INFERD and Entropy for Situational Awareness

MOISES SUDIT
MICHAEL HOLENDER
ADAM STOTZ
TERRY RICKARD
RONALD YAGER

As technology continues to advance, services and capabilities become computerized, and an increasing amount of business is conducted electronically, there is an interesting need for real-time decision-making systems with many capabilities in various domains. In this paper we introduce INFERD (INformation Fusion Engine for Real-time Decision-making), an adaptable information fusion engine which performs fusion at levels zero, one, and two to provide real-time situational assessment. The advantages to our approach are threefold: (1) The level of abstraction in which the analyst interacts with the engine, (2) the speed at which the information fusion is presented and performed, and (3) our ability to give the user the choice to disregard ad-hoc rules or a priori parameters, which have both advantages and disadvantages. We present both a parameterized approach founded in statistical mechanics theory and a non-parameterized approach using concepts in entropy as understood in the context of information theory.

Manuscript received October 12, 2005; revised June 29, 2006; released for publication January 4, 2007.

Refereeing of this contribution was handled by Stephane Paradis.

Authors' addresses: M. Sudit and M. Holender, Center for Multisource Information Fusion, SUNY at Buffalo, Buffalo, NY; A. Stotz, Calspan-UB Research Center, 4455 Genesee Street, Buffalo, NY; T. Rickard, Lockheed Martin, 4637 Shoshone Drive, Larkspur, CO; R. Yager, Machine Intelligence Institute, Iona College, New Rochelle, NY.

1. INTRODUCTION

1.1. INFERD

INFERD was created in the context of cyber security [25] as a decision aid tool to improve the analyst understanding of the situation and ultimately expedite their processing. To cope with the volumes and data rates of current sensed environments such as cyber security and others, decision aid tools must provide their assessment of the situation in a very time efficient manner. In most cases, this time constraint eliminates the possibility of some non polynomial approaches such as optimal inexact graph matching and must instead rely on heuristics to provide *good* results in a timely manner. INFERD's hierarchical fusion approach was developed to do such a task. The two forms of input to INFERD are in the form of a *Guidance Template* (a priori), and sensor data (runtime). The actual fusion process, both a parameterized and unparameterized approach, and how these two forms of input produce valuable output will be addressed throughout the paper.

INFERD's unique approach to Information Fusion can arguably provide these basic advantages: (1) The flexibility of the system to be transitioned to different environments, (2) the level of abstraction of the output of the system compared to the specificity of the models, and (3) the rate at which INFERD can process data and produce results.

1.2. Parameterization v. Non-Parameterization

Parametric approaches are typically general enough to be applied to a variety of environments. Deploying a parametric system to networks of varying topologies usually consists of retraining the system on test data obtained for that environment. The problem of systems using parametric approaches based on training data sets is a sensitive one that can often lead to large numbers of false positives or inaccuracies when working on data not in the training set. The two classical cases of *overfitting* and *overtraining* can arise when a parameter vector v is obtained that configures the system to be very accurate on the training data but generalizes poorly to non-training data. The accuracy/generality tradeoff problem is a well-studied one in many academic areas such as statistics (known as the *bias-variance tradeoff* [15]), Bayesian inference (known as *penalized likelihood* [6], [19]), and in pattern recognition/machine learning (known as *minimum message length* [39]).

Rule-based approaches are expressive in the way that the security analyst provides system configuration. Rules are created or modified in accordance with the environment in which the system is running. This methodology, however, has arguable deficiencies in that every possible condition for the environment in which the system is running must be accounted for in its rule set.

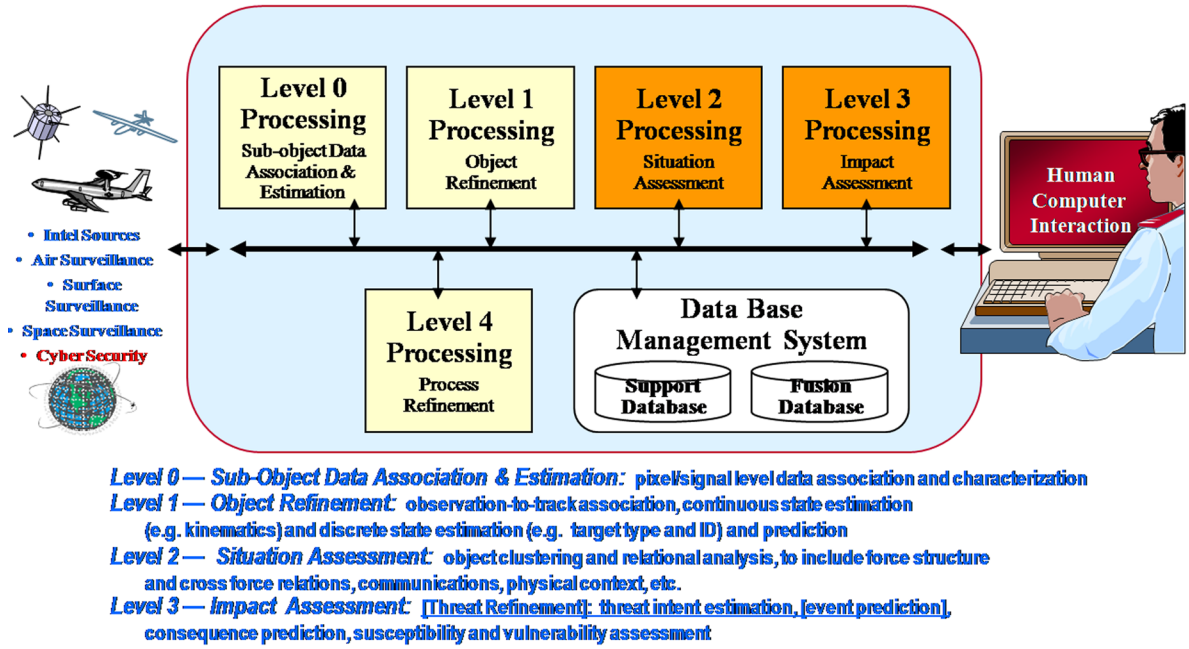


Fig. 1.3.1. JDL fusion model.

TABLE 1.2.1
Methodology Comparison

Methodology	Advantages	Disadvantages
Parametric	Portability Generality	Need for a priori training process Accuracy variance
Rule-Based	Expressiveness	Accuracy variance Rule sets become unwieldy

Many domains provide very dynamic systems; on any given day there may be topology changes in the templates (to be explained later), patches applied making certain vulnerabilities dissipate or even materialize as a side effect, discovery of new exploits, realization of new attacking strategies, the list goes on. With such frequent changes in the environment, the rule sets can quickly become too complex and unwieldy to remain synchronized with the latest changes. As rules are left out and changed the chances of system accuracy being maintained diminish and the system becomes legacy providing no benefit to the present.

See Table 1.2.1 for an overview of the advantages and disadvantages of parametric and rule-based systems. In summary, we wish to solve the problem of performing real-time detection of complex, multistage, systems in such a fashion that minimizes a priori settings and is sustainable across the breadth and frequency of changes that can occur within the deployment environment.

1.3. Information Fusion Overview

In order to address the problems in data fusion, the US Joint Directors of Laboratories (JDL) developed a five-level Data Fusion Model, shown in Figure 1.3.1 [33]. Level 1 on Object refinement seems to have received the most attention. Level 1 processing functions include: data alignment, association, tracking, and identification. Less mature are Level 2 processing [16] [30], situation assessment, which seeks a higher level of inference above Level 1 processing, and Level 3 processing which performs threat assessment. Threat assessment is an iterative process of fusing the combined activity and capability of enemy forces to infer their intentions and assess the threat that they pose. Level 1 is very often called "low-level" processing, and the others as "high-level" processing.

Higher level fusion problems are generally more difficult than Level 1 because they involve higher dimensionality corresponding to the relationships among entities identified at Level 1. Higher level fusion also concerns modeling behavior of aggregate entities, through the understanding of their individual behaviors and relationships. Some commonly recognized relationships are spatio-temporal relationships, part/whole relationships, organizational relationships, various casual relationships, semantic relationships, similarity relationships, etc.

- Level 0: (Sub-Object Data Association & Estimation)—This deals with signal level data association and characterization.
- Level 1: (Object Refinement)—This deals with track-to-truth and track-to-track association, kinematics estimation and target type and ID prediction.
- Level 2: (Situation Assessment)—This deals with object clustering and relational analysis, to include structure and relations, communications and physical context.

- Level 3: (Impact Assessment or Threat Assessment) —This deals with threat intent estimation, consequence prediction, susceptibility and vulnerability assessment.
- Level 4: (Process Refinement)—This is an adaptive search and processing step.

Level 0 is a special case of Level 1, where entities are signals/features. Level 3 is a special case of Level 2, where relations are cost impact. Level 4 is a special case of Resource Management. Here we will be looking at computational techniques applied in Level 2 and Level 3 data fusion.

1.4. Approaches to High Level Data Fusion

There have been many approaches to performing high level data fusion (L2+) which have been developed, extended, modified, and refined over the years. Many of these approaches which will be discussed shortly have seen success through modification to specific problems, but no single approach has proven to be a single solve-all solution. Every approach has its advantages and disadvantages and the key is to exploit these properties in an optimal fashion for the problem at hand. The various INFERD terms used within this section will be defined and discussed in Section 2 of this paper.

1.4.1. Knowledge Based Expert Systems

Knowledge Based Systems (KBS) are computer systems that contain stored knowledge and solve problems like humans would. KBSs are drawn from the broad discipline of artificial intelligence (AI) where a knowledge base is defined in terms of rules, facts and meta-knowledge. These systems are utilized for combining expert knowledge and sensor information to form a knowledge base which is used for reasoning about the current situation or threat. They are symbolic programs which solve problems by symbol manipulation. Base techniques of knowledge-based systems are rule-based techniques, inductive techniques, hybrid techniques, symbol-manipulation techniques, case based techniques, qualitative techniques, model-based techniques and temporal reasoning techniques.

