Intelligent Surface Threat Identification System (ISTIS)

Richard Stottler, Ben Ball, and Robert Richards
Stottler Henke Associates, Inc.,
951 Mariner's Island Blvd, Suite 360
San Mateo, CA 94404
{Stottler, Bball, Richards}@stottlerhenke.com
http://www.stottlerhenke.com

Abstract—[12]Stottler Henke in conjunction with the US Navy has developed and continues to enhance the Intelligent Surface Threat Identification System (ISTIS). ISTIS improves the surface threat ID process, quality, and efficiency in the Littoral Combat Ship (LCS) Surface Mission Module. This improved performance includes better use of scarce ID resources, better ID estimations from available information, sooner ID determinations, ID accomplished at a greater range, prevention of ID "surprises", and improved operations in more complex environments. Other LCS applications and other ship types can also benefit from the techniques.

ISTIS is needed because the type of threat that the US Navy faces has evolved, the problem of identification has gotten more complex and severe. Rather than a military entity that tends to operate far from civilian traffic, the current and future threat will tend to operate close to and hide within groups of civilian surface craft and even utilize such craft themselves. This causes an overload in information to US Navy personnel tasked with performing the identification (ID) function. Additionally, especially in the littoral environment, the number of tracks detected by radar will be far greater than the number of sensors available that can perform reliable ID. Specifically, the manned and unmanned aircraft used to perform visual ID are simply far too few in number to ID all the detected surface tracks.

ISTIS, based on Artificial Intelligence (AI) techniques, automatically analyzes the data associated with a track, hypothesizes, draws inferences, and makes ID related recommendations. These data include the tracks' location and/or velocity reported over time and other ID related reports such as IFF (Identification, Friend or Foe) codes, visual ID reports, acoustic signature, specific emitter ID (SEI), ELINT (Electronic Intelligence) signature, FLIR (Forward Looking Infrared Radar) reports, intelligence reports, communications, etc. ISTIS is also based on many years of research and development on related projects including: the Intelligent Identification Software Module (IISM). IISM has a well-developed merge-split, multiple hypothesis maintenance and reasoning system based on Truth Maintenance techniques and process of elimination reasoning. This operates in conjunction with an uncertainty representation and reasoning system such that there can be degrees of uncertainty associated with the multiple hypotheses for a track. IISM performs Threat and Trigger Processing using user definable, graphical behaviors. These components are embedded in an agent architecture to facilitate communication between and separate development of the components. In addition, in the LCS Undersea Decision Support project IISM components have been adapted to the undersea warfare area for the Littoral Combat Ship (LCS). Several threat assessment agents can run in tandem, providing assessments of the situation from various points of view (e.g. ownship, battle group, mission, etc.).

ISTIS itself implements several capabilities including: 1) Track Merge-Split and Fade/Reappear Processing using Multiple Hypotheses Reasoning to prevent ID swap and engage in process of elimination reasoning. 2) Maneuver Correlation Detection to ID tracks maintaining an intercept course, working cooperatively, avoiding an intercept, etc. 3) Behavior Analysis and Surface ID Data Fusion and Processing to automatically determine the likely platform type, its affiliation and intentions and associated certainty. 4) Action Recommendations based on Heuristics for ownship, UAV, UGV, and helicopter maneuvers and actions. 5) Route Planning to ID a set of contacts and/or search an area. 6) Ability of tactical personnel to edit and change ID, alerting, and recommendation behaviors in real-time.

In this paper, specifics have been replaced with unclassified values and concepts when discussing particular examples.

TABLE OF CONTENTS

1. INTRODUCTION

As the type of threat that the US Navy faces has evolved, the problem of identification has gotten more complex and severe. Rather than a military entity that tends to operate far from civilian traffic, the current threat will tend to operate

[1] 1-4244-0525-4/07/$20.00 ©2007 IEEE
[2] IEEEAC paper #1283, Version 7 , Updated December 21, 2006

close to and hide within groups of civilian surface craft and even utilize such craft themselves. This causes an overload in information to US Navy personnel tasked with performing the identification (ID) function. Additionally, especially in the littoral environment, the number of tracks detected by radar will be far greater than the number of sensors available that can perform reliable ID. Specifically the manned and unmanned aircraft used to perform visual ID are simply far too few in number to ID all the detected surface tracks.

What is required is an intelligent examination of each track's behavior and other information available to determine the most suspicious and most threatening tracks. These can then receive priority for the limited ID sensor resources. This examination would need to address a number of aspects. Considerations include whether the track is proceeding on a known commercial shipping route or at least directly between two known commercial ports, has the track recently veered off such a route, is it operating in known fishing grounds, is it proceeding toward friendly military surface platforms or other assets protected by those platforms, and is it exhibiting military surface behaviors. Some analysis involves possible relationships between the track of interest and other ones and involves looking for correlations between maneuvers. Considerations involving these correlations include whether two or more tracks are operating in formation or otherwise cooperating, whether a track is avoiding military ships, is attempting to intercept another track, or trying to maintain a particular distance from another track. A thoughtful observer might see suspicious correlating behavior and recommend course and/or speed changes to see if a response is elicited from the suspicious track. Such a response tends to confirm the suspicions.

Other issues relate to the track data itself. When two tracks get closer than the radar system's ability to discern them, they merge. When they again are further away, they split. If the tracks have been previously IDed, they should now be treated as ambiguous where each can be hypothesized to be either of the original two. Upon getting confirmation of the ID of one of them, the ID of the other can also be resolved. This merge-split hypothesis reasoning can get very complicated if tracks are involved in several merge-splits. Track ambiguity is also created by tracks that fade and reappear within a close enough proximity that they could have switched places in the time they were untracked. Track fades are very common with the small craft of concern to the LCS. An exponential number of hypotheses need to be maintained and complex process of elimination reasoning is required as ID information comes in. Scenarios involving multiple tracks involved in even just 2 or 3 merge-splits will exceed a human's ability to reason about them in real-time. Although these problems are rare during uncluttered, 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 and the littoral environment tends to have a large number of tracks which provide ample opportunities. (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.) The problem is also exacerbated by existing tracking systems which typically ignore the merge-split problem. Typically the merged (i.e. combined) track will be assigned the track number of one of the merged tracks with no indication that it consists of at least two craft. Upon splitting, one of the tracks will usually be assigned a completely new number and one will be assigned the merged track's number (which was the number of one of the original tracks but may or may not correspond to the correct craft). Thus, if the ID was known for a particular track number that was then involved in a merge split, this track number, and the associated ID, may be transferred to the other craft involved in the merge split. This is known as "ID swap" and is very insidious, since the casual observer will think that the track in question has been positively IDed. Only careful, constant observation of each individual track path will reveal whether a track has been involved in merge-splits or possible ID swaps.

Another problem relating to the track data is caused by noise and errors. These include duplicate tracks (often caused by multiple platforms with slight position errors sensing the same track), velocity spikes (usually caused by small accumulated position errors being corrected all at once), and track jump (caused by the same track number being used for different craft or incorrectly reported positions, in the case of friendly craft). Obviously incorrect data must be recognized as such and handled appropriately. A related problem is certainty in the data and associated inferences. For example, ID-related information short of visual ID in good weather often leads to classification certainty of "Probable" or "Possible". Obviously this complicates hypothesis management and process of elimination reasoning.

So with enough personnel and sufficient time (perhaps one dedicated to each track) the behavior of every track could be analyzed and just the suspicious and threatening ones investigated. There are too many tracks and not enough personnel to pay close attention to all of them. What is required is the human expertise described above implemented in software and effectively replicated for every track. This is the realm of artificial intelligence and ISTIS automates much of this human expertise.

2. ISTIS FUNCTIONAL OVERVIEW

The Intelligent Surface Threat Identification System (ISTIS) is based on Artificial Intelligence (AI) techniques and the automation of the practices of the best ID experts. ISTIS automatically analyzes the data associated with a track, hypothesizes, draws inferences, and makes ID related recommendations by processing trajectory and other ID related data in a number of levels as shown in the figure below. Each level consists of a number of components embedded in an existing agent architecture which facilitates

communication and requests between components and ensures balanced processing. Each component is described briefly here, then in more detail in subsections below. The Data Feed manager simply handles data incoming from several different sources and puts it in a common format and location. The first real level of processing filters, cleans, and transforms the incoming data for use by the higher levels and includes Path Segmenting and Sanity Checking. Sanity Checking serves both as a filter for obviously incorrect data and to correct misconceptions from previous erroneous data. Path Segmenting breaks track trajectories into relatively straight segments separated by maneuvers. It also splits track data based on merge-splits and initiates some of the processing in other components (analysis based on either the beginning or ending of a maneuver or merge-split).

The next level seeks to draw conclusions about what various tracks are and what actually happened, i.e. type of craft a track is, what organization owns it (country, civilian, terrorist), its true path through various merge-splits or fade-out incidents, and whether it has any type of relationship with other craft. ID processing examines ID related data (visual reports, IFF, FLIR, etc.) and makes its best determination as to the type of platform and likely ownership in a hierarchical way and attaches a level of certainty to those determinations. Path analysis essentially performs the same function by examining the known path of platforms and comparing them to known commercial routes and areas, and to the paths of other platforms. It also estimates the hostility of the track, based on intentions inferable from the platform's path. Merge-Split Processing manages the creation of hypotheses resulting from merge-splits and/or fade-reappears and performing the process of elimination reasoning as ID information is received in order to resolve the ambiguity in other, related tracks.

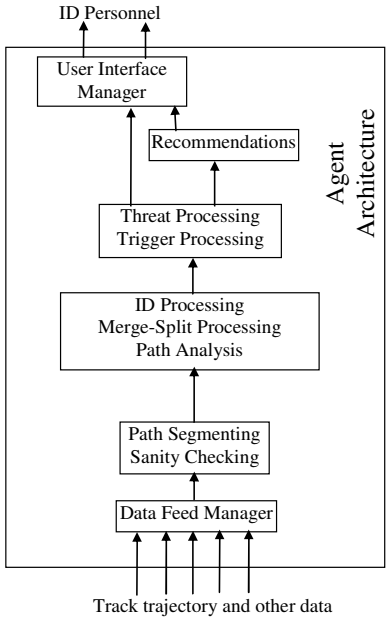

Figure 1. ISTIS Architecture

The third and highest level of processing is more geared to understanding the intent of sensed platforms and translating that into an estimation of the threat they pose. Specifically threat processing uses the likely owner and platform type along with relevant weapons release distances, sensor ranges, speeds, etc. to estimate the level of threat posed by each track to each friendly or protected asset. Trigger processing examines all data associated with a track and determines whether any triggers should be executed. These are typically defined by the warfighter to short-circuit some of the reasoning. For example, a typical trigger states to notify the responsible human anytime a platform is proceeding at high speed toward a friendly or protected platform, regardless of any other conclusions that were drawn. Presumably this protects from mistakes, such as a mistaken visual ID of a threat platform.

The recommendations layer examines the tactical situation and the information state of each track and makes ID related recommendations. For example, which tracks should be visually IDed first and with what assets and possible maneuvers for friendly surface assets either to gather additional ID data directly or indirectly, such as observing unknown tracks' responses to those maneuvers. These recommendations as well as other important information are sent to the User Interface Manager for routing to the appropriate personnel.

3. CAPABILITIES

Merge-Split Processing for Radar Merges and Track Fades and Reasoning through Multiple Hypotheses

This capability is a high priority because of the density of traffic and the fact that the platforms of interest are small boats with poor radar returns. As tracks merge and split (or fade out in relative near proximity to each other then re-appear later), the system must keep track of the fact that each outgoing track could be either of the incoming tracks. It keeps multiple hypotheses for each track. This processing must handle cascading situations where the incoming tracks are themselves the outgoing tracks of previous merge-split situations and therefore already include multiple hypotheses. As ID information is received, sufficient to disambiguate the various hypotheses, process of elimination reasoning is used to automatically infer the identity of related tracks.

Maneuver Correlation Detection

This capability analyzes track maneuvers to determine if they are correlated with recent maneuvers of friendly forces or each other. This is important because the number of tracks in the littoral environment is too large to allow a human observer to detect correlations of maneuvers of different tracks. The type of correlations the system looks for between a track and a friendly ship are whether it is trying to avoid an intercept or close point of approach, trying to maintain an intercept course, following, trying to maintain a specific distance, or trying to avoid a detection or identification by avoiding a particular CPA (closest point of

approach) distance. This capability also looks for cooperation between tracks such as maintaining a formation, rendezvous, or a third track that has rendezvoused with each of them.

Surface ID Processing, Fusion, and Behavior Analysis

Because of the large number of tracks involved, it is impossible for a human watchstander to adequately monitor and process the ID reports for and behavior of the individual tracks. As ID report information is received (VID, FLIR, Acoustic, SEI, ISAR, ES, etc.), this information along with sensed and speed and agility must be used to automatically determine the likely platform type and its affiliation with each individual surface platform and its intentions. The behavior of the platform including any merchants turning off of known shipping lanes, fishing boats not heading toward fishing areas, pleasure boats moving toward non recreational areas, turning toward friendly ships, interception course of friendly ships, successive merge-splits, cooperative behavior with other platforms, and correlations of maneuvers with friendly forces, should all be considered when determining the likelihood of enemy affiliation and hostile intentions.

Action Recommendations based on Heuristics

There are many heuristics, or rules of thumb, that can be used to make specific recommendations for specific actions in specific situations. These include ownship maneuvers to avoid a threat, elicit a response, or buy time; UAV, USV, and Helicopter employment to ID, elicit response, or cover a suspicious track or provide escort; employment of non-lethal capabilities; and external communications such as radio queries to approaching traffic or 5 horn blasts. In making these recommendations, ISTIS will have to consider traffic; geography such as dead zones (areas that are in radar "shadows" or otherwise can't be currently observed; underwater topography (safe/unsafe maneuver areas); environments such as storms, visibility, and sea state; which systems are up and available; the diversion issue; and the fact that there will be different doctrine for different types of missions (area defense, convoy, carrier battle group defense, etc.) even for the surface warfare mission package.

Additionally, automatic help managing the resources under the LCS's control should also be provided. This includes allocating specific ID tasks to specific ID resources (such as a specific UAV) for specific times taking into account proximity, flight times, priorities, and the limited resource quantity. This capability takes the recommendations described above and determines how they can be best accomplished.

Route Planning to ID a Set of Contacts and/or Search an Area

Given a set of contacts to be classified and/or identified and/or an area to be searched, this capability plans a close to optimal route for the selected manned or unmanned vehicle.

Ability of Tactical Personnel to Edit and Change ID, Alerting, and Recommendation Behaviors in Real-time

Because the LCS will be applied to a very wide range of scenarios and situations, it needs to be the case that the tactical personnel can alter the behavior of the system to fit the current circumstances when required. The behaviors that control most aspects of the system exist in graphical form similar to a flow chart. This representation was specifically designed to be easily understood and modified by tactical personnel with no programming experience.

4. IMPLEMENTATION DETAILS AND METHODOLOGIES

There are a number of AI and other techniques that we have applied to this problem and which is used in ISTIS. Fuzzy Logic (FL) and behavior transition networks (BTNs), which were originally developed to simulate real-time decision-making in tactical situations, are used to analyze track behavior and ID related information. These FL rules and BTNs typically utilize various vector operations applied to track velocities, relative positions, and other vectors to determine intercepts, closest point of approach, relative velocities, traveling along known routes, or veering off routes. When a maneuver completes, an analysis is done relative to recent maneuvers of other tracks to determine whether the new segment's velocity, relative to the velocity of the track before the maneuver in relation to another track that recently maneuvered, tends to maintain a nearby parallel course (formation), maintain an intercept, maintain a particular distance, or maintain an avoidance course where that avoidance was degraded by the other track's earlier maneuver. Multiple maneuvers by a track correlated in the same way to another track or group of tracks (e.g. multiple maneuvers of a single track all correlated to maneuvers of friendly military that all tended to restore avoidance from the friendly military tracks) increase the level of suspicion.

Truth maintenance software manages the multiple possible hypotheses resulting from a merge-split, including hypotheses with various levels of certainty, and performs the logical process of elimination reasoning. ISTIS, based on expert system and planning technology as well as human-derived heuristics, would make recommendations for active actions or employment of sensors. A system that took those general recommendations and selected specific resources to execute them at specific times would be based on constraint-based, resource selection and optimization, and scheduling technology which we have applied in numerous domains, including scheduling of ELINT sensor resources.

The different methodologies are discussed more fully in each subsection below; refer back to Figure 1 to see how the following fit into ISTIS's multi-level objective.

Path Analysis

Path Analysis exploits the fact that most commercial vessels proceed in straight paths. They typically proceed in the most direct path from their departure to their destination at their most efficient speed. They also do not generally react to the movements of other ships (other than to give way). Unexplained maneuvers inherently invite further processing. Path Analysis looks at all aspects of the path a vessel takes. It typically uses fuzzy logic rules to do this. For example consider the following fuzzy logic statements:

```
If Track turns off commercial route
Then
{
    If Track is heading toward High Value Asset
    Then
    {
        Hostility.Certainty=inversely proportional to CPA;
        Threat.Value = inverse of time to weapons release
        range;
        Threat.Certainty = PROB
        Increment Owner Red Certainty;
    }
    Else
    {
        If Track is heading away from intercepting Blue
        Track
        Then
        Increment Owner Red Certainty
        Else
        Decrement White Certainty
    }
```

Where Turns Off has a value of 0 if there is no maneuver or the maneuver results in a new heading less than 10 degrees off the route and Turns Off has a value of 1 if the new heading is greater than 20 degrees off the route and is linear between 0 and 1 when the angle is between 10 and 20 degrees.

The semi standard color classification for ownership is used where *blue* represents friendly military, *gray* represents neutral military, *white* represents non-threat commercial, and *red* represents opposing military, insurgents, or terrorists. Each may be further broken into specific countries. E.g., Red might include China and North Korea and Gray might include India and Egypt.

High Value Asset (HVA) is any Blue track or a track, point, route, or area being defended by blue forces.

Heading Toward is a fuzzy number between 0 and 1 and is the vector dot product of the track's velocity with its relative position to the thing being compared to, scaled to the track's assumed max velocity, or 0 if that dot product is negative. In other words if it is heading directly toward the thing being compared to at it's maximum assumed speed then the value is 1 and as the relative velocity goes to 0, Heading Toward also goes to 0.

So to the degree that a surface track has been following a typical commercial route, turns off of it, and starts heading toward a high value asset, then our certainty in its hostility will be increased to a high value if it has a close closest point of approach (CPA) and a lower value if the CPA is distant. The degree of the threat will be high if it is almost to its weapon release range (less than 10 minutes) and lower if there is more time. The certainty that the track does represent a threat will be probable and we will increase our certainty that the owner of the track is an opponent.

The Certainty levels are 0, Possible Low, Possible High, Probable, and Certain, where certain is "as sure as you can be in war" which means it doesn't correspond, logically, to 100% certain. Occasionally "certain" facts will be found out to be wrong and ISTIS is designed to handle these types of situations.

The above logic also considers the possibility that an opponent has turned off the commercial route to avoid a blue track (making it more likely to be an opponent but not necessarily a threat). And (though not shown here) even if the track is neither approaching an HVA nor avoiding a blue track, if it is turning off a commercial route, it is less likely to be commercial.

Other fuzzy logic rules check if any track is approaching an HVA even if it hasn't turned off a commercial route, which is somewhat less threatening since less deception appears to be involved, and similarly checks if it is trying to avoid blue forces or seems to be maintaining a set distance to a particular blue track. Fuzzy logic rules also utilize interception calculations which are somewhat different than Heading Toward, since a more sophisticated enemy will take the velocity of the track that it is intercepting into account. In fact, the best intercepting course may be nearly orthogonal to the relative position vector (and therefore have a Heading Toward value of near 0). This would occur when using a lower velocity to intercept a faster moving craft by "cutting in front of" it. This type of interception is especially aggressive and threatening.

Other relationships between tracks are also examined besides interception, avoidance, constant distance or heading toward. Some involve possible cooperation between platforms. These relationships are formation, rendezvous, transport between (two tracks each rendezvous with the same third track), and same point of origin. Getting ID information about one track of a cooperating pair allows some of the same information to be inferred for the other track of the pair albeit with lower certainty. The certainty is scaled to the degree of certainty in the relationship. For example, two tracks proceeding together in formation at high speed for a relatively long time have a high degree of cooperating certainty. If one is later found to be a threat, the other will too, with the same high degree of certainty the system has in the relationship.

A final consideration and also segue into the next section is that a track that maneuvers multiple times in order to merge-split with other tracks is likely trying to engage in deception which is inherently suspicious.

Merge-Split Processing

The Merge/Split ID tracking system keeps track of the travel paths of objects. In particular, it is responsible for resolving identities after two or more tracks merge and split. For each track, it keeps track of the possible predecessors, and gradually narrows these predecessors down, depending on which are logically consistent with the current hypotheses. When only one direct predecessor is left, it sets that predecessor as the parent, and propagates the change to other tracks at the same level (note that it cannot only consider the other tracks resulting from the split, because setting the parent has ramifications for the ancestor lists as well). The ancestor lists are necessary (rather than a simple predecessor list) for those cases that the direct parents may remain ambiguous but the more distant relations can be established.

Uncertainty is represented by a certainty value on the hypothesis between zero and one, where zero means the hypothesis is definitely false, and one means the hypothesis is definitely true. As tracks merge and split, the identity of new tracks are initially unknown. Suppose tracks T1 and T2 merge and split to form tracks T3 and T4. It is unclear whether T3 matches T1 or T2, and the same holds for T4. Besides the identity of T3 and T4, we may be concerned with hypotheses about various attributes of the tracks such as country of ownership and the platform type. T3 and T4 will have the hypotheses from T1 and T2 until new information disambiguates their identity or updates the hypotheses. The certainty module maintains this history of merges and splits and updates the track hypotheses. Based on new information about the T3, the certainty module will attempt to disambiguate whether T3 matches T1 or T2. If so, then T4 will also be updated to have the correct predecessor track.

Under the current structure for hypotheses, each hypothesis is discrete, containing a single combination of platform and owner; these are then collected in a list. Within a given hypothesis, the values are described hierarchically. E.G. a specific type of fishing boat is a fishing boat is a commercial craft. FFG-8 is a FFG-7 is a fast frigate is a small military vessel is a military vessel. North Korea is a Red country of ownership.

In Figure 2, a single hypothesis is represented by a Certainty Hypothesis (CH) object. All lists are of variable length, although the Pair List will always have two values (one for platform and one for owner). The two certainty pairs will each express that a property (Name provides the property name such as Owner or Platform) has s value represented by a symbol or hierarchy of symbols. The Certainty Value represents the nesting of certainty values through multiple merges and splits. Each certainty value corresponds to one of the possible tracks that a merged-split track might be and represents the certainty that the track is the platform type and has the owner specified by the pair list. Many of these will probably be 0, when the tracks that entered the merge split were definitely known not to be something (probably because they were known to be something else). Each Certainty Value includes not one, but two values. One is the base value, set by things such as assertions and ID processing; it sets the general level of certainty. The mod value is the value that should be used to adjust (increment of decrement) the base value, such as that provided by Path Analysis (e.g. decrement a White hypothesis when a track turns off a shipping lane).

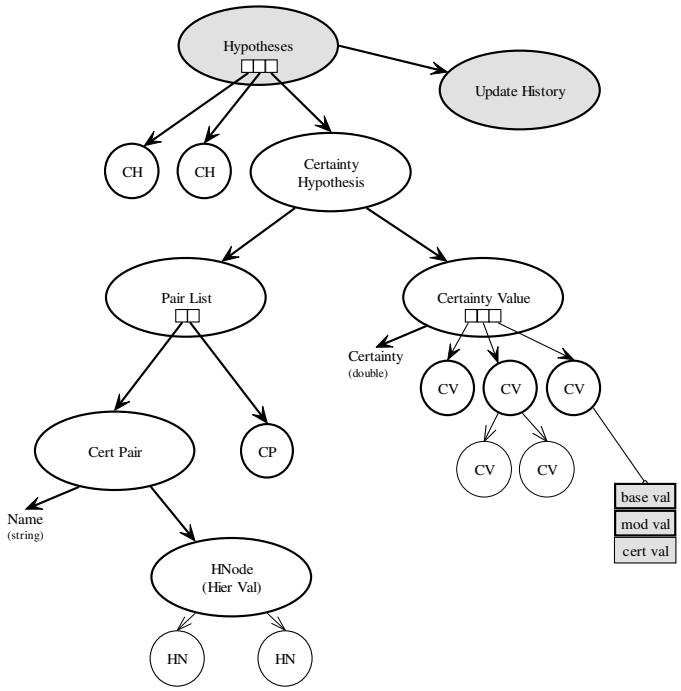

Figure 2. Hypothesis structure

Figure 5 shows an actual example of the hypothesis structure at an intermediate stage during a scenario being processed by the Merge-Split component. Table 1 shows one of the midpoint states, at the point of maximum confusion. There are three initial platforms, T1, T2, and T3, which have merged and split multiple times so that T6 might be any one of the three. Furthermore, each of T1, T2, and T3 had multiple hypotheses. For example T1 had a 64% certainty of being Red and a 41% certainty of being White. Each line in the table represents the certainty that each of (T3, (T1,T2)) fits the hypothesis. So, for example looking at the first line, T3 had a 68% certainty of being White and T2 had a 0% certainty of being white.

Table 1. Merge-Split Example

(68,(41,0))[(Owner = White) & (Platform = Any Platform)] XOR
(50,(0,20))[(Owner = Gray) & (Platform = Any Platform)] XOR
(0,(64,83))[(Owner = Red) & (Platform = Any Platform)]

The hypothesis structure from Table 1 is mapped in Figure 3 to the hypothesis structure already presented. (For space reasons, the white section of the hypothesis is not broken out). Each line in Table 1 is represented by one certainty hypothesis. Note the 20, 0, 50 certainty values and associated gray owner on the left in the figure below and how it corresponds to the gray hypothesis line in the table.

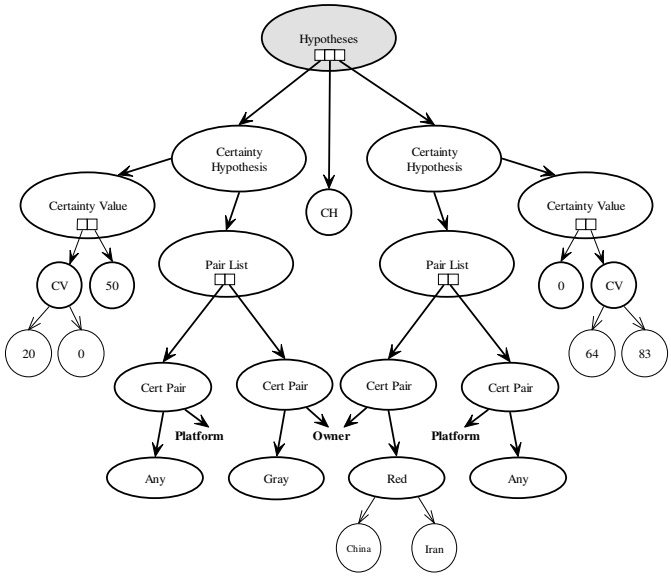

Figure 3. An actual example of the hypothesis structure used

An example of the merge-split functionality incorporated into ISTIS is provided in [1].

ID Processing

The ID processing application determines what platform type a track is or who the owner might be and how likely those estimations are. This analysis is performed by BTNs that can be graphically edited by the user, providing a high degree of flexibility. The current behavior 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. New data on a track causes the analysis to be re-run. The current behaviors are summarized below.

A surface track is classified with certainty, CERT (certain), if the track has been continuously tracked and one of the following is true for the track:

a. Is visually sighted with consistent ID information for exactly one ID category in its history
b. Has IFF valid mode IV (then classified as BLUE) or
c. Has an SEI (specific emitter identification) hit (in which case the ownership is determined by the platform type).

The track is classified with confidence level PROB (probable) if the track satisfies either of the following conditions:

a. If track is previously classified with confidence level CERTAIN but has now not been continuously tracked (then it is classified as the previous classification) or
b. If track is classified to Fine Naval Class by any visual sighting (then the ownership is set to the owners of that platform type).

The track is also classified with confidence level PROB (probable) if the track satisfies any two of the following conditions:

a. Consistent track history
b. Unique electro-magnetic emissions
c. Unique acoustic emissions
d. Position coincides with intelligence data
e. Unclassifiable visual sighting

The track is also classified with confidence level POSS HIGH (possible, high) if the track satisfies any of the following conditions:

a. Consistent Track history
b. Classified to the Gross Level
c. Unique electro-magnetic emissions
d. Unique acoustic emissions
e. Position coincides with intelligence data

The track is also classified with confidence level POSS LOW (possible, low) if the track satisfies any of the following conditions:

a. ISAR (Inverse Synthetic Aperture Radar)
b. FLIR (Forward Looking Infra Red)
c. ES (Electronic Support)
d. Unclassifiable visual sighting

Threat Processing

The Threat Processing component determines when a tracked element poses a possible threat, and notifies the user. The threat analysis is handled using BTNs which can be edited graphically by the user. Threat Processing looks at a potential threat and makes determinations as to the

seriousness of the threat, based on a variety of considerations. For example, if an unidentified ship is approaching a friendly vessel, determinations are made whether this ship is friendly or not, what type of weapons it may be carrying and whether it intends to do harm. Based on these determinations, a threat level will be determined.

Threat determination considerations:

- The Red Owner certainty level is the most important feature in determining the threat. We can safely ignore platforms that we are certain are White (non-threat commercial) or Blue (friendly military).
- If the track is Gray and has a high hostility level, then it will be treated as a Possible Red.
- Sensor Range
- Weapon Release Range
- It is assumed that to attack a friendly platform, that platform must be within both the sensor range and the weapon release range of a threat platform, though not necessarily the same threat platform (cooperative engagement).
- Time until within weapons release and sensor range
    - <= 2 minutes IMMEDIATE_THREAT
    - 2-10 minutes HIGH_THREAT
    - 10-60 minutes MEDIUM_THREAT
    - > 60 minutes LOW_THREAT
- No weapons implies NO_THREAT, except for the case in which a platform without weapons is providing targeting information for another platform with weapons.

Data Sanity Checking

The sanity checker verifies that incoming data is reasonable, given the current internal state. If it is not, it tries to determine whether the data is useful, or if it is just noise. Much of the data received by the data feed manager may be noisy or erroneous. The Sanity Checker is responsible for identifying insane data, and determining whether it is noise or is in some way useful.

Sanity checking occurs with each track update, verifying that the given point could reasonably belong to the specified track. There are two basic reasons that a new piece of data could be "insane": the data is incorrect, or the system's picture of what is going on is incorrect. The duty of the Sanity Checker, then, is twofold: first, to determine that data is insane; second, to determine whether it is noise or an important piece of information that can be used to correct the track model. The sanity checking itself is fairly straightforward: is the new movement of data reasonably possible, given the known track information? If it is not, the Sanity performs basic interpretation checks which are:

1. Threat Processing. If the insane data indicates movement defined such that it will intersect an HVA, it should be assumed to be a threat, and the system should not wait for additional information (which may well be too long in coming) to notify the user. This is primarily defined as BTN, and leverages the existing threat processing behavior.
2. Cumulative Error Correction Check. One of the most common reasons for "insane" data is that the internal picture of the situation is incorrect because of cumulative error. Many of these cases can be checked fairly easily; if any of them come up positive, the track should be updated with the "insane" data, possibly with a flag showing that it is a correction to earlier cumulative error.
3. Inconsistency with Identity. The milder forms of insanity may only seem insane because the system thinks it knows what the track is (platform and owner), and the behavior is inconsistent with that. In such a case, the system should wait for a bit more information before notifying the user (since it passed the threat processing). If the inconsistency holds through multiple updates, this checker notifies the user that the identity is questionable, and requests confirmation.

The Sanity Checker will include a repository of insane data, against which it can check new data. This will be the first step when new information comes through, even before it is checked for insanity. This will help the Sanity Checker either compound the information (since two insane points usually express interesting information, but one is just noise) or weed out genuine noise.

Trigger Processing

The Trigger Processing Component processes a number of trigger BTNs defined graphically by the user. Most of these behaviors consider the standard behavior for the supposed type of the track (i.e., commercial), and compare that to what the platform is actually doing. If there is a notable discrepancy, it notifies the user. This is similar to (and may be somewhat redundant to) but separate from Threat Processing to provide an independent mechanism of alerts. For example the user may decide to create a behavior that notifies him anytime a platform is heading toward any blue platform at high speed and at a close intercept course, regardless of the determined ownership of that track. Thus if a threat platform was somehow able to get itself misclassified (e.g., appearing to be a fishing boat to the helicopter that visually IDed it) as commercial, the user would still be notified of the dangerous behavior. (If the speed of the supposed fishing boat exceeded the expected maximum speed for the type fishing boat it was supposed to have been, that would also be cause for automatic notification and automatic reprocessing of the ID. A better example would be a threat platform getting itself misclassified as a fast commercial craft).

The Trigger Processing watches the tracked elements for unexpected, unusual, or interesting behavior. The "rules" for

what constitutes "interesting behavior" for a variety of platforms and owners is defined in a set of graphical behaviors to give the users maximum flexibility over this fail-safe component. The majority of the behaviors of interest are suspicious behaviors, but the trigger capabilities are by no means restricted to this category.

Recommendations

ISTIS includes automatic recommendations for ID related actions. There are varying degrees of sophistication. A simple set of heuristics are implemented as rules or BTNs. These are as simple as recommending that high threat tracks be visually IDed. More complex BTNs are used to recognize certain types of situations and recommend the actions appropriate for those situations. BTNs are often used independently and in parallel, where each examines the situation from a particular perspective.

Behavior Transition Networks (BTNs)

BTNs are used to create intelligent behaviors by dividing the behaviors hierarchically into tasks connected with transitions. The current task executes until one of its outgoing transitions becomes true. Then control transitions to task indicated by the true transitions arrow. If multiple outgoing transitions from one task are true at the same time, they are evaluated in order as indicated by the numbers in the BTN on the right. Tasks with a heavy outline are themselves BTNs which can be further expanded.

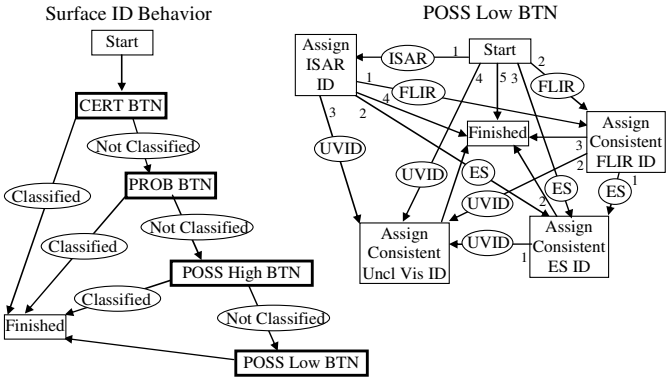

Figure 4. Example Behaviors

For example the Overall Surface ID behavior is shown above on the left. It begins in the Start Task and immediately transitions to the CERT BTN. The dark outline of CERT BTN indicates that it is a BTN itself, but it is not shown. If the CERT BTN is able to classify the track then the BTN will transition to the Finished Task, since Classified will be true. Otherwise it transitions to the PROB BTN. The effect of the Surface ID behavior is to attempt to classify the track with the highest certainty first then try to classify it with successively less certainty until it is either classified or all behaviors are exhausted. The POSS Low BTN, shown with a dark outline on the left is expanded on the right. This happens to have the same number of Tasks but a more complicated set of transitions. Also all Tasks in the POSS Low BTN are atomic, they execute directly without being further divided as defined in another BTN. From the Start Task, this BTN will transition immediately to the Assign ISAR ID if ISAR sensor data exists. If it does not, the existence of FLIR data for the track will be checked. If this does not exist, Electronic Support (ES) data will be checked. If this does not exist, then Visual ID information will be checked. Only in the case where the visual ID was unclassifiable would the POSS Low BTN be called. If an Unclassifiable Visual ID exists, BTN will transition from Start To Assign Consistent Uncl. Vis. ID. If none of these 4 types of data exist for the track, the fifth link from Start to Finished will be followed, thus ending the behavior. This BTN has many transitions to cover the cases where multiple pieces of data exist for the track. These Tasks all assign an ID based on particular sensor values with certainty at most POSS Low. If multiple data exists and is contradictory, different tasks may assert different competing hypothesis. If it is possible to find a platform classification that is consistent across the different sensors, the individual tasks will do so. Note that each BTN here is implicitly passed the track being processed. Though not shown here, different Tasks and BTNs can communicate with each other using a variety of means.

Truth Maintenance

Truth Maintenance refers to techniques relating to keeping track of multiple competing hypothesis, multiple sets of consistent hypotheses (sometimes called "worlds"), and the dependencies between hypotheses. Truth maintenance systems use nonmonotonic logic, where facts are not just added to the logical structure but also retracted. (The fact that the number of facts sometimes decreases, i.e. does not monotonically increase, leads to the name.) Very general truth maintenance systems are computationally intensive. However for the ID problem, we utilize the fact that dependencies between hypotheses are only of two kinds. The most obvious is the exclusivity principle – a track can only be one of the competing hypotheses for a track. In fact we purposely group orthogonal dimensions (such as platform and owner) into one hypothesis so that hypotheses have to compete. The second type of dependency between hypotheses is based on the fact that if one track resulting from a merge-split, where the incoming tracks were both known, becomes known, the other resulting track must be the remaining hypothesis. Our system also makes use of the fact that there will be a relatively small number of merge-splits active at any one time. The merge-split logic draws logical conclusions based on incoming information.

Fuzzy Logic (FL)

One type of rule representation which seems well suited to tactical problems, especially sensor data interpretation, is Fuzzy Logic (FL). In FL, rules are captured which reference qualitative, inexact, or fuzzy values such as High, Low, and Medium. For example, a fuzzy rule might state:

If the Velocity is High and CPA is Close and the ID is Unknown

Then Assume Enemy

An FL system operates on quantitative data - such as sensor data, and through a process called fuzzification, converts that data to a set of qualitative values with associated fuzzy membership values. The rules referencing these qualitative values are each fired to the degree indicated by the membership values. The results of several competing rules are then combined, making fuzzy assignments of qualitative values. These can be used directly or defuzzified into quantitative data.

5. APPLICATION EXAMPLE

The following example illustrates some of ISTIS's capabilities.

Merge-Split Processing and Reasoning through Multiple Hypotheses

This example includes enough tracks in close enough proximity that several merge-splits occur, including cascading situations, requiring ISTIS to reason through the multiple resulting hypotheses for each track. The current ISTIS is sophisticated enough that it is theoretically possible to disambiguate it well and can handle complexities greater than is possible by human watchstanders.

Maneuver Correlation Detection

The current ISTIS implements the highest priority correlations. It determines if a track is trying to maintain an intercept course with the LCS, following the LCS, or traveling with other tracks. ISTIS has been run in scenarios containing well over 100 maneuvering tracks to show that the maneuver correlation calculation can scale well. There are several tracks intercepting or following the LCS which shows the benefit of an automatic system and the difficulty a human watchstander has picking these correlated maneuvers out from all of the background maneuvering.

Surface ID Processing, Fusion, and Behavior Analysis

The example scenarios include different types of ID reports (VID, FLIR, Acoustic, SEI, ISAR, ES, etc.) and different types of behaviors that ISTIS will detect and include in its analysis including turning off of commercial shipping lanes and turning toward the LCS. ISTIS executes a doctrinal procedure to classify ships with varying levels of certainty.

Action Recommendations based on Heuristics

The current ISTIS includes recommendations for ownship maneuvers to avoid identified threats and buy time, and recommendations to ID suspicious tracks with UAVs.

Ability of Tactical Personnel to Edit and Change ID, Alerting, and Recommendation Behaviors in Real-time

ISTIS was implemented with a set of graphically defined behaviors that can be edited and changed without programming. The current ISTIS demonstration includes an example of editing one of the triggering behaviors that notifies the watchstanders whenever a ship is approaching the LCS at a speed greater than the threshold. In the demonstration the threshold is lowered which leads to additional alerts when run on the same scenario.

Scenario Description

USS Freedom (LCS-1) is tasked with preceding a carrier battle group through the Singapore Strait and the Strait of Malacca in order to protect it from the small boat threat. This will involve building the maritime surface picture, identifying any suspicious vessels, and sweeping the transit route and staging areas within striking distance.

Along the 60 miles of the straits that are of interest to the LCS during the scenario, there are about 50 large merchant ships transiting the straits at speeds which vary from 10 to 20 knots. There are approximately another 100 tracks including fishing boats and pleasure craft. These mostly stay in their respective areas or transition to other, like areas. Some pleasure craft wander out, essentially aimlessly.

We assumed a 1% positional error for range and angle on all radar tracks. E.g., a radar track 10 NM away has positional error of 0.1 NM in both the radial and lateral directions. If separate tracks get within this error distance of each other they are merged. The LCS has 3 UAVs for ID purposes.

There will be 3 attacks that will occur in the scenario:

1) Two hidden swarms of boats: Each has several filled with men with RPGs and with a speed of 50 knots. Each swarm would leave their hidden location at full speed at the time when the transit time is shortest.

2) 2-3 tracks all maintaining an intercept course on the LCS for an extended period of time.

3) Pirate/Terrorist: The scenario will have a pirate attack on a merchant ship.

Additionally there will be 2-3 tracks involved in merge-splits to try to hide their identity.

Demonstration

The current ISTIS:

- Correctly reasons through several merge/split situations.
- Successfully detects the tracks maintaining an intercept course on own ship with a low false alarm rate.
- Utilizes several different types of ID reports to determine the likelihood of a track's platform type and affiliation.

- Determines the likelihood of a track's affiliation and intentions by utilizing the track's observed behavior, such as turning off a shipping lane or closing ownship, and whether it is maintaining an intercept course.
- Makes recommendations based on the heuristics mentioned above.
- Outputs its recommendations and the results of its analysis through a simple user interface.
- The simulation can be halted or sped up to 10x or 50x real-time, which is done during the demonstration to save time.

Demonstration Scenario Events

Figure 5 shows the ground truth map display for the simulator. Mousing over a track shows the track number and additional information. Clicking a track causes track information to be displayed in the Track Information display and continually updated. The Track Information, Recommendations, and Alerts Display, shown in Figure 6, displays alerts and recommendations. The Command Line Display (not shown) allows entry of commands and the display of some of the underlying logic structures.

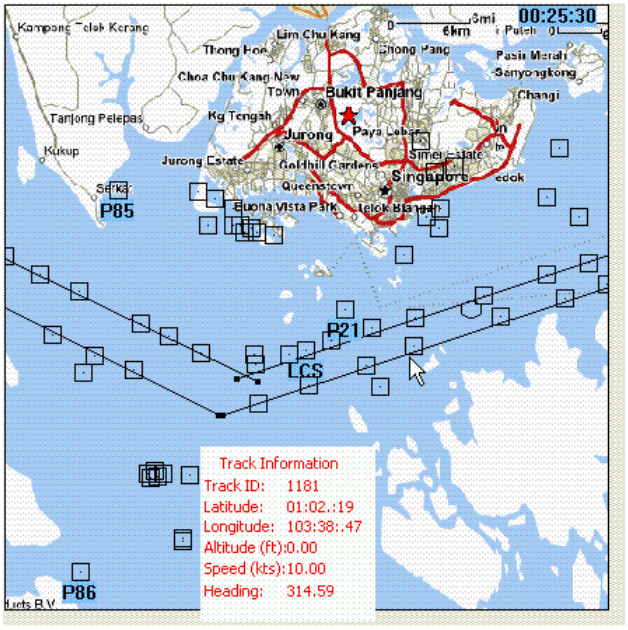

Figure 5. ISTIS Ground Truth Scenario Dynamic Map Display

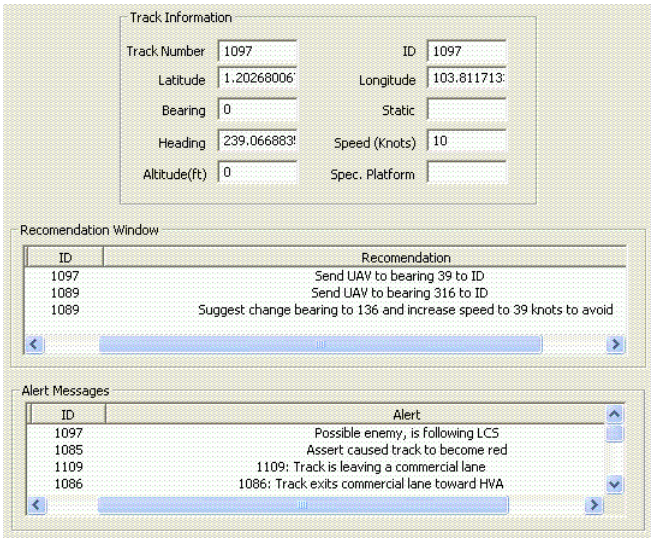

Figure 6. ISTIS Track Information, Recommendations, and Alerts Display

A demonstration described here is available as a video from the authors; however, to show some of the processing that is occurring by ISTIS, the following timeline is provided, reference Figure 5 and Figure 6.

Timeline:

0:46 Show where 1089 and 1086 are.
0:50 1089 is seen as closing LCS (going 40+knots, moving toward LCS, and is close)
1:13 Show 1099 and 1091
1:26 1086 leaves commercial lane as it moves toward LCS, Alert: exit toward HVA
1:29 Remove blue and white from 1099
1:35 Remove blue and grey from 1091
2:16 Show 1083,1084,and 1085 because they will eventually m/s
2:26 Set 1083 owner grey
2:47 Set 1085 red and see the alert that it is red
3:00 Show where track 1097 is (which is following)
3:21 Platform recommender recommends changing 1085 to combatant/small boats (since that is the only possibility left in the knowledge base based on owner and max speed)
3:20 Alert: 1097 is following LCS and recommendation to send UAV to ID it.
4:31 Alert: 1083,1084, and1085 are correlated with each other
4:40 m1099 is formed from the merge of 1091 and 1099; also m1085 is formed from merge of 1085 and 1083, and then m1084 is merge of m1085 and 1084. So, m1084 is merge of all 3 tracks
4:52 Show 1083,1084, and 1085 related hypotheses
5:32 Show where track 1136 is (which will later be an intercept)
5:56 Alert: the cluster near the island just south of Singapore is correlated
6:30 Show m1099 hypotheses
7:12 Alert: 1136 is fast approaching
8:14 m1099 splits
8:17 Alert: 1117 has left the shipping lane (verified by showing the location with the mouse)
8:20 m1084 splits into 2, then soon after into 3 (by 8:34)
9:00 Show m1099-1 and m1099-2 hypotheses are same as m1099 (which is combination of 1091 and 1099)
9:15 Set m1099-1 to grey, and the prototype differentiates the tracks, then show that the correct track's hypotheses match

10:00 Set m1099-2 red, and see that m1099-2 and the associated 1091 become red (from backward reasoning) and get the alerts
10:28 Set m1084-1 to grey and see that the other two are unaffected because there were 2 grey tracks and a red (so can't yet differentiate)
11:00 Set m1084-2 to red. Get the alert for red, and see that all split tracks now have singular hypotheses corresponding to the original 3 tracks
12:47 Alert: 1135 (P85) is on intercept with LCS
13:32 VID on 1101 as a pleasure craft
14:10 Track 1101 CERT
16:24 Alert: 1135 is now shadowing LCS
16:34 Cluster on bottom left starts moving, and one of them is fast towards LCS
18:00 Alert tracks in bottom left group are correlated
19:19 Alert: intercept from a member of the bottom left group
20:44 Alert: tracks in top left group are correlated

6. CONCLUSION

ISTIS is an Artificial Intelligence (AI) solution to the challenges the US Navy is currently facing when tracking ambiguous surface tracks, storing and handling past track data, assessing threat levels of tracks, and filtering out insane data. This solution will help lessen the burden on human watchstanders and assist them in tracking and identifying this type of threat that tends to operate close to and hide within civilian surface craft and even utilize such craft themselves. ISTIS automatically analyzes the data associated with a track, hypothesizes, draws inferences, and makes ID related recommendations. ISTIS is based on many years of research and development on related projects that have, for example, provided ISTIS with a well-developed merge-split, multiple hypothesis maintenance and reasoning system based on Truth Maintenance techniques and process of elimination reasoning.

ISTIS implements several capabilities including: 1) track merge-split and fade/reappear processing using multiple hypotheses reasoning to prevent ID swap and engage in process of elimination reasoning. 2) maneuver correlation detection to ID tracks maintaining an intercept course, working cooperatively, avoiding an intercept, etc. 3) behavior analysis and surface ID data fusion and processing to automatically determine the likely platform type, its affiliation and intentions and associated certainty. 4) action recommendations based on heuristics for ownship, UAV, UGV, and helicopter maneuvers and actions. 5) route planning to ID a set of contacts and/or search an area. 6) ability of tactical personnel to edit and change ID, alerting, and recommendation behaviors in real-time.

REFERENCES

[1] Richards, Robert, Richard Stottler, Ben Ball, Coskun Tasoluk (2006) "Intelligent Identification Software Module (IISM) for the US Navy's Combat Centers", 2006 IEEE Aerospace Conference Proceedings. Big Sky, Montana, March 4-11, 2006.

BIOGRAPHIES

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.*

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.*