MEASURING PERFORMANCE FOR SITUATION ASSESSMENT

June 2000

Suzanne M. Mahoney, Ph. D.
Kathryn Blackmond Laskey, Ph. D.
Ed Wright
Keung Chi Ng, Ph. D.
Information Extraction and Transport, Inc.
Arlington, VA 22209

Abstract

We define a situation estimate as a probability distribution over hypothesized groups, units, sites and activities of interest in a military situation. To be useful to decision makers, there must be a way to evaluate the quality of situation assessments. This paper presents and illustrates an approach to meeting this difficult challenge.

A situation assessment integrates low-level sensor reports to produce hypotheses at a level of aggregation of direct interest to a military commander. The elements of a situation assessment include 1) hypotheses about entities of interest and their attributes, 2) association of reports and/or lower level elements with entities of interest, and 3) inferences about the activities of a set of entities of interest. In estimating the quality of a stiuation assessment, these elements may be scored at any level of granularity from a single vehicle to the entire situation estimate.

Scoring involves associating situation hypotheses with the ground truth elements that gave rise to them. We describe why this process may present technical challenges, and present a method for inferring the correspondence between hypotheses and ground truth. Conditional on this correspondence, the quality of the estimate must be assessed according to criteria that reflect the user's requirements. Scores must then be integrated over different plausible assignments. Our scoring methods can be tailored to user requirements, and include rigorous treatment of the uncertainties associated with a situation estimate. We also discuss how to score the situation assessment when ground truth is not available. We discuss how our approach is instantiated in algorithms and describe an example from DARPA's Dynamic Database program.

1. Introduction

Maintaining an accurate assessment of military situations is essential to prevention, containment, and effective response to military conflicts of all kinds. In the post Cold War era, our forces can no longer assume a monolithic enemy operating according to a predictable doctrine using characteristic and easily recognized equipment in a conventional force-on-force conflict. The conflicts most likely to arise in the future are limited engagements or peacekeeping operations in which combatants may use unexpected combinations of equipment and exhibit unpredictable patterns of behavior. Troops are likely to face combatants whose doctrine and tactics are poorly understood. Conflict is likely to break out in areas of the world where the geography is unfamiliar to soldiers and for which military maps are scarce or unavailable. Small, less structured forces and mobile equipment combine to decrease the time window for effective response to enemy actions.

The decreasing predictability of equipment, terrain, and force behavior is counterbalanced by orders of magnitude improvements in sensors, communications, and computing. If the large quantities of available data can be turned rapidly into information, the potential exists for major improvements in the ability of our forces to engage in decisive and timely action in response to military threats. However, the sheer quantity of available data frustrates any attempt at manual integration, and presents difficult challenges to any attempt to counteract the analyst's refrain of "Data, data everywhere, and not the time to think!"

The most well developed fusion methods apply to "level one" fusion (Antony, 1995), or recognition of individual entities such as radars, vehicles, or tracks. Algorithms for level one fusion are becoming increasingly sophisticated and can incorporate information from multiple sources, including different sensors, terrain, and other contextual information. However, there is a real need for aggregating information to form higher level hypotheses of direct interest to commanders. Such higher level hypotheses typically concern collections of entities acting in concert, such as groups of vehicles engaging in certain types of activities of interest to the commander in the given situation. For example, the the Tactical Site, Group, Unit and activity Detection and Assessment system described in Section 2 below has been designed to recognize groups of entities such as reconnaissance units, artillery platoons, and platoon-sized groups of vehicles. These elementary groupings can be aggregated into higher-level operations that involve several such small groups acting in concert to carry out certain activities in a spatiotemporal pattern. The capability to reason about symbolic hypotheses regarding enemy activities and collections of entities is called "level two" fusion, or situation assessment (Antony, 1995).

Although it is clear that an effective capability for automated support for situation assessment would be extremely useful for decision makers, such support would be worse than useless if the situation estimates produced were of poor quality. An essential ingredient, then, to the lifecycle process for situation assessment systems, is a capability for evaluating the quality of a situation assessment. For "live" system test and evaluation, it is necessary to compare a situation assessment produced by an automated system with the actual situation and score how well the estimate matches the situation. There is also a need for metrics to evaluate a situation in the absence of ground truth information. Such metrics can be used in the field to provide commanders with information about how confident they should be in the conclusions reported by the system.

A scoring metric comparing a situation estimate with ground truth must be able to produce more than a simple "yes/no" answer. A military situation is quite complex, involving many actors and entities represented at different levels of granularity. There may be uncertainty about many of the hypotheses reported by the system. To score the estimate, hypotheses must be matched up to and compared with corresponding ground truth situation elements. This matching process itself involves uncertainty. The

quality of a given situation estimate depends crucially on the objectives of the decision maker using it. A given situation estimate may be good for some purposes and disastrous for others.

In developing our approach to evaluation of situation estimates, we considered the goals of the Defense Advanced Research Projects Agency's (DARPA) Dynamic Database (DDB) program: 1) Accurate locations and type identifications; 2) Fresh, continuously updated information; 3) Integrated estimates for easy access and retrieval; 4) Responses that meet the user's information needs while reducing information overload; 5) Very low false alarm rates; 6) Symbolic representations that the user can assimilate.