There are many advantages of using knowledge based expert systems. In expert systems the changes in field of interest are well-tracked and increase the expert's ability and efficiency. In addition to advantages, there are some limitations to knowledge based expert systems. Their knowledge is from a narrow field of interest and they don't know the limits to which it can extend. There can be many exceptions and this can increase the size of knowledge base and eventually the running time of the algorithm. The answers from the expert systems are not always correct, hence the advice has to be analyzed before actually applying it. The expert systems don't have common sense and so all of the self-evident checking has to be predefined. Some examples of applied expert systems for decision support can be found in [3], [42], [1], and [2].

It is typical for expert based knowledge to be required in the classification of observables into detailed domain specific concepts. Otherwise, complex inference processes and a large ontology must be defined, making the solution intractable for time critical applications. The Guidance Templates in INFERD contain *Feature Nodes* that define a set of constraints which (when satisfied) map sensor data into events. This allows INFERD to take advantage of the speed efficiencies of classification in the same manner as KBS for low level fusion, but does not require the definition of complex and interrelated rules needed for higher levels of fusion.

1.4.2. Graph Based Matching Techniques

Graph based matching techniques [10] have been used as a powerful tool for a number of decades, but most notably in the early eighties. Graph based pattern recognition or graph matching is the process of finding a correspondence between the nodes and the edges of two graphs that satisfies some constraints ensuring semantic and syntactic relationships. Graph matching techniques are divided into two broad categories: (1) the exact graph matching method that requires stringent correspondence among the graphs to be matched and (2) the inexact graph matching method, where two graphs can be compared even though they are semantically or topologically different.

Graph matching has been used in high level fusion to abstract complex situations from large amounts of data. The ease of representation of graph patterns and the cognitive advantages of representing situations as a matching between graph based patterns has made the approach increasingly popular with the introduction of new high powered computers. The fundamental problem however, is the theoretical complexity of the graph matching problem. The matching problems mentioned above are all *NP-complete*, with the exception of attributed graph matching in which the nodes are guaranteed to have distinct attributes. In this case the problem becomes polynomial.

To take advantage of the expressiveness and ease of defining graph based patterns, the INFERD Guidance Template has adopted a graph based structure. The structure will be detailed in Section 2, but the similarities stop here in terms of INFERD's fusion process in comparison to graph matching techniques. Because of the theoretical complexities of the matching process, the research team investigated and developed an alternative approach. Remember that the motivation was to provide timely hypothesis generation. These high level hypotheses can very well be linked to graph matching patterns, effectively producing a ranked list of patterns to be matched. This linkage between INFERD and graph matching techniques makes the matching problem more time tractable when there are large numbers of patterns to be matched to a given data graph.

1.4.3. Bayesian Belief Networks

In recent years there has been a surge in use of Bayesian Belief Networks (BBNs) to solve the problems of situation and impact assessment. BBNs have become a popular knowledge inference scheme for probabilistically related evidence and inferences. Their attractiveness lies in the fact that BBNs provide both a sound theoretical framework and a conceptually simple interpretation for representing and manipulating knowledge graphically in a probabilistic domain. BBNs are directed acyclic graphs (DAG), which provide a framework for a structured representation of knowledge about uncertain quantities [12] where nodes and arcs represent conditional probabilistic dependency between variables.

The sound theoretical foundation of BBNs in Bayesian theory can be either an advantage or a disadvantage depending upon the application. In well known environments, BBNs can work very well, however this is not the case in highly dynamic or unknown environments. BBNs are highly dependent upon, and only as good as, the conditional probability tables which are defined. In unknown environments where some or all of these conditional probabilities are not known, or can only be grossly estimated, the accuracy of the BBN will suffer. INFERD does not rely on likelihood functions for this reason. An example of a BBN used in a decision support problem can be seen in [14].

1.4.4. Fuzzy Logic

Fuzzy Logic (FL) is an inferencing methodology that is directed toward vague relationships between evidence and assertions. Fuzzy inference is the process of formulating the mapping from a given input to an output using FL. Because of its multidisciplinary nature, fuzzy inference systems are associated with a number of names, such as fuzzy-rule based systems, fuzzy expert systems, fuzzy modeling, fuzzy associative memory, fuzzy logic controllers, and simply (and ambiguously) fuzzy systems.

Fuzzy logic systems have the advantage of introducing more flexibility into the processing layer to symbolic manipulations or calculations through the definition of *fuzzy membership functions* which can be useful in making decisions in light of information that is imprecise and/or incomplete. Fuzzy logic techniques have become popular to address various processes for multisensor data fusion. Examples include the following: fuzzy membership functions for data association [29] [34], evaluation of alternative hypotheses in multiple hypothesis trackers, fuzzy-logic-based pattern recognition (target identification) [18], and fuzzy inference schemes for sensor resource allocation [23].

A future extension to INFERD could incorporate fuzzy logic into the mapping process of observables into events. Currently observables are mapped to events on a $\{0,1\}$ basis, this could be extended to allow multiple mappings in a fuzzy sense ([0,1]) relaxing INFERD's fusion process to be a multi-hypothesis evaluation system.

1.4.5. Genetic Algorithms

Genetic Algorithms (GAs) are a type of evolutionary algorithm which are the result of studying the natural adaptation of living organisms and are a way of incorporating a similar adaptation into computer systems. They try to mimic environmental factors such as reproduction, random variation, competition, and selection of competing individuals. Genetic algorithms are now widely applied in science and engineering as adaptive algorithms for solving practical search problems particularly suited to multidimensional data where global solutions are found within multiple local minima.

In the information fusion community, GAs are being utilized in many different applications relative to the threat assessment. One of the challenges in a GA based course of action (COA) optimization system is the ability to generate and evaluate thousands of candidate COAs in order to generate the best solution. This consists of two key aspects: the ability to encode the enemy COA into a set that comprises the GA population under evaluation and the ability to quickly evaluate each COA to determine which survives to the next generation. Because the key to success for a GA is evaluating many candidates, it is necessary to be able to abstract the battlefield in order to be able to both encode the situation as a solution string and to be able to rapidly war game each COA in order to evaluate it. Examples of GAs used in high level information fusion problems can be found in the following references: [32], [24], [8], [4], and [5].

As stated in Section 1.4.2, there will be a future need for generation of Template Graphs within certain problem domains. Genetic Algorithms along with Graph Matching could provide a means for creating such templates.

1.4.6. Neural Networks

Artificial Neural Networks (ANNs) are computational systems premised upon the principles of biological neural systems. In general, this means that ANNs are characterized by having many low-level processing units with a high degree of interconnectivity. It is difficult to characterize the field of ANNs succinctly, because the approaches and the results are so diverse. Recently fuzzy logic is been used extensively along with neural networks [20] [9]. Fuzzy logic uses approximate human reasoning in knowledge-based systems while the neural networks aim at pattern recognition, optimization and decision making. A combination of these two technological innovations delivers better results than when used independently.

The advantage of ANNs is that when trained appropriately they produce accurate results for similar problems without the need of any type of parameterization.

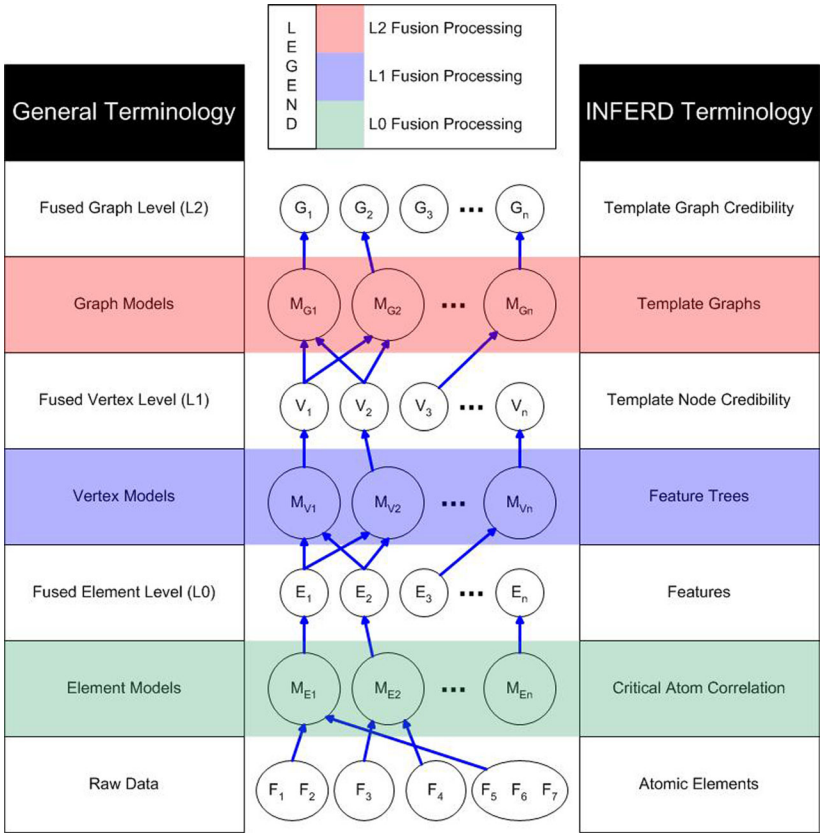


Fig. 2.1.1. General fusion framework.

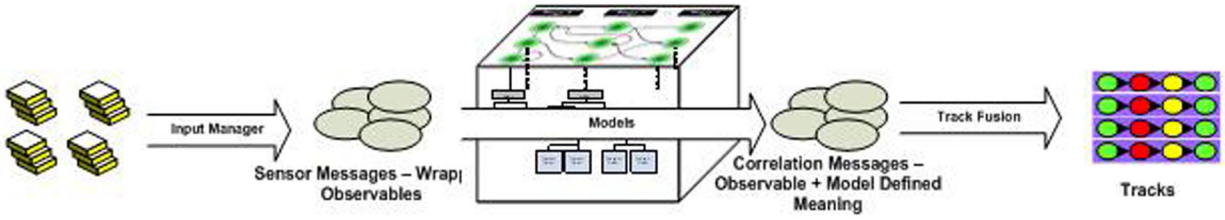


Fig. 2.1.2. INFERD high level information flow diagram.

The disadvantage of neural networks is that their results are contingent upon their level of training. Often in the information fusion community realistic data sets are unavailable or scarce at best. In these situations, neural networks will not be the best solution approach.

Wang and Archer [40] have proposed a neural network based fuzzy set model to support organizational decision making under uncertainty. The model makes use of single back propagation neural network to generate a crisp fuzzy membership function. The authors [41] have used a connectionist approach to multi criteria decision making.

2. THE INFERD ENGINE

2.1. General Fusion Methodology in INFERD

Great care has been taken in the processing structure of the INFERD engine to minimize necessary computation time. In many domains, data rates produced by sensors are computationally intensive to process, so there is not much overhead to spare. The fusion being performed in INFERD is bottom-up in a hierarchical fashion at Levels 0, 1, and 2 according to the JDL model for information fusion [17]. Figure 2.1.1 shows the general terminology and how our system and terminology maps. In the INFERD fusion framework, each subsequent level of fusion feeds off of the previous levels output. This is not a requirement of the JDL model, but suited our system and its network and sensor environment well.

For a summary of the overall fusion processes in INFERD consider the information flow diagram in Figure 2.1.2 as it would apply to the cyber security problem. In this diagram, we can see the flow of basic sensor information, to ultimately, a set of tracks of that information.

The first stage of processing, performed by the *Input Manager*, wraps incoming sensed observables (sensor

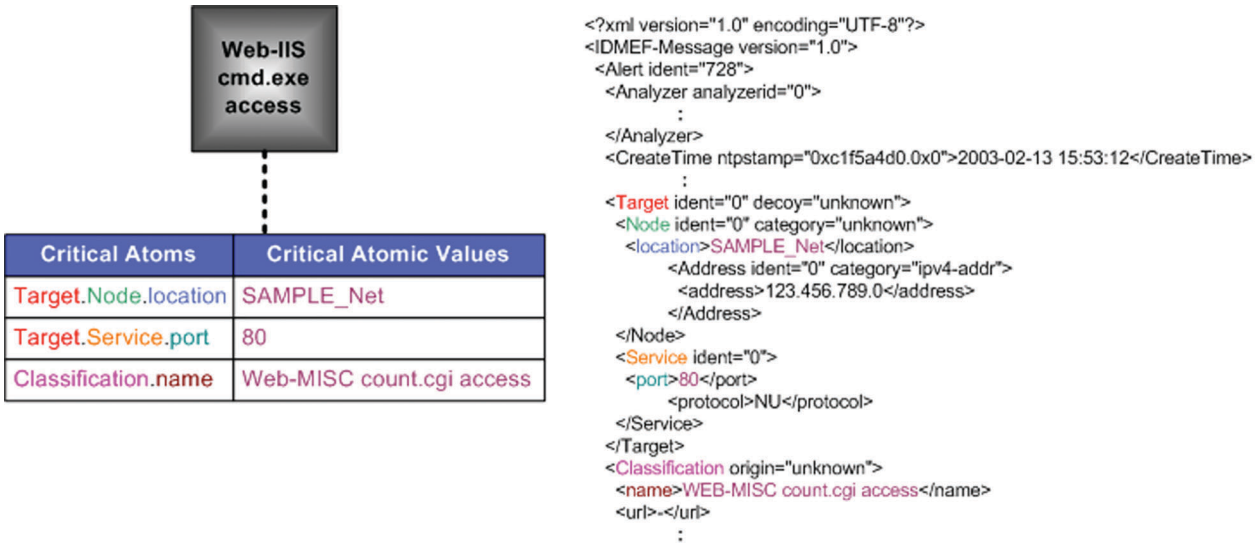


Fig. 2.1.1.1. Example feature node definition.

output) into *Sensor Messages*, a format which is understood by the *Model* and *Track Fusion* Processes. By isolating the fusion processing from the I/O architecturally, INFERD is able to fuse information from sensors of radically different formats and types but still define the *Guidance Templates* in a common language. In the case of cyber security the Input Manager would transform the sensor alerts into an XML object and provide a common referencing method to retrieve values from the object.

The second stage of processing, performed by the Model Fusion Process, assigns model-based meaning to the Sensor Messages. In this stage of processing, *Guidance Templates*, or a priori models, classify the Sensor Message into a higher level event type and expose valid relationships to other previously classified alerts. This newly added information to the Sensor Message, forming a *Correlation Message*, is then understood by and sent to the *Track Fusion Process*. In the cyber security example, this process might reference the target IP address, and signature found within the Sensor Message and classify it as a Scanning Reconnaissance attack on the corporate web server. It would also add the knowledge that this could be a predecessor step for a number of intrusion type attacks on that machine.

The third stage of processing, performed by the *Track Fusion Process*, takes the Correlation Message and fuses it to the existing runtime set of tracks already in existence, possibly resulting in a new track. By fusing piecemeal event steps into unified event tracks, INFERD offers similar advantages to ground target tracking systems, but in a multi-int environment and in new fusion application domains. By analyzing tracks instead of low level sensor events, the analyst is able to prune his search space much more efficiently and have a better understanding of the situation when it is time to make decisions.

Sections 2.1.1 through 2.1.3 will now detail the fusion process in more specificity.

2.1.1. Level 0 Fusion—The Atomic Element

Level 0 fusion is the first processing that occurs once a piece of information is accepted as input into the engine. This piece of information can be of any type such as numeric, text, or file based information. Input to the L0 process is taken in raw data form and then necessary information is extracted by generalized data objects which connect the abstracted data types to the actual sensor message data values. In many instances, more information is taken into the system than is required to analyze what is happening within the desired domain. These desired pieces of information are arranged into Feature Nodes in a tree-structure as understood in basic graph theory. This structure is described later in the fusion discussion.

Once a piece of information (discrete sensor message) is published to a Feature Node, the Critical Atomic Values contained in the message are checked against those specified in the Feature Node in the form of constraint satisfaction. These constraints can take a number of forms such as greater than, less than, equality, string equality, regular expression pattern matching, etc. If all of the defined constraints are satisfied then that Feature Node becomes asserted. The credibility values of Feature Nodes are binary $(0, 1)$ with respect to their assertion state.

In addition to specifying Critical Atoms and Critical Atomic Values, each Feature Node has a specified lifetime associated with it. These lifetimes indicate the maximum amount of time the Feature Node should stay in the asserted state since the time of the last piece of incoming information correlated to it. If Feature Nodes did not de-assert themselves in some fashion, the credibility values of the Template Graphs containing them would never decrease and there would be no temporal

aspect to the INFERD engine. In many domains, it is important that the relative timing of incoming information be considered as to its relevant effects on the system at hand.

Whenever a Feature Node changes state, the parent nodes in the Feature Tree containing that node and subsequently the Template Node specified by that Feature Tree re-calculate their credibility values. This credibility calculation will be discussed in Section 2.1.2 as the L1 fusion process, but it is important to note the bottom up processing which occurs in INFERD. The publish-subscribe service for information input to the system saves a great deal of computation time by not performing the Critical Atomic Value comparisons for the possibly thousands of Feature Nodes which can ignore the alert.

2.1.2. Level 1 Fusion—The Feature Tree

Level 1 fusion processing or Template Node credibility calculation takes over once a Feature Node changes assertion state. The input to the L1 fusion process or Fused Element Level is the Feature Node which has changed assertion state, the Vertex Model is the Feature Tree containing that Feature Node, and the output or Fused Vertex Level of L1 fusion processing is the credibility value or estimated likelihood of occurrence of the Template Nodes who's Feature Tree contains that Feature Node which has changed assertion state. The calculation of L1 credibility values is inherent in the structure of the Feature Tree and the values of the Relation Nodes within that tree.

Every Relation Node specifies a function determining how its children relate to each other. We use Yager's Generalized Ordered Weighted Average (GOWA) function as a means of calculating the relation [43, 44, 45]. Assume $\{A_1, A_2, \ldots, A_n\}$ are n criteria of concern in this multi-criteria decision problem. These are the criteria described in the atomic elements above. Let us further assume that the values $a_1, a_2, \ldots, a_n$ represent credibilities associated with the above set A of n elements. We can then construct a function $F(a_1, a_2, \ldots, a_n)$ that will be used to aggregate its children at the relation node. Yager describes many properties of such a function. His OWA operators are designed by introducing two vectors B and W. Let B be an ordering vector that "rearranges" the credibilities $a_1, a_2, \ldots, a_n$ in descending order. Let W be a weighting vector such that $\sum_{i \in W} w_i = 1$, $w_i \geq 0$. In vector form, the OWA operator is expressed as $F(a_1, a_2, \ldots, a_n) = W^T B$. Numerical examples are shown in Yager's referenced papers. The theory is carried out further to describe a concept known as "attitudinal character" that describes the level of "ANDness and ORness" that the W vector takes on. The attitudinal character is described by the following: $AC(W) = \sum_{j=1}^{n} w_j (n-j)/(n-1)$. For example, if $W = [1, 0, 0, \ldots, 0]$, then $AC(W) = 1$ thus saying that we have the greatest possible "ORness" since this would give us a maximization function. This is true since we are multiplying W and B where only the first element of B would be considered (since $w_1 = 1$). Note the first element of B is $\max(a_i)$. Similarly, we have maximum "ANDness" when $W = [0, 0, \ldots, 0, 1]$; $AC(W) = 0$. Finally, we simply compute the average value when $W = [1/n, 1/n, \ldots, 1/n]$; thus $AC(W) = 1/2$. Such a general function has unlimited possibilities and can be applied to any domain using aggregation functions.

The Feature Tree used in INFERD consists of a Template Node at the "top" of the tree. Below it may be a series of child nodes. Each of these child nodes may be a series of child nodes to them (or grandchild nodes to the Template Node). The above GOWA function is used to describe the relationships between the child nodes and their respective parents. To calculate the values of the Template Node (or parent node as it is known in graph theory), INFERD begins with understanding of the child nodes at the very bottom of the tree, then it works its way upward. The "bottom-most" nodes of the tree are the pieces of information taken in via L0 fusion discussed in the above section. Once these binary values are obtained, we can apply the OWA function to obtain the probabilistic value of their parent nodes. This process continues up the tree structure until a value is figured for the Template Node and further used in the L2 fusion steps.

