", Stottler Henke Associates Inc., July 1998.

[3] Maher, T., "Tactical Action Officer (TAO) Intelligent Tutoring System (ITS) Version 1.0 Instructor's User Manual", Stottler Henke Associates Inc., July 1998.

[4] Navy Warfare Development Command, "Sea-Based Theater Air and Missile Defense: A 21st-Century Warfighting Concept", http://www.ndcweb.navy.mil

[5] 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 Naturalistic decision making (pp359-370).

BIOGRAPHY

Robert Richards is a Principal Investigator and Project Manager at Stottler Henke. Current and past projects range from training system development spanning from aviation to medicine, to applying automation and artificial intelligence techniques to data and voice network configuration and optimization, to machine learning techniques for real-time data mining, and to decision support tool development for high-stress life-critical situations such as landing signal officers on aircraft carriers. He received his PhD from Stanford University in mechanical engineering with an emphasis on machine learning and artificial intelligence.

Richard Stottler co-founded Stottler Henke Associates, Inc., an artificial intelligence consulting firm in San Mateo, California in 1988 and has been the President of the company since then. He has been principal investigator on a large number of tactical decision-making projects conducted by Stottler Henke, including projects for the Navy, Army, Air Force, and Marine Corps. He has a Masters degree in computer science specializing in Artificial Intelligence from Stanford University.

Ben Ball is a lead software engineer at Stottler Henke and was a lead developer for IISM. The work included developing intelligent algorithms for determining platform identity, and for filtering noisy information. Other work includes development of hardware for miniature robotic platforms, and construction of electronic systems consisting of sensors, power supplies, microcontrollers, actuators, and memory storage. Ben has a B.S. from Stanford University in Electrical Engineering with a Controls Specialty.

Coskun Tasoluk is a lead software engineer at Stottler Henke. His primary interests are autonomous agents, intelligent training systems (ITS), and planning and robotics. He has participated in the development and design of a wide range of artificial intelligence systems at Stottler Henke, and is currently the lead developer for a Dismount Soldier Simulator (DSS) Intelligent Structured Trainer. He holds an M.S in electrical engineering and an M.S in computer science from Drexel University.