Our approach meets these goals by considering:

1) *Situation elements at varying levels of granularity*. Our approach can be applied to any type of situation element. Most importantly, the approach allows us to give added weight to those elements that are of most interest to the end user. The situation elements that we measure are at a level that a user can assimilate.

2) *The accuracy of all attributes of situation elements*. Initially, we measure the accuracy of both locations and identification. The methodology supports extending the measures to additional attributes. Furthermore, our methodology permits us to make measurements over a set of time steps, to take into account inferences for which there is no direct evidence, and to measure the accuracy of projections.

3) *False alarms and misses.* The proposed metric reasons about false alarms and misses, so that as the system is tuned to maximize its overall quality, it will also reduce its false alarms and misses.

In this paper we propose an approach to evaluating situation estimates and estimating their quality for a given purpose. Section 2 below describes the elements of a military situation estimate, discusses our approach to constructing situation estimates from low-level reports, and describes the user requirements for a situation assessment scoring metric. Section 3 describes our approach to scoring situation estimates. Key aspects of our approach include scoring hypotheses about a given ground truth element, matching hypotheses to ground truth elements that could have produced them, and combining individual hypothesis scoring with matching to produce an overall score. We also consider the incorporation of user needs in the form of weighting factors for different elements of the situation estimate. In addition, we describe a confidence metric to use when ground truth is not available. Section 4 illustrates our approach with an example.

2. Representing Military Situations

In support of DARPA's DDB program, IET has developed TSGUDA to hypothesize situation elements from incoming vehicle tracks and other evidence. TSGUDA uses sound probabilistic reasoning methods to generate a situation estimate. Each hypothesized situation element that it produces is qualified by an associated probability distribution. First, some definitions:

- *Situation element*: Examples of situation elements are the existence of entities of interest, relationships among entities of interest; attributes of interest for the entities of interest and association of observations with entities of interest.

- *Ground Truth*: Set of actual situation elements.

- *Score:* A qualitative or quantitative value assigned to an element of a situation.

- *Situation*: A set of hypothesized situation elements.

- *Situation Estimate*: A representation of a situation that incorporates a probabilistic evaluation of the hypothesized elements.

Bayesian networks (Pearl, 1988; Neopolitan, 1990), based upon probability theory, are a knowledge representation that effectively captures the uncertainties and conditional independencies present in a given domain. To meet the requirements of DDB, TSGUDA generates a situation-specific Bayesian network (Mahoney and Laskey, 1998) that reflects the current set of observations and inferences that can be made based upon those observations. The ability to generate an on-the-fly Bayesian network to reason about situations requires a sophisticated and efficient set probabilistic inferencing tools. See Figure 1.