We will now introduce an example of a system that could be analyzed using INFERD. We will continue using this example throughout the paper. Airport Security is an increasingly important issue in today's society. There are many measures taken to prevent unsafe situations. We will present a somewhat simplified way to answer the question: Is this passenger of any danger to their fellow passengers? This question will be answered probabilistically through determining its Credibility Factor (discussed later in the paper). There are many different considerations in answering this question; to describe the Feature Tree, we will look into the verification of a passenger's identity. The node in the Template Graph is called "ID Verification." We will see later how this becomes a part of the Template Graph and how it interacts with other nodes. For now, let us look at its underlying Feature Tree such that we can obtain credibility for the node. For our understanding, let a higher credibility indicate a higher chance of this passenger being an immediate danger. Figure 2.1.2.1 provides a visual of the Feature Tree.

For simplified understanding of INFERD, we will consider only three measures taken to verify a passenger's identification prior to their boarding of a commercial aircraft. Applying the GOWA function on our Relation Nodes, we choose $W = [1/n, 1/n, \ldots, 1/n]$ to be our measure, hence we will take a weighted average of its immediate children, Risk Assessment, Photo ID, and Biometrics. When a passenger books a flight, they may be asked a series of personal questions that will lead to an assessment of their risk. When this is completed, the airline representative will then assess the risk based on

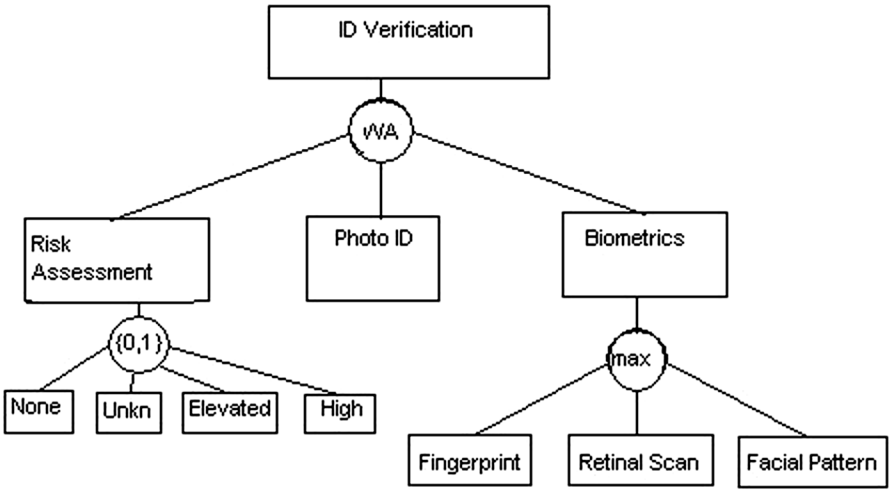


Fig. 2.1.2.1. The feature tree underlying the ID verification node.

the answers to the questions. To illustrate that any function may be used in this level of INFERD, we will introduce a binary function such that each of the four levels of risk assessment will be given a value in $\{0,1\}$ where 0 = the level of risk was not given to the passenger and 1 = the level of risk was given to the passenger. We will then take a weighted average of the binary values against the weights $[0.00, 0.33, 0.67, 1.00]$ for no risk, unknown risk, elevated risk, and high risk respectively. Next, when a passenger claims their boarding passes at the airport, they are asked to show their photo ID. If that ID matches all known information about the passenger, we give a 0 value to that node; conversely, if there is a discrepancy we will assign a value of 1. Finally, there is a system being worked on and nearly in place in most major airports called CAPPS II (Computer Assisted Passenger Prescreening System). CAPPS II takes biometric information about the passenger and attempts to verify their identity. The system will test fingerprints, retinal scans, and facial patterns of passengers. In our model, we will assign a 0 value if there is no problem identifying the passenger positively. However, if there is an issue with these, we will assign the value 1. Under the biometrics node we will take the maximum using the GOWA function by setting $W = [1, 0, 0, \ldots, 0]$.

Let us assume that the passenger being screened when purchasing their tickets was given a risk assessment of "unknown." When they arrived at the airport, their photo ID matched up. However, when CAPPS II was used there was an identification discrepancy with the retinal scan and the facial pattern (the fingerprint appeared to be correct). The node for Risk Assessment would be given a value $0*0 + 1*0.33 + 0*0.67 + 0*1 = 0.33$. The photo ID node would have a value of 0. The biometrics node has a value of $\max\{0,1,1\} = 1$. Hence, we take the average to obtain the credibility of the Template Node ($j = 1$), ID Verification. $c_1 = (0.33 + 0 + 1)/3 = 0.443$. We will use this value going forward.

2.1.3. Level 2 Fusion—Template Credibility Calculation

Level 2 fusion or Situation Refinement is currently the highest level being implemented in the INFERD engine. The input or Fused Vertex Level to L2 are the credibility values of the Template Nodes, the model is the given template and the output or Fused Graph Level is an overall credibility value for that template. It is these credibility values coupled with the ranking of the templates that provides the system analyst with a situational estimation of their system's current environmental status.

Credibility Values exist for each node in the Template Graph (Feature Nodes and Template Nodes) and the Template Graph itself. While the methods of calculation of these values vary, the meanings of the values remain consistent. A credibility value is simply a likelihood of occurrence that INFERD produces. For Feature Nodes, this value is in $\{0,1\}$ because either the observable captured by that node was input to the system or it was not. For Template Nodes which can represent events, objects, or abstract concepts the value is in $[0,1]$ because this is a much more fuzzy process. The same argument is made for Template Graph credibility calculation as well.

The INFERD engine has imbedded into it, by the system user, templates specific to the given system being studied suggesting the way it works within its environment. Each Template Node may have an underlying Feature Tree that gives INFERD its credibility via L0 and L1 fusion described earlier. The functions applied to the children in the Feature Tree are chosen by the user, the following figure shows maximum and weighted average. The Template Nodes are then linked to each other as deemed reasonable to make up the Template Graph. See Figure 2.1.3.1 for an illustration.

These Template Nodes connect to one another to form a Template Graph. There are three types of nodes that can exist in the graph as defined by their links to other nodes. Extrinsic Nodes are those that have no

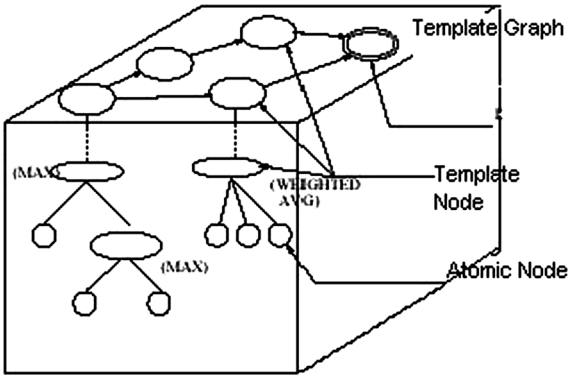


Fig. 2.1.3.1. Illustration of L1 and L2 fusion within INFERD.

precursor Template Nodes. Their credibility value or likelihood of occurrence is based solely on the Feature Nodes contained within its Feature Tree. Intrinsic Nodes are those that have one or more precursor nodes, and also are not reported on at any level by any sub-structure. These nodes can only be possible if triggered by another node connecting to it in the Template Graph; there is no underlying Feature Tree. Bi-trinsic Nodes are those that are reported on at some level by L0 fusion and also have precursor nodes. The credibility value of nodes of this class can leverage data from its Feature Trees along with its precursor nodes.

For example, let us say we have a cyber network alert system [35] being monitored by INFERD; we may have Attack Templates imbedded into the engine. Each of the Template Nodes would be some sub-situation that could imply a possible attack on ones network. Hence, underlying these Template Nodes would be the Feature Tree including the steps possibly leading up to this part of an attack happening. Note that there can, and most likely will be, many more than one single Template Graph being analyzed by INFERD at any given time.

Continuing our Airport Security example, we design a Template Graph containing seven nodes that each has an important contribution to understanding the credibility of a passenger's safety. INFERD stores many Template Graphs and analyzes them at the same time. Hence, in this example, individual passengers would have their own Template Graph. However, in many other domains there may not be a consistency among Template Graphs; there could be many with different factors. Figure 2.1.3.2 illustrates our Template Graph with node numberings in parentheses.

In our example, we will consider ID Verification among other actions taken by the passenger, most of which are understood in context. When in an airport, one is not allowed under law to speak of "terrorism," "bombs," "guns," etc. Hence, we include node 7 as "forbidden" words. Underlying each of these Template Nodes, there may be a Feature Tree giving a credibility factor denoted by c_j where $j = 1,\ldots,7$. Notice how some Template Nodes also have influences from other Template Nodes in the Template Graph. From above, we have $c_1 = 0.443$; let $c = [0.443, 0.55, 1, 0.01, 0.4, 0.1, 0.15]$. We will work with this Template Graph in the next sections.

Now we describe two approaches that can be used to determine the credibility factor of the Template Graph in the INFERD engine. The first method described will be a parametric approach with the advantages and disadvantages discussed above. The second approach will be the Entropy approach used to combat many of the drawbacks of the parameterized approach.

2.1.3.1. The Parameterized Approach for Credibility Factor

Our first L2 algorithm uses concepts in Statistical Mechanics. During the late 1800s, M. L. Boltzmann and J. W. Gibbs studied in the field of thermodynamics and pioneered what we now know as statistical mechanics. While thermodynamics (in the classical sense) deals with a single system called a macrostate, statistical mechanics studies the sub-components of this system, called microstates. Statistical mechanics is the application of probability theory to the field of mechanics for large populations of particles with respect to their motion subject to forces. The greatest benefit of such a methodology from a physics point of view is that statistical mechanics contains the ability to make macroscopic predictions based on microscopic properties. This ability lends itself directly to a Data Fusion system since raw data enters the system as microscopic properties and the desired result of Situation Awareness is a macroscopic prediction based on the raw data.

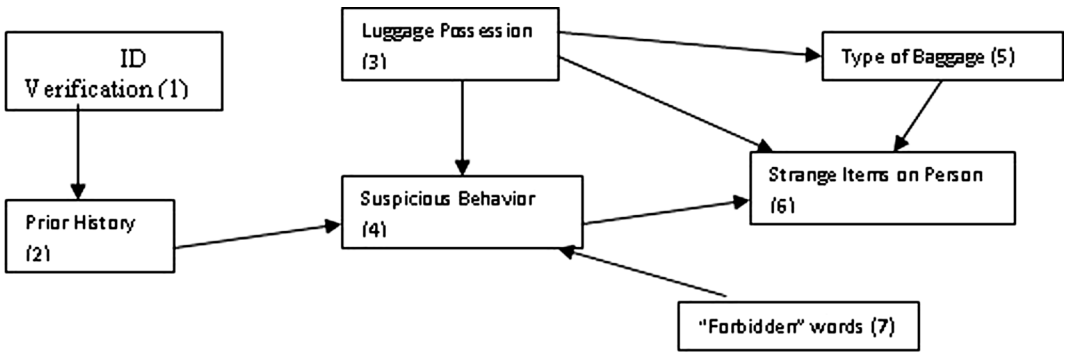


Fig. 2.1.3.2. Example of Template Graph.

One of the very first applications of statistical mechanics to optimization was in the field of Simulated Annealing. Simulated annealing is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems [27]. The concept is based on the manner in which liquids freeze or metals recrystalize in the process of annealing. In an annealing process a melted material, initially at high temperature and disordered, is slowly cooled so that the system at any time is approximately at thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a frozen ground state. The original Metropolis scheme was that an initial state of a thermodynamic system was chosen at energy E and temperature T, then by holding T constant the initial configuration is perturbed and the change in energy dE is computed. If the change in energy is negative the new configuration is accepted. If the change in energy is positive it is accepted with a probability given by the Boltzmann factor $e^{-dE/T}$. This process is repeated a sufficient number of times to give good sampling statistics for the current temperature, and then the temperature is decremented and the entire process repeated until a frozen state is achieved at $T = 0$. By analogy the generalization of this Monte Carlo approach to combinatorial problems is straight forward [21]. The current state of the thermodynamic system is analogous to the current solution to the combinatorial problem—the energy equation for the thermodynamic system is analogous to the objective function, and the ground state is analogous to the global minimum. Hence, this notion of simulated annealing can be used in optimization problems that are NP-hard as a brilliant heuristic approach. The basic components of simulated annealing are in statistical mechanics, thus showing a strong tie between the fields of statistical mechanics and optimization. We therefore recommend its use for our purposes in data fusion and as a heuristic for situation state estimation. Claude Shannon found deep links between information theory and thermodynamics. Following the same reasoning a possible link can be drawn between the probability of occurrence of the activity of track of hacker behavior in a noisy environment and the heating and cooling of a metal to a steady state. Thus we investigate this approach as applied to the problem of computer network security.

One of the more important results discovered by Gibbs and Boltzmann describes the probability of a microstate being within a certain energy state. We denote the energy state as E_s; under the assumption of the system at hand being independent of other systems, we can write the probability as follows:

$$P(E_s) = \frac{e^{-E_s/T}}{Z(T)}$$

where T denotes the temperature of the system and $Z(T)$ is a partition function which normalizes probabilities across all states such that $\sum_{s\in S} P(E_s) = 1$.

This concept has been used in various applications in Information Theory [31], Optimization [21], and Decision Theory [13]. The application of the Gibbs-Boltzmann Equation in INFERD begins with defining the Template Graph as the system's macrostate. It's sub-components (Template Nodes) represent the microstates. Let $G(N,A)$ be the macrostate (Template Graph) where N is the set of Template Nodes and A is the set of arcs connecting the nodes. Each node $j \in N$ has a probability of belonging to one of four possible energy states:

$$E_j^H = \text{High}$$
$$E_j^M = \text{Medium}$$
$$E_j^L = \text{Low}$$
$$E_j^0 = \text{Insignificant(Zero)}.$$

Given the discrete nature of the energy states, we must introduce thresholds to determine to which energy state each Template Node j belongs. Hence, we are introducing a parameter that may be set by the INFERD user as they see fit within their system. Let TH^H, TH^M, and TH^L denote the threshold values between the high, medium and low energies respectively. Note that $TH^L < TH^M < TH^H$, and $0 \le TH^i \le 1$ for all i in $\{L,M,H\}$. Let c_j denote the credibility of node j coming from the Feature Tree in L1 fusion described above. We can determine the credibility (or probability of a Template Node occurring), P_j, using equation (2.1.3.1.1):

$$P_j = \begin{cases} P(E_j^H) = \dfrac{e^{-\alpha^{-2}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}, & c_j \in [TH^H, 1] \\ P(E_j^M) = \dfrac{e^{-\alpha^{-1}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}, & c_j \in [TH^M, TH^H) \\ P(E_j^L) = \dfrac{e^{-\alpha^{-0}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}, & c_j \in [TH^L, TH^M) \\ P(E_j^0) = \dfrac{e^{-\alpha^{1}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}, & c_j \in [0, TH^L) \end{cases} . \quad (2.1.3.1.1)$$

These thresholds and the constant α ($\alpha \ge 1$), will allow for a parametric approach in determining the energy level of each Template Node j. A higher value of α results in more emphasis being placed on the higher energy states and vice versa. The assigning of value α can be attributed to many different reasons specific to each user; however we suggest that if the user believes their L0 sensors (information detection sensors) are highly reliable, they may opt for a higher α value. In contrast, if the user has less confidence in their sensors detecting incoming information, they may wish to use an α value closer to 1.

We define our partition function as $Z(T) = |N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}$ such that the sum of the probabilities over all energy states is equal to 1, hence meeting the

basic axioms of probability:

$$\sum_{s\in S} P(E_s) = \sum_{j=1}^{|N|}(P(E_j^H) + P(E_j^M) + P(E_j^L) + P(E_j^0)) = |N|(P(E_j^H) + P(E_j^M) + P(E_j^L) + P(E_j^0))$$
$$= |N|\left(\frac{e^{-\alpha^{-2}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}} + \frac{e^{-\alpha^{-1}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}} + \frac{e^{-\alpha^{0}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}} + \frac{e^{-\alpha^{1}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}\right) = 1.$$

Now that we have the probabilities of each Template Node in $G(N,A)$ being in a certain energy state, we use them to obtain an overall credibility factor (CF) for the entire Template Graph. The probability of the state of a node j that has other nodes directed to it (N_j) will be affected by its neighboring nodes as long as the last occurrence of the two events depicting the node (at times r_j and r_h) are within a desirable time frame as set by the user, denoted t_{jh}. It is necessary to define a new set of probabilities for each Template Node that not only take into account its own state probability, but also the current states of the Template Nodes directed to it. Equation (2.1.3.1.2) defines Q_j as these desired probabilities:

$$Q_j = \lambda_j^0 P_j + \sum_{h\in N_j, |r_j - r_h| \le t_{jh}} \lambda_j^h Q_h \qquad \forall \quad j \in N. \tag{2.1.3.1.2}$$

$G(N,A)$ cannot contain any directed cycles. In particular, there will always be at least one sequence for obtaining the revised probabilities, such that no Q_j that depends on another is calculated without the proper adjustment.

Now that we have obtained probability values for each node of the Template Graph considering the topology of $G(N,A)$, we can introduce the overall Credibility Factor (CF) as seen in equation (2.1.3.1.3):

$$CF = \frac{\sum_{j=1}^{|N|} Q_j}{\left(\frac{e^{-\alpha^{-2}}}{\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}\right)}. \tag{2.1.3.1.3}$$

The denominator of equation (2.1.3.1.3) simply normalizes the overall Credibility Factor so that when the probabilities of all of the Template Nodes are equal and in the high-energy state, then:

$$CF = \frac{\sum_{j=1}^{|N|} Q_j}{\left(\frac{e^{-\alpha^{-2}}}{\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}\right)} = \frac{\sum_{j=1}^{|N|} \frac{e^{-\alpha^{-2}}}{|N|\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}}{\left(\frac{e^{-\alpha^{-2}}}{\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}\right)} = \frac{\left(\frac{e^{-\alpha^{-2}}}{\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}\right)}{\left(\frac{e^{-\alpha^{-2}}}{\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}\right)} = 1.$$

The parameters λ are the constraints used to obtain a weighted sum of the state probabilities, such that:

$$\lambda_j^0 + \sum_{h\in N_j} \lambda_j^h = 1, \qquad \forall \quad j \in N, \quad \text{with}$$
$$\lambda_j^0 \ge 0, \qquad \lambda_j^h \ge 0 \qquad \forall \quad h \in N_j, \quad \text{and} \quad \forall \quad j \in N.$$

These λ values represent the importance of connecting nodes as desired by the user. For instance, in some application domains, it can be such that the influence of the Feature Trees with respect to Template Nodes be weighted heavily, and the correlation influence of connected nodes be only marginally considered. In this case, λ_j^0 can be set close to 1, and the λ_j^h values closer to 0. Note that each individual λ_j^h value does not necessarily have to be equal; one can place different weights on each node directed at the node in question.

It is important to note that in order for a consistent calculation of the Q_j values to be possible,

Let us continue our Airport Security example referring back to Figure 2.1.3.2 showing our Template Graph. Recall $c = [0.443, 0.55, 1, 0.01, 0.4, 0.1, 0.15]$. We will set our parameter $\alpha = 2$ and our set of thresholds as $TH^i = [0.25, 0.5, 0.75]$ for low, medium and high respectively. Then using equation (2.1.3.1.1), we can see the probability of being in high, medium, low, and insignificant energy states are $[0.059, 0.046, 0.028, 0.010]$ respectively. Now we can find the Q_j values given (2.1.3.1.2) and the parameters as follows:

$$\lambda_j^0 = \left\{ \begin{array}{ll} 1, & \text{if} \quad N_j = \emptyset \\ 0.5, & \text{o.w.} \end{array} \right\}$$

$$\lambda_j^h = \frac{0.5}{|N_j|} \qquad \forall \quad h \in \{N_j : N_j \neq \emptyset\}, \quad \text{and} \quad \forall \quad j \in N.$$

We can see from the values given by the c vector, nodes 4, 6, and 7 have insignificant energy lev-

els; nodes 1 and 5 have low energy; node 2 has medium energy; and node 3 has a high energy level. Hence, we use the P_j values calculated above along with the P_j values for each node of the Template Graph with the appropriate weighting to determine the Q_j value for each node j in N. We get $Q = [0.028, 0.037, 0.059, 0.024, 0.044, 0.021, 0.01]$. Then we compute the credibility factor as:

$$CF = \frac{\sum_{j=1}^{|N|} Q_j}{\left(\frac{e^{-\alpha^{-2}}}{\sum_{i=0}^{3} e^{-\alpha^{-(i-1)}}}\right)}$$

$$= \frac{0.028 + 0.037 + 0.059 + 0.024 + 0.044 + 0.021 + 0.01}{0.412}$$

$$= \frac{0.223}{0.412} = .5413 = 54.13\%.$$

Hence, in our example, under the statistical mechanics with parameterization methodology, we obtain a Credibility Factor of 54.13%, suggesting that this given passenger is about 54% probable to be a danger to others in the airport or on the aircraft. It is left to the user's discretion as to what is a large enough credibility in their given system in order to react accordingly.

2.1.3.2. The Entropy Approach for Credibility Factor

To liberate our system from the parametric and rule-based deficiencies listed in Table 1.2.1, we have implemented a novel approach of using Entropy, or a measure of randomness, to calculate the credibility values for our Template Graphs. By determining the inherent level of randomness in a template, and relating it to the maximum amount of randomness possible, we can derive meaning about how likely (credible) that particular template graph is taking place.

The theory of statistical mechanics is governed primarily through the second law of thermodynamics, better known as entropy. Entropy was first used as a measure within the study of thermodynamics, but has since been shown to be valuable in many other areas including psychodynamics, thermoeconomics and information theory. Information theory is useful in many disciplines but is most basically defined as a means to measure the amount of data that can be stored in a communication type medium. Claude Shannon, in 1948, composed a famous work [31] wherein he began to understand the transmission of information through a noisy channel. His fundamental results include the "source coding theorem" which states that the average number of bits of information required to represent the result of an uncertain event is given by entropy. Shannon's "noisy channel coding theorem" suggests that *reliable* communication is possible over noisy channels provided that the rate of communication is below a certain threshold. INFERD is a fusion system where a large amount of information is input, some of which is valuable and some of which is not. This non-valuable information can be considered noise in Information Theoretic terms. Since entropy measures amounts of valuable information throughout a channeling system, it seems appropriate to use such a measure for Situational Assessment within INFERD.

The study of entropy has evolved greatly throughout the years. It has been shown that there are many types of entropy that can be used in many domains. Tsallis [36, 37] introduces a generalized entropy function based on a parameter q.

$$H_q = k\frac{1 - \sum_{i=1}^{W} p_i^q}{q-1}, \qquad \left(\sum_{i=1}^{W} p_i = 1,\ k > 0\right).$$

The question arises as to what the value of q should be in any given domain. In [37], Tsallis discusses three optimization methods that can be used to find the optimal q. In [36], he discusses how in many optimization algorithms and information theory domains, $q \to 1$. He later suggests that while considering a Gaussian distribution, $q \to 1$ thus is the case for many natural phenomenon. Hence, we use the above Tsallis General Entropy Function with $q \to 1$. This gives us Shannon's Entropy Function as seen below.

Claude Shannon studied the discovery of statistical knowledge about a source by use of proper encoding of the information and defined entropy in cooperation with Boltzmann's famous H-Theorem as shown in equation (2.1.3.2.1), where H is entropy, p_i is the probability of being in state i and K is a constant (Boltzmann's constant in thermodynamics) [31].

$$H = -K\sum_{i=1}^{n} p_i \log_2 p_i. \qquad (2.1.3.2.1)$$

Shannon's application of entropy to information theory allows one to find the total amount of randomness embedded in a state-system process. In doing so, there must be an existing alphabet with known probabilities of symbols. Consider the example where we have an alphabet consisting of four symbols with the following probabilities (1/2 1/4 1/8 1/8). Using equation (2.1.3.2.1) we get $H = (1/2)\log_2 2 + (1/4)\log_2 4 + (1/8)\log_2 8 + (1/8)\log_2 8 = 1.75$ bits/symbol. Next consider the case where the probabilities of each symbol are at equality (1/4 1/4 1/4 1/4). Using equation (2.1.3.2.1) we get $H = 2.0$ bits/symbol. Next consider the case where we have the following probabilities (0 0 0 1). Equation (2.1.3.2.1) gives $H = 0.0$ bits/symbol. Note that as the probabilities of each symbol move to equality, the entropy moves to a maximum. This corresponds intuitively with the idea of randomness in a system—as each symbol in the alphabet becomes equally likely to occur, the symbols in the words constructed from that alphabet become less predictable. Also note that as the number of symbols in the alphabet increases, so does the randomness. This follows intuitively as well—if there are more symbols to choose from, predictability becomes more difficult.

Our system does not use alphabets and alphanumeric symbols as discussed in Shannon's paper, but our application in INFERD is in line with the requirements of entropy as defined by Shannon. Here, the "system" is our Template Graph, and the "symbols" are the Template Nodes within the Template Graph. We measure the entropy of the Template Graph by growing or shrinking the Total State Space defined below according to the credibilities of the Template Nodes and keeping the probabilities of each state within each sub-space at equality. By altering the size of the Total State Space to determine entropy as opposed to altering probabilities of states within the space, we develop a monotonically increasing H function with respect to the credibilities of the Template Nodes (c_j).

Before we describe the entropy method for calculating the Credibility Factor we must take into account the topology of the Template Graph. This is a similar procedure to the "Q-function" used in the Statistical Mechanics Methodology in Section 2.1.3.1. In fact, the only parameters used in the Entropy Approach are the same λ values as defined in the previous section. We will use the following equation to determine the new c_j* (the c_j values that take the directions of the Template Graph edges to nodes into account) values to be used in the entropy calculation.

$$c_j* = \lambda_j^0 c_j + \sum_{h\in N_j, |r_j - r_h| \le t_{jh}} \lambda_j^h c_h * \qquad \forall \quad j \in N$$

$$\lambda_j^0 + \sum_{h\in N_j} \lambda_j^h = 1, \qquad \forall \quad j \in N, \quad \text{with}$$

$$\lambda_j^0 \ge 0, \qquad \lambda_j^h \ge 0 \qquad \forall \quad h \in N_j, \quad \text{and} \quad \forall \quad j \in N.$$

We have a Template Graph $G(N,A)$ with a node set N and an arc set A, where the jth node has a normalized credibility factor value of c_j. We seek a normalized scalar aggregation function that approaches zero when all node credibilities tend to zero and approaches unity when all node credibilities tend to unity, and does not require us to take account of the arc set A (which would require an extensive parameterization of the aggregation function.)

Since the only data we intend to use in the aggregation function are the normalized credibility factors c_j, which can be interpreted as individual probabilities of their corresponding Template Nodes being "true," we are motivated to consider the Shannon entropy function as a convenient starting point for building our aggregation function. Shannon entropy is very simple to calculate under the assumption that a system has K equiprobable states, and is given by $H = \log K$ in this case. (The base of the logarithm is immaterial, as changing bases only induces a constant factor multiplying H.)

Thus we consider a system having equiprobable states, where the overall number of states is a decreasing function of the variable $x = \sum_{j=1}^{|N|} c_j$, i.e., the more certain we are of the truth of our composite set of Template Nodes (such that $x \to |N|$), the lower the number of states and hence the smaller the value of H; conversely, as the truth probabilities approach zero ($x \to 0$), the larger the number of states and the larger the value of H.

A simple function for the number of states K that satisfies these properties is

$$K = |N| - \sum_{j=1}^{|N|} c_j + 1.$$

In the two extreme cases, we have

$$H_{\min} = H(x = |N|) = \log K|_{c_j \equiv 1 \ \forall_j} = \log(1) = 0$$

$$H_{max} = H(x = 0) = \log K|_{c_j \equiv 0 \ \forall_j} = \log(|N| + 1).$$

For all values $0 < x < |N|$, we have $\log(|N| + 1) > H(x) > 0$.

Our desired credibility factor $CF(x)$ for the Template Graph should range monotonically between zero and unity as x ranges between its maximum and minimum values, respectively. The simplest function satisfying these properties is similar to the work presented by Pierce and John [28]:

$$CF(x) = \frac{H_{\max} - H(x)}{H_{\max} - H_{\min}} = \frac{\log(|N| + 1) - \log\left(|N| - \sum_{j=1}^{|N|} c_j + 1\right)}{\log(|N| + 1)}.$$

Now let us continue our ongoing example and consider the Template Graph with the same values given before: $c = [0.443, 0.55, 1, 0.01, 0.4, 0.1, 0.15]$. Here we illustrate the entropy approach via example. First, we must account for which nodes are pointed at which (the topology of the Template Graph). We will define our parameter λ just as is done in the prior example in Statistical Mechanics. Using the same routine, we obtain $c_j* = [0.433, 0.492, 1, 0.288, 0.7, 0.285, 0.15]$. We can now use the above equation to find the credibility factor (CF).

$$CF(x) = \frac{H_{\max} - H(x)}{H_{\max} - H_{\min}} = \frac{\log(|N| + 1) - \log\left(|N| - \sum_{j=1}^{|N|} c_j + 1\right)}{\log(|N| + 1)}$$
$$= \frac{\log(8) - \log(7 - (0.433 + 0.492 + 1 + 0.288 + 0.7 + 0.285 + 0.15) + 1)}{\log(8)} = 0.261.$$

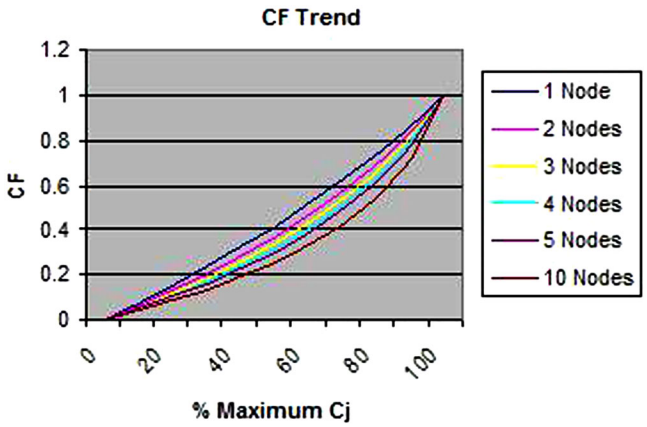


Fig. 2.1.3.2.2. *CF* Trends as node count increases.

Hence we say there is a 26.1% chance that this particular passenger is a danger to those around him.

As mentioned before, *CF* is the credibility value of the Template Graph and represents the likelihood that the given scenario is taking place. This value is used to rank the Template Graphs and is a simple indicator to the analyst helping them in their decision process of which situations to look into further.

A question can be raised to why a ten node Template Graph is not ranked as credible as a 1 node Template Graph when the sum of the credibilities of the nodes contained within them are at the same percentage level with respect to the maximum $\sum_{j \in N} c_j *$ (refer to Figure 2.1.3.2.2). Recall that as this value increases and decreases we determine the entropy for the graph by decreasing and increasing the size of Ω, respectively. Each state in this state space represents a piece of knowledge defining the scenario that has not been detected in the stream. Templates Graphs with more nodes have more of these states when at the same $\sum_{j \in N} c_j *$ level, which makes intuitive sense because we must detect many more occurrences in the system to be of definite certainty that it has taken place.

2.1.3.3. Other Credibility Factors

The above stated credibility factor calculations determine the reliability of information. However, we believe that although this is a very valuable measure, it is not all inclusive in terms of aiding the system user to make a complete decision. There should be more measures allowing a user to be more well informed of the current situation.

We provide two examples of possible measures that can be defined and embedded into a future version of INFERD. Let's consider the cyber domain as an example. One helpful measure could be "Depth." Cyber attacks are usually accomplished in a progression toward an end goal. This progression is understood, and thus a measure can be defined in order to determine how far into an attack a hacker may be at a certain time. This helps explain the current situation (L2 fusion) as well as understand possible immediate ramifications of a continued attack (L3 fusion). Another helpful measure could be "Breadth" of an attack. Breadth would help understand the entire scope of an attack, thus providing the user with information regarding how many possibilities an attacker would have in the near future.

These credibility measures among others can be very helpful to a user in terms of both situational awareness and impact assessment and will be further explored in future versions of INFERD.

3. CONCLUSIONS AND FURTHER WORK

In this paper we have described our INFERD system in a general sense as it can be applied to various domains depending on the needs of the consumer. We offer a system that is flexible in that a user can adjust the functions used at L1 and L2 fusion as well as input their own scenarios as Template Graphs in order to meet their needs. We describe the JDL definitions used for information fusion and show how INFERD incorporates those steps into its analysis of the system at hand. We offer two opposing viewpoints at the second level of fusion (L2) along with the advantages and disadvantages of each. The Entropy approach we discuss is a new and improved approach with direct ties into information theory as pioneered by Shannon [31].

To initially test INFERD and its fusion capabilities related to the cyber warfare domain, AFRL tasked Skaion Corporation with the job of generating a number of synthetic cyber attack traffic data sets labeled "Blind Tests." These data sets are actual packet and IDS alert information generated from attacks that were run on a virtual computer network with common data set components such as noise injected in. In this first test, INFERD was able to handle 86.4 million alerts over a 24 hour period. These data processing rates are highly above even large computer networks allowing us to claim real-time performance. Future tests will be performed against ground truth information to assess the "quality" of the generated hypotheses and the sensitivity of the generated hypotheses as a function of the parameters of the algorithm.

Advancements in the fusion process itself have been considered and proposed as research for a future phase of the project. Being able to determine credibilities in a given system is just a first step in the process of being able to successfully use that information to perform a desired task with ones system. In the future, there will be work done to make INFERD a self-acting, as well as a self-learning, system. An upcoming stage of our research will be to determine how to make INFERD a self-acting engine for various applications, hence creating a self-governing machine.

Currently, the user of the system must enter the Template Graphs to be analyzed by INFERD. For many application domains, it may be necessary to generate

thousands or tens of thousands of these templates in order to appropriately analyze the system. Hence, it would be highly useful to create some sort of automated Template Generation technique. The next stage of our research will be to find a method to generate desirable templates to be inserted into INFERD for analyzing.

ACKNOWLEDGMENTS

This work has been performed under joint funding from the Air Force Research Lab in Rome, NY (AFRL-Rome) and the Advanced Research & Development Activity (ARDA).

REFERENCES

[1] D. Ballard and L. Owsley
Artificial intelligence in the helicopter cockpit of the future.
In *IEEE/AIAA 10th Digital Avionics Systems Conference*, 1991, 125–130.

[2] D. Ballard and L. Rippy
A knowledge-based decision aid for enhanced situational awareness.
In *IEEE/AIAA 13th Digital Avionics Systems Conference*, 1994, 340–347.

[3] S. B. Banks and C. S. Lizza
Pilot's associate: A cooperative, knowledge-based system application.
IEEE Intelligent Systems and Their Applications, **6**, 3 (1991), 18–29.

[4] E. Bengoetxea, P. Larranaga, I. Bloch and A. Perchant
Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems.
In *EMMCVPR '01: Proceedings of the Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, London, UK, 2001, 454–468.

[5] E. Bengoetxea, P. Larranaga, I. Bloch, A. Perchant and C. Boeres
Inexact graph matching using learning and simulation of Bayesian networks.
In Proceedings of CaNew workshop, ECAI, 2000.

[6] C. Bishop
Neural Networks for Pattern Recognition.
New York: Oxford University Press, 1995.

[7] P. Bradley, U. Fayyad and O. Mangasarian
Mathematical programming for data mining: Formulations and challenges.
INFORMS Journal on Computing, 1999.

[8] A. L. Buczak and R. E. Uhrig
Hybrid fuzzy-genetic technique for multisensor fusion.
Information Sciences, **93**, 3–4 (1996), 265–281.

[9] L. Chin-Teng and C. S. G. Lee
Reinforcement structure/parameter learning for neural-network-based fuzzy logic control systems.
IEEE Transactions on Fuzzy Systems, **2**, 1 (1994), 46–63.

[10] D. Conte, P. Foggia, C. Sansone and M. Vento
Thirty years of graph matching in pattern recognition.
International Journal of Pattern Recognition & Artificial Intelligence, **18**, 3 (May 2004), 265–298.

[11] F. Cuppens and A. Miege
Alert correlation in a cooperative Intrusion Detection Framework, In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002.

[12] B. Das
Representing uncertainties using Bayesian networks.
(DSTO-TR0918), 1999.

[13] N. Denis and E. Jones
Spatio-temporal pattern detection using dynamic Bayesian networks.
In *42nd IEEE Conference on Decision and Control*, **5** (2003), 4533–4538.

[14] M. H. DeGroot
Optimal Statistical Decisions.
New York: McGraw-Hill, 1970.

[15] J. Friedman
On bias, variance, 0/1-loss, and the curse of dimensionality.
Data Mining and Knowledge Discovery, 1997.

[16] D. L. Hall and J. Llinas
An introduction to multisensor data fusion.
Proceedings of the IEEE, **85**, 1 (1997), 6.

[17] D. Hall and J. Llinas
Handbook of Multisensor Data Fusion.
CRC Press, 2001.

[18] D. Hao, J. Haifeng, H. Zhiyao and L. Haiqing
Data fusion algorithm based on fuzzy logic.par In *Fifth World Congress on Intelligent Control and Automation*, **4** (2004), 3101–3103.

[19] D. Heckerman
Bayesian Networks for Data Mining, Data Mining and Knowledge Discovery.
1997.

[20] N. K. Kasabov
Hybrid fuzzy connectionist rule-based systems and the role of fuzzy fules extraction.
In *International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, **1** (1995), 49–56.

[21] S. Kirkpatric, C. D. Gelatt, and M. P. Vecchi
Optimization by simulated annealing.
Science, **220**, 4598 (May 13, 1983), new series, 671–680.

[22] A. Lockett
Graph-matching intrusion detection system (GMIDS) whitepaper.
21st Century Technologies, 2003.

[23] A. Mahajan, W. Kaihong and P. K. Ray
Multisensor integration and fusion model that uses a fuzzy inference system.
IEEE/ASME Transactions on Mechatronics, **6**, 2 (2001), 188–196.

[24] I. V. Maslov and I. Gertner
Multi-sensor fusion: An evolutionary algorithm approach.
Information Fusion, in press, corrected proof, 2005.

[25] S. Mathew, C. Shah and S. Upadhyaya
An alert fusion framework for situation awareness of coordinated multistage attacks.
Technical Report 2004-18, CSE Department, SUNY at Buffalo, 2004.

[26] S. Mathew, C. Shah, S. Upadhyaya, M. Sudit, V. Garach, A. Stotz and M. Holender
A thermodynamic approach to intrusion alert fusion.
(submitted to *IEEE*).

[27] N. Metropolis, A. Rosenbluth, R. Rosenbluth, A. Teller and E. Teller
Equation of state calculations by fast computing machines.
Journal of Chemical Physics, **21**, 6 (1953), 1087–1092.

[28] J. Pierce and R. John
An Introduction to Information Theory—Symbols, Signals, and Noise.
New York: Dover Publishers, 1980.

[29] C. Qu and Y. He
A method of threat assessment using multiple attribute decision making.
In *6th International Conference on Signal Processing*, vol. 2, 2002, 1091–1095.

[30] C. Shannon
A mathematical theory of communication.
The Bell System Technical Journal, **27** (1948), 379–423.

[31] J. Salerno, M. Hinman and D. Boulware
Building a framework for situation awareness.
In *Proceedings of the Seventh International Conference on Information Fusion*, FUSION 2004, **1** (2004), 219–226.

[32] J. L. Schlabach, C. C. Hayes and D. E. Goldberg
A genetic algorithm for generating and analyzing battlefield courses of action.
Evolutionary Computation, **7**, 1 (1998), 45–68.

[33] A. N. Steinberg, C. L. Bowman and F. E. White
Revisions to the JDL data fusion model.
In *Proceedings of the SPIE Sensor Fusion: Architectures, Algorithms, and Applications III*, Orlando, FL, **3719** (1999), 430–441.

[34] S. C. Stubberud and K. A. Kramer
Data association for multiple sensor types using fuzzy logic.
In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, **3** (2005), 2154–2159.

[35] M. Sudit, A. Stotz and M. Holender
Situational awareness of a coordinated cyber attack.
SPIE Defense & Security Symposium, Orlando, FL, Mar. 2005.

[36] C. Tsallis
Nonextensive statistics: Theoretical, experimental and computational evidences and connections.
Brazilian Journal of Physics, **29**, 1 (Mar. 1999).

[37] C. Tsallis
Entropic nonextensivity: A possible measure of complexity.
Chaos Solutions and Fractals, **13** (2002), 371–391.

[38] A. Valdes and K. Skinner
Probabilistic alert correlation.
In *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, 2001.

[39] C. Wallace and J. Patrick
Coding Decision Trees.
Monash University, Melbourne, Australia, 1991.

[40] S. Wang and N. P. Archer
A neural network based fuzzy set model for organizational decision making.
IEEE Transactions on Systems, Man and Cybernetics, Part C, **28**, 2 (1998), 194–203.

[41] J. Wang and M. Bender
Connectionist decision support systems for multiple criteria decision making.
In *IEEE International Conference on Systems, Man and Cybernetics*, 1991, 1955–1960.

[42] C. Jeung Won, J. Woo and C. Dong Lae
Situation/threat assessment fusion system (staffs).
In *Proceedings of the Fifth International Conference on Information Fusion*, **2** (2002), 1374–1380.

[43] R. R. Yager
On Ordered weighted averaging aggregation operators in multi-criteria decision making.
IEEE Transactions on Systems, Man and Cybernetics, **18** (1988), 183–190.

[44] R. R. Yager
Hierarchical aggregation functions generated from belief structures.
IEEE Transactions on Fuzzy Systems, **8**, 5 (Oct. 2000), 481–490.

[45] R. R. Yager
Generalized OWA aggregation operators.
Fuzzy Optimization and Decision Making, **2** (2004), 93–107.

Moises Sudit obtained his Bachelor of Science in industrial and systems engineering from Georgia Institute of Technology, his Master of Science in operations research from Stanford University and his Doctorate in operations research from Purdue University.

His primary research interests are in the theory and applications of discrete optimization. More specifically, he has been concerned in the design and analysis of methods to solve problems in the areas of integer programming and combinatorial optimization. One primary goal of this research has been the development of efficient exact and approximate (heuristic) procedures to solve large-scale engineering and management problems. As managing director of the Center for Multisource Information Fusion, Dr. Sudit has merged the interests of operations research with information fusion. He has an appointment as research professor in the School of Engineering and Applied Sciences at the University at Buffalo. Dr. Sudit is a NRC Fellow through the Information Directorate at the Air Force Research Laboratory and has received a number of scholarly and teaching awards. He has a number of publications in distinguished journals and has been the principal investigator in numerous research projects.

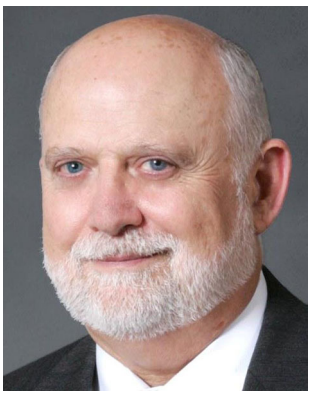
Michael N. Holender received his M.S. degree in industrial and systems engineering at the State University of New York at Buffalo (UB) with a focus in operations research in 2005. He is currently working towards his Ph.D. in the same field of study and will complete it in 2008. He received his B.S. in mathematics and statistics at Miami University, Oxford, OH, in 2002.

He began his career as an actuarial analyst for Univera Healthcare in Amherst, NY performing underwriting functions and supplying rates for employer groups. He streamlined many reporting and data gathering processes allowing future users to cut their time working on projects drastically while also improving accuracy. These interests lead him to graduate school whereupon he began his studies in optimization and process efficiency. He held both teaching and research assistantships throughout his career at UB. He has also taken on summer internships both with CUBRC (Calspan and University at Buffalo Research Center) and the United States Air Force Research Labs in Rome, NY. He currently works as a research assistant through CUBRC on various projects involving data fusion and conceptual spaces as they apply mathematical programming and optimization techniques. He tutors students of all ages in many different fields of mathematics and statistics. He also consults local businesses by creating easy-to-use software systems to increase process efficiency. Michael will continue his studies while acting as principal instructor of an introductory probability course for junior and senior undergraduate engineers at UB.

During his schooling, Michael has served at local president of Omega Rho. He is also a member of Pi Mu Epsilon Mathematics honorary fraternity. His research interests include data fusion, conceptual spaces, stochastic processes and business efficiency.

Adam D. (David) Stotz received a B.S. in computer engineering from the State University of New York at Buffalo in 2003 and is currently there working toward a Ph.D. in operations research with a primary scholarly research focus on distributed situation assessment.

He is currently employed at CUBRC, Inc. in Buffalo, NY where he has served as a research scientist for four years in the information fusion business line. He participates in the technical oversight of numerous projects within the CUBRC and UB comanaged Center for Multisource Information Fusion (CMIF) led by information fusion pioneering researchers Dr. Jim Llinas and Dr. Moises Sudit. His research activities have focused on high level (L2, L3, and L4) fusion in various application domains including cyber security, maritime domain awareness, and threat based routing among others. He has publications in various journals and has presented at numerous conferences on these topics. Adam was appointed as a National Research Council Fellow at AFRL-Rome Information Directorate in 2005 and received an Ethical Hacking certification from the EC-Council in 2006.



John T. (Terry) Rickard (S'67—M'75—SM'01) received the B.S. and M.S. degrees in electrical engineering from Florida Institute of Technology, Melbourne, Florida, in 1969 and 1971, respectively, and a Ph.D. degree in engineering physics from the University of California at San Diego, La Jolla, California, in 1975. He also received Series 7 and 63 General Securities Licenses in 1995 and a Series 24 General Securities Principal License in 1995.

He has 34 years of experience in technology and financial organizations, all of it in management and technology development positions. He began his career working in digital design and testing with Harris Corporation in 1969. He worked part-time as a graduate student for what was then the Naval Electronics Laboratory Center (now SPAWAR Systems Center) in San Diego, California from 1973 to 1975. In 1975, he cofounded ORINCON Corporation, a San Diego-based company specializing in the design and development of state-of-the-art data and information processing solutions for government and commercial customers. He ended his first career with ORINCON in 1994 as senior vice president and technical director. From 1994 to 2001, he served as President and later Chief Scientific Officer of OptiMark Technologies, Inc. He is a coinventor of the OptiMark transaction matching system and was instrumental in the company's development from a start-up enterprise to an operating entity. Rejoining ORINCON in 2001 as senior vice president, his focus has been on broadening the company's technology base, particularly in machine intelligence. When ORINCON was acquired by Lockheed Martin in 2003, he was appointed to the position of senior principal research scientist. In 2005, he was elected a Senior Fellow of Lockheed Martin, for whom he now works from his home in Larkspur, Colorado. His technical expertise includes signal processing, optimization, neural networks, fuzzy and expert systems, and graphical knowledge representation and inference for machine intelligence. His additional expertise includes financial engineering disciplines such as transaction systems, market structures, financial analytics, data mining, derivatives pricing, risk analysis, and trading strategies. His current research interests are in computational intelligence, conceptual spaces, information fusion, content based information retrieval, and nanotechnology.

Dr. Rickard has served on the boards of directors of three companies and has authored numerous technical publications that have appeared in refereed technical journals, books, and conference proceedings. In addition, he has authored several patents, and has several pending patent applications. He currently serves as the Vice Chairman of the IEEE Computational Intelligence Society, Denver Chapter, and is a board member of the non-profit Golden Triangle Research Institute and a Technical Advisory Board member of a nanotechnology hedge fund. In 2006, he received the Author of the Year Award from Lockheed Martin Integrated Systems and Solutions.

Ronald R. Yager received his undergraduate degree from the City College of New York and his Ph.D. from the Polytechnic University of New York.

He has worked in the area of fuzzy sets and related disciplines of computational intelligence for over twenty-five years. He has published over 500 papers and fifteen books. He is considered one the worlds leading experts in fuzzy sets technology. He was the recipient of the IEEE Computational Intelligence Society Pioneer award in Fuzzy Systems. Dr. Yager is a fellow of the IEEE, the New York Academy of Sciences and the Fuzzy Systems Association. He was given an award by the Polish Academy of Sciences for his contributions. He served at the National Science Foundation as program director in the Information Sciences program. He was a NASA/Stanford visiting fellow and a research associate at the University of California, Berkeley. He has been a lecturer at NATO Advanced Study Institutes. Currently, he is Director of the Machine Intelligence Institute and Professor of Information and Decision Technologies at Iona College. He is editor and chief of the International Journal of Intelligent Systems. He serves on the editorial board of a number of journals including the *IEEE Transactions on Fuzzy Systems, Neural Networks, Data Mining and Knowledge Discovery*, *IEEE Intelligent Systems, Fuzzy Sets and Systems*, the *Journal of Approximate Reasoning and the International Journal of General Systems*. In addition to his pioneering work in the area of fuzzy logic he has made fundamental contributions in decision making under uncertainty and the fusion of information.