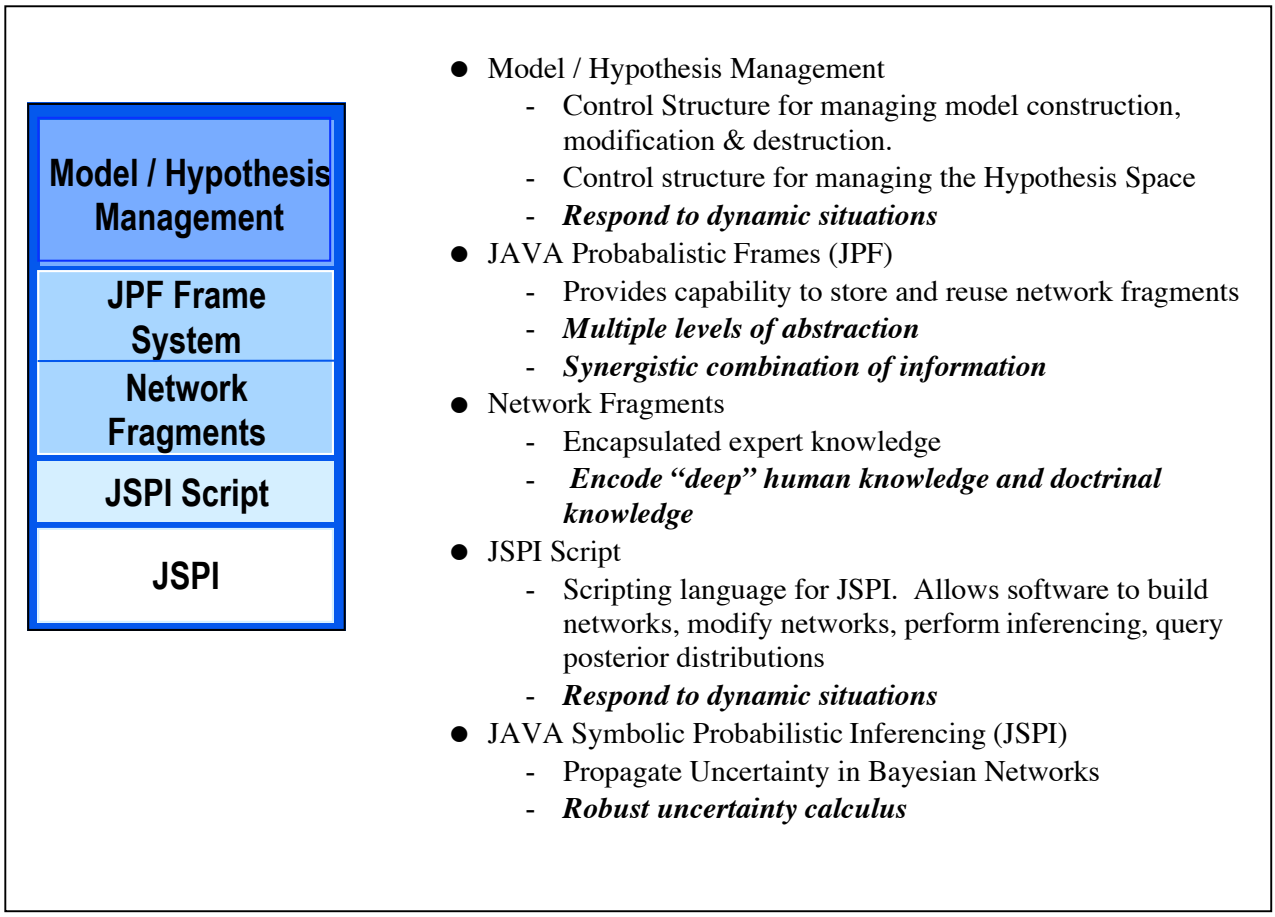


Figure 1. TSGUDA Architecture

The TSGUDA inference engine has several parts: the Symbolic Probabilistic Inference (SPI) engine, the JAVA SPI (JSPI) script language, and the JAVA Probabilistic Frames (JPF) language. The engine constructs problem-specific Bayesian networks from Network Fragments (Mahoney, 1999) and domain specific model / hypothesis management software.

IET's Symbolic Probabilistic Inference (SPI) algorithm (Li & D'Ambrosio, 94) is one of only two known general solution algorithms for Bayesian networks. In contrast to the alternate "join tree" approach to inference in Bayesian networks, SPI has the following two important characteristics. First, SPI is query based. SPI extracts the minimum subset of a Bayesian network that is necessary for each query,

minimizing the amount of computation required for answering the query. Second, SPI has local expressions (D'Ambrosio, 1995), an extension of Bayesian networks, used to express local structure within a node. Local expressions can be used to express many independence relationships including independence of causal influences and context-specific independence. SPI exploits these independence relationships in addition to the conditional independences inherent in Bayesian networks for efficient inference in large Bayesian networks. SPI has successfully computed queries for large "bench mark" Bayesian Networks, which no other exact inference system is able to perform. SPI's query-oriented approach allows for compilation of any probabilistic query into an efficient and small procedural code. Because both the memory and CPU requirement of this generated code is fixed; it is readily usable in an embedded and/or real-time environment. The current version is implemented in JAVA and is called JAVA SPI (JSPI). JSPI Script is an object-oriented scripting language designed specifically for Bayesian network applications. It can be used to dynamically construct Bayesian networks, make situation-specific queries, and define and replace software components on the fly.

JAVA probabilistic Frames (JPF) is a knowledge representation language based on frames (widely used for knowledge representation in Artificial Intelligence) augmented in various ways to express uncertainties. In addition to frame (class) abstractions organized by "is-a" hierarchies inherited from the frame system, JPF supports mechanisms to express uncertainties about the value of variables, the reference to instances, the existence of instances, and the type of instances. JPF allows for expressing domain knowledge as a set of Network Fragments in a modular and compact way, facilitating reuse. Instances of probabilistic frames are created dynamically for each situation, allowing situation specific probabilistic inference. The probabilistic inference is done by JSPI using a Bayesian network created dynamically from the current set of probabilistic frame instances. The generation of Bayesian networks utilizes JSPI's local expressions to exploit all sorts of independence relationships to speed up the inference. JPF is fully integrated with JSPI Script, allowing the user to define frames, create instances, and make situation-specific queries interactively.

In TSGUDA, the probabilistic engine is supplemented by additional code that provides an engineering display, as well as ingestors to read the input data, and detectors to propose hypotheses to be assessed by the engine. The detectors themselves may be either 'rules of thumb' or pre-compiled Bayesian networks that reason about when and which hypothesis to add or remove from the situation estimate.

To meet the general DDB requirements outlined in Section 1 and to handle the uncertainties present in any situation estimate, the metric is designed to meet the following requirements:

1) The user's preferences are key to an overall score:

    a) Situation elements can be scored at level of aggregation of interest to commander;

    b) The user can define attributes of interest (e.g., location, type, activity);

    c) User can specify the relative importance of these attributes;

2) The scoring function:

    a) Rewards situation estimates that place high probability on correct or near-correct values of attributes of interest.

    b) Penalizes false alarms and missed detections.

    c) Handles uncertainty in values of attributes of hypothesized situation elements

d) Handles uncertainty in assigning hypotheses to ground truth elements.

In addition, there are some desirable operational characteristics for the scoring function.

1) *Automatic calculation:* Other than subjective measures and the ground truth when available, the measure for the situation estimate should be calculable from information available in the DDB.

2) *Single measure:* We need to compare the situation estimates generated by different versions of the same processes, different combinations of processes, and the same processes running under different sets of user direction. The measure should be capable of producing a single overall measure for a situation estimate.

3) *Be an aggregation of submeasures*: For example, TSGUDA improves the situation estimate by hypothesizing the existence, identity, location and activities of platoon-sized groups and by improving the identification, location and activities of individual vehicles. We would like to use the situation estimate measure to determine how much of the improvement in the situation estimate is due to better identification of individual vehicles.

4) *Provide partial situation estimates:* We want to measure the changes in the accuracy, timeliness, integration, value, and ROC point of a situation estimate whenever some process modifies the situation estimate. For example, the situation estimate produced by TSGUDA should be more integrated and accurate and have more value than the situation estimate that was input to TSGUDA.

3. Scoring Situation Assessments: A Methodological Approach

To meet requirements identified in Section 2, we have identified four aspects of a situation estimate score. They are defined as follows:

- *Fidelity of individual situation elements*: This is a vector of scores, in which each score compares an element of the situation with a corresponding ground truth element.

- *Overall Fidelity*: The individual fidelity scores can be aggregated into an overall estimate of how well the situation estimate represents ground truth

- *Value:* The value of a situation estimate element measures how well the element meets the needs of a particular user and his/her application.

- *Utility*: The utility of a situation estimate element weighs the value of the element against the cost of producing the element.

Section 3.1 below describes how fidelity of individual elements are measured, assuming that we know the correspondence between situation hypotheses and ground truth elements. However, even when we have ground truth against which to compare a situation estimate, it may not be a straightforward matter to unambiguously assign each situation element hypothesis to the ground truth element that gave rise to it. Section 3.2 describes how to incorporate uncertainty about this assignment. Section 3.3 describes how to aggregate fidelity vectors into an overall fidelity estimate, how to incorporate user preferences, and how to compute overall value and utility assessments. Section 3.4 describes confidence, a measure derived from the model and observations of the situation of interest. We propose that situation estimates with a high confidence also have a high fidelity.

3.1 Measuring Fidelity of Individual Elements

The fidelity vector for a situation estimate is a vector of scores that measure how well an element of the situation estimate matches the ground truth. The fidelity vector is independent of any particular user or application. We simply want to determine whether a hypothesized element has a corresponding ground truth element, and if so, how closely the descriptors of the hypothesized element match those of the ground truth element.

Both the ground truth and the situation estimate vary over space and time. We can score the situation estimate in a number of different ways. Depending upon the objectives of the user, he many use any or all of the ways described below.

1) A simple way is to compare the situation estimate for a given number of reports with the ground truth that generated those reports. The latest time that a ground truth entity gives rise to a report should be the latest time associated with that ground truth entity. The situation estimate provides only the latest fused belief for an entity's existence and its attributes.

2) Another approach is to regard the situation estimate as a history of hypothesized elements. In this case, we compare the situation estimate over an interval of time with the ground truth for that same time interval. The situation estimate provides the belief history for an entity's dynamic attributes (e.g. activity, position, velocity, …) as well as the latest fused belief for the entity's identity and static attributes (e.g. radio, size, …). The history gives the user information to project the forces ahead in time. Again, the latest time that a ground truth entity gives rise to a report should be the latest time associated with that ground truth entity.

3) Another approach is to regard the situation estimate as a snapshot. In this case, we compare the situation estimate for time *t* with the ground truth for time *t*. Time *t* may be any time covered by the reports, as well as short-term projections into the past or future. This assumes that the situation estimate either interpolates or extrapolates all hypothesized elements to time *t*.

The next section describes the problem of matching ground truth with hypotheses. The matching approach is closely tied with the method for scoring situation elements for fidelity, as well as with the models used for recognizing and reasoning about situation elements. In this section we consider how hypothesized elements are scored against ground truth elements.

An important aspect of both scoring and matching is the question of how to define an element of ground truth. Because we are dealing with ground truth elements spread over time, we may choose to divide the temporal scale into *t* time intervals and define *t* ground truth elements for a given physical element, one ground truth element for each time interval. Such a definition permits us to create multiple truthful hypotheses for a given ground truth physical entity.

Scoring a match takes into account a number of different factors. These factors are laid out in Figure 2. The fidelity score vector includes scores for the following attributes of a situation estimate element:

1) *Association Accuracy:* This measures how well reports and hypotheses are associated with a corresponding element of ground truth.

   These measures are based upon a very basic assumption. That assumption is that each ground truth element is associated with no more than one situation estimate element. This assumption is critical to the definition of the association accuracy scores. In developing these measures, we drew upon work

done on evaluation metrics for DARPA's Dynamic Multiuser Information Fusion program (Alphatech, 1998).

a) The *existence score* combines the degree of belief that an entity in the situation estimate exists with whether the entity has a corresponding ground truth element.

   i) The obvious method for calculating this score is to multiply the belief that the entity exists by the truth value for its corresponding ground truth entity. A corresponding ground truth entity either exists (1) or does not (0).

   ii) An existence score should be present for all elements of a situation estimate with the possible exception of the entire situation estimate. While the existence of a corresponding ground truth element is always true, the research question is how to calculate a belief for the situation estimate.

   iii) An element that has an associated ground truth element is called a *true* element while an element that does not have a corresponding ground truth element is called a *false* element.

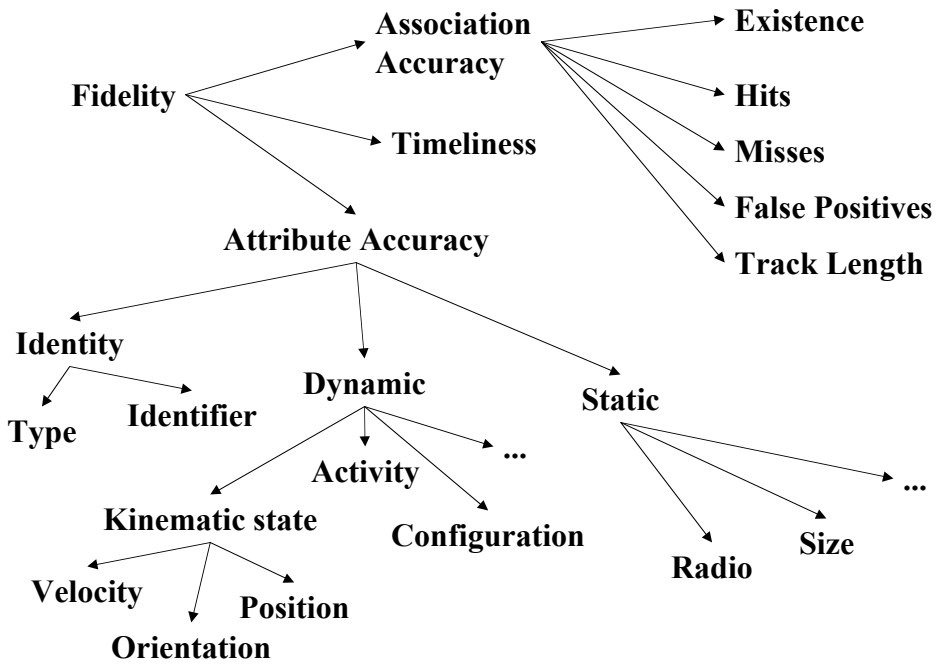


Figure 2 The Fidelity Measure

b) *Hits*, *Misses*, and *False Positives* scores are all ratios of hypothesized elements associated with the next higher element. The denominator is the total number of associated elements. See Figure 3. It shows four hypotheses, H*1* through H*4*, aggregated into hypothesis, H*A*. Each hypothesis on its own may or may not match an element of the ground truth. Each hypothesis is marked with whether or not it matches an element of the ground truth. These ratios may be calculated for all hypotheses.

i) For hypotheses about the lowest level entities, the association accuracy scores are calculated from the number of correctly associated true reports (hits), the number of reports associated with the entity that should not have been associated with the entity (false positives), the number of true reports that should have been associated with the entity and were not (misses).

ii) At the situation estimate level, the score counts the hypotheses that match ground truth entities (hits), hypotheses that match no ground truth entities (false positives) and ground truth entities that have not been hypothesized (misses).

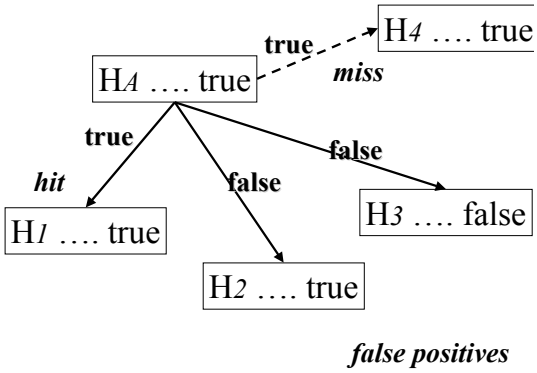


Figure 3 Hits, misses and false positives

There are two kinds of false positive associations: ones that are true hypotheses are called *real false positives* while ones that are false hypotheses are called *bogus false positives*. These should be distinguished in the measurements. An excessive number of real false positives may indicate that we are not associating correctly while an excessive number of bogus false positives may indicate that we are generating too many hypotheses.

iii) While the ratio of hits and false positives to overall number of associations is always less than 1, the ratio of misses to total number of associations may exceed 1.

iv) For hypothesized elements that have no corresponding ground truth element, the score for hits and misses is 0 and the score for false positives is 1.

c) The track length measures the number of associations and span of time or number of time intervals covered by the hits.

2) *Timeliness:* This is a measure of the processing time between an observation and its incorporation into the situation estimate. It includes the time to reason about aggregate entities.

3) *Accuracy:* This measure compares how well ground truth elements match corresponding elements in the situation estimate. Accuracy scores can be interpreted as measuring a kind of distance between ground truth and the hypothesis, in that they are always non-negative and lower values indicate better matches.

a) A situation estimate identifies hypothesized events and their attributes at multiple levels of aggregation from the identities, location and activities of individual vehicles to the identity and composition of operations. The comparison with ground truth should be made across the attributes of an element and at each level of aggregation.

b) For discrete qualitative variables (e.g., type, activity), the match is considered good if the correct attribute value in the ground truth element is given high probability in the hypothesized element. The simplest type of scoring rule penalizes all "wrong answers" equally. For discrete variables we a quadratic rule:

$$S_{Attrib}(G,H) = \sum_{Attrib.value} (1_{Attrib.value} - P_H(Attrib.value))^2$$

In this expression, $P_H(Attrib.value)$ is the probability assigned by the hypothesis to a given value of the attribute. The function $1_{Attrib.value}$ is the indicator function that takes on value 1 if the attribute takes on the given value in the ground truth frame and 0 otherwise.

c) A more complex scoring approach could account for the fact that some kinds of errors cost more than others. For example, it would be worse to misclassify a truck as a small building than as a tank. We also need to consider different levels of granularity in classification. The best case would be to classify a truck as a truck; next best would be to classify it as a vehicle; worst would be to classify it as the wrong kind of vehicle. For such a score, we would identify a cost with each type of error (with zero cost for no error) and compute a weighted sum of probability times cost.

d) For numerical variables, the score is better when the hypothesized value is closer to the ground truth value. If only a point estimate is given, the squared error is an appropriate measure. If an uncertainty distribution is available, an appropriate measure is the mean squared deviation of the hypothesized value of the variable and its actual value in the ground truth frame.

4) *Integration:* This is degree to which the situation estimate correctly associates reports from different sources and temporal-spatial locations with the same entity.

5) *Value:* This is how well the situation estimate responds to the user's guidance.

The commander's information requirements specify situation elements that are important to him. A situation estimate that identifies elements important to the commander is of more value to the commander than a more accurate situation estimate that fails to identify those elements.

6) *ROC point:* This measure compares the false alarms, misses and hits.

7) *Comprehensibility*: This is a subjective measure of how well a trained military user understands the symbology chosen for the situation estimate.

3.2 Incorporating Association Uncertainty

The fidelity scores described in the above section were based on comparing an hypothesized situation element to a ground truth situation element assumed to have generated the hypothesis. Each attribute (e.g., location, activity, type) of the ground truth element has an associated random variable in the hypothesis with an associated distribution over its values. However, there may be uncertainty about which hypotheses were generated by which ground truth elements. A schematic illustration of this situation is shown in Figure 4. For illustrative purposes, only location is pictured. Based on location, hypotheses H3 and H4 could each be assigned to ground truth elements G1 and G2, and hypotheses H1 and H5 could each be assigned to ground truth element G4. Hypothesis H2 is a possible match for G3, but the location error is relatively large. No other combinations of hypothesized and ground truth elements are at all reasonable given location estimates associated with sensor reports.

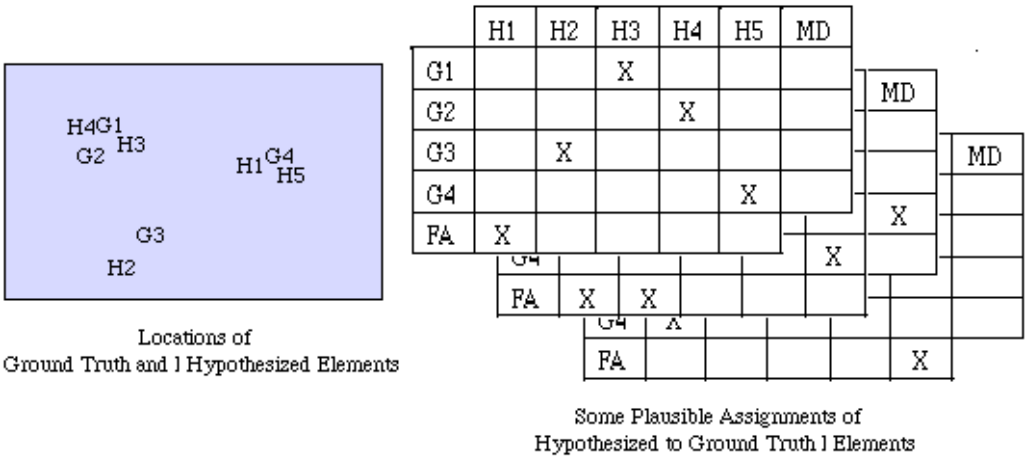


Figure 4 Matching Hypothesized to Ground Truth Elements

In selecting the possible assignments to consider, we made the following assumptions:

1) Each hypothesis was generated by at most one ground truth element.

2) Each ground truth element gives rise to at most one hypothesis.

3) A distance gate is applied to determine feasible assignments of hypotheses to ground truth elements. No pair for which the distance between ground truth and hypothesized location exceeds the distance gate is considered as a possible match. This reduces the combinatorics associated with computing fidelity scores.

All matches consistent with these constraints are considered.

Each *match combination* consists of an assignment of a subset of the hypothesized elements to a subset of the ground truth frames in a way that satisfies the above constraints. Any ground truth element not matched by a hypothesis element is considered a missed detection. Any hypothesized element not matching a ground truth element is considered a false alarm. Thus, each match combination consists of a set of matched pairs, a set of false alarms, and a set of missed detections (any of these sets may be empty).

Now we need to assign likelihoods to the match combinations. It seems reasonable to say that the best assignments (i.e., the ones in which the hypotheses are most faithful to ground truth) are the most likely to be correct. For example, in Figure 4 we would consider match combinations in which H4 matches G1 to be far more likely than match combinations in which H4 matches G3 (the latter would not even meet the distance gate). However, matches between hypotheses and ground truth elements cannot be considered in isolation of the other matches. For example, H5 is near G4 in Figure 4, so we might expect these to be paired in the likely match combinations. However, the likelihood of a match combinations in which H5 does not match G4 depends on whether G4 is matched by something else (H1 in this example). Match combinations in which a hypothesis scores well against a ground truth element, but the hypothesis is a false alarm and the ground truth frame is a missed detection, ought to be assigned very low likelihood.

To implement this idea, we need to define the following likelihoods:

1) *False alarm*:: False alarms are given *a priori* probability β. This is the probability that a given hypothesis is a false alarm.

2) *Missed detections*: Missed detections are given *a priori* probability γ. This is the probability that a given ground truth element will not be hypothesized.

3) *Score if matched*: A score vector $(s_1, \ldots, s_k)$ is assigned a "match likelihood" $f_M(s_1, \ldots, s_k)$, which is interpreted as the *a priori* likelihood that a score vector of this numerical value would be observed for a hypothesis / ground truth pair if the hypothesis matches the ground truth frame. The likelihood $f_M(s_1, \ldots, s_k)$ is computed for each Hi/Gj pair meeting the distance gate.

4) *Score if not matched*: A score vector $(s_1, \ldots, s_k)$ is assigned a "match likelihood" $f_N(s_1, \ldots, s_k)$, which is interpreted as the *a priori* likelihood that a score vector of this numerical value would be observed for a hypothesis / ground truth pair if the hypothesis does not match the ground truth frame. The likelihood $f_N(s_1, \ldots, s_k)$ is computed for each Hi/Gj pair meeting the distance gate.

We can represent a match combination as a table like the ones shown in Figure 4. Each cell in the table corresponds to a likelihood values. Cells representing matches (Hi/Gj cells with an "X") are given a likelihood value of $f_M(s_1, \ldots, s_k)$. Cells representing non-matches (Hi/Gj cells with no "X") are given a likelihood value $f_N(s_1, \ldots, s_k)$.[1] False alarm cells (cells in the FA row with an "X") are given a likelihood value equal to the false alarm probability. Missed detection cells (cells in the MD column with an "X") are given a likelihood value equal to the missed detection probability. Other cells (blank cells in the FA row and MD column) are given a likelihood value of 1 (i.e., ignored in the computation). The overall likelihood is then computed by multiplying these cell likelihood values together.[2] These likelihoods are then normalized to obtain a probability for each match combination meeting the constraints 1-3 above.[3]

Now that we have a probability distribution over match combinations, we can sum over all match combinations to obtain:

1) Expected number of false alarms;

2) Expected number of missed detections;

3) An overall probability, for each Hi/Gj pair, that hypothesized element Hi matches ground truth element Gj.

---

[1] Values $f_M(s_1, \ldots, s_k)$ and $f_N(s_1, \ldots, s_k)$ for Hi/Gj pairs not meeting the distance gate can be ignored in the computation if because these will be non-matches for all match combinations considered. Thus, they contribute only a constant factor, and will drop out upon normalization of probabilities.

[2] To avoid numeric underflows, it is preferable to work in logarithms and add rather than multiply.

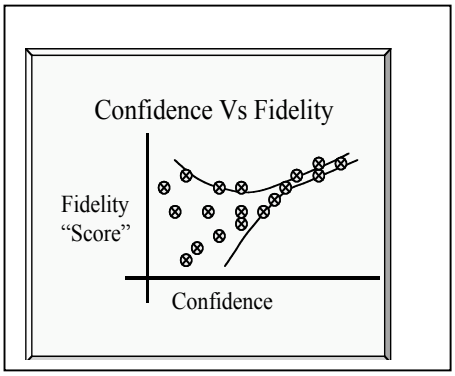
[3] In the scenario of Figure 4, the problem can be broken into three non-overlapping independent problems, for each of which there are no potentially overlapping confusions. When this can be done, the computation can be simplifed by treating the problems separately.

3.3 Aggregating into an Overall Evaluation Metric

To calculate the value of a situation estimate, we assign ask the user to weights to the elements of the situation estimate. Each weight is multiplied by the fidelity of the element and the products are summed to produce an overall value for the situation estimate. Note that the same situation estimate may have different values for different combinations of users and applications.

Utility extends the metric by incorporating the computational cost. For each situation element, the cost is that of the resources required to achieve the calculated fidelity.

3.4 Model Confidence



In general we do not have the luxury of knowing the ground truth at the time that we compute a situation estimate. In the absence of ground truth we want to estimate how good the situation estimate is. We propose a measure of that 'goodness', called *confidence*. Our hypothesis is: if we have high confidence in a situation estimate, then we have a high fidelity measure as well.

We view confidence as a multi-faceted measure. Each facet answers one or more key questions about the situation estimate. The facets include model fit, observational error, observer credibility, and completeness. The questions are:

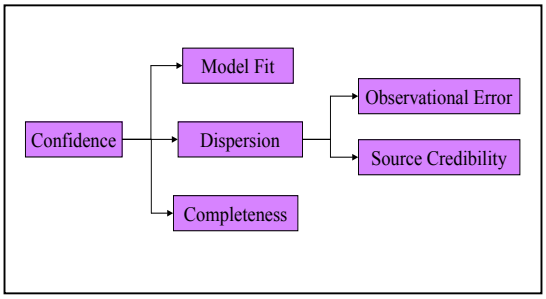
1) Does the situation estimate explain the observations?

2) Are our information sources credible?

3) Are the observations specific/accurate enough to come to a conclusion?

4) Do we have enough observations to make a good estimate?

A desirable trait for confidence is that it be actionable. The measured facets should provide insight into what actions the user can take to improve the confidence.



The following definitions for confidence and its primary facets assume that a Bayesian network model, *M*, has been constructed for a specific situation. A specific confidence measure is relative to a specific node *n*, the *node of interest*.

Model Fit$_M$*:* This is a measure of how well the observations are explained by the model, *M*. It varies between zero (poor fit) and one (good fit). A low score may indicate that the explanation has not been modeled or that we have an extremely rare case. In either case, the model and the data will have to be examined carefully.

Dispersion$_{nM}$*:* This is a measure of the dispersion of the posterior probability distribution for the node of interest *n* in the model, *M*. It also varies between zero (highly dispersed probability distribution) and one

(no dispersion in the probability distributions). Small dispersions are preferred because they indicate that one state of a node has most of the probability mass. Hence, small dispersions indicate a high degree of certainty about which state the node of interest is in.

$Completeness_{nM}$: This is a measure of the number of different types of observations for a model, *M* and a specified node of interest, *n*. It also varies between zero (no observed indicators) and one (all possible indicators observed). In general, values near one are preferred.

$Confidence_{nM}$: This is a measure of the confidence that we have in the posterior probability of a given node *n* in the constructed model, *M*. It varies between zero (no confidence) and one (certainty). Confidence is a mathematical combination of Model Fit, Dispersion and Completeness.

$$Confidence_{nM} = Model\ Fit_M\ *\ (\ W_D\ *\ Dispersion_{nM} + W_C\ *\ Completeness_{nM})$$
$$where\ \ 0 < W_j < 1$$

If the model explains the observations well, then the dispersion and model completeness dominate the overall model confidence measure. If the model does not explain the observations well, then, the quality of the observations (dispersion) and numbers of observed variables (completeness) are inconsequential. The result is a confidence measure that varies from zero to one with one being the ideal.

An overall model confidence is the normalized sum of the confidences for the nodes of interest in the model.

3.4.1 Model Fit

Model Fit is a measure of how well the model fits the data. The basic measure compares the joint probability of the observations with the probability of each observation considered as an independent event. See Laskey (1991). The node of interest can be thought of as an event that explains the relative likelihoods of one or more factors, each of which has indicators that can be observed. Any one event will have a characteristic pattern. When the observations of some of those indicators match a pattern that has been modeled, then the probability of seeing additional indicators matching the pattern rises. Intuitively, the joint probability of observed indicators that fall within a particular pattern should be relatively large compared to those same observations occurring by chance. Unless the event of the node of interest is extremely rare, if the joint probability of the observations is less than the probability of the observations occurring independently, it is probably the case that the model has not included the explanation of the observed events.

Let $O = \{O_1, \ldots, O_k\}$ be a set of observations. Using available model parameters the inference engine can easily calculate: 1) the joint probability, $J_O = P(O_1, \ldots, O_k)$ and 2) the probability of the observations occurring independently, $I_O = P(O_1)*\ldots*P(O_k)$. Now, we want the measure for model fitness to vary between zero and one, with one being the desired state. Ideally, I_O is small compared with J_O. If we take the ratio

$$R = I_O / J_O \text{ and if } I_O / J_O > 1, \text{ then } R = 1,$$

then a measure for Model Fit that meets the requirements is:

$$1 - R.$$

3.4.2 Dispersion

Essentially dispersion is a measure of how 'peaked' the distribution over the sates of a node is. The measure itself is straightforward. How good such a measure is depends upon the accuracy of the observations and the credibility of the observer.

The dispersion of the node of interest can be measured directly from the posterior probability distribution. In any model, the dispersion of the node of interest automatically incorporates the precision of the observations and the number of the observations being made. We use a standard distribution when the node represents a numerical value -either discrete or continuous. Standard distribution may also be used for ordinal states (for example: small, medium, large) where a numerical value can be assigned to each state. When the discrete node states have no numerical value and the order of the states is not important (For example: sniper, bomb, gas) we use entropy as a measure of dispersion. Entropy of a node N with n states, s_i, each with a probability, $P(s_i)$, is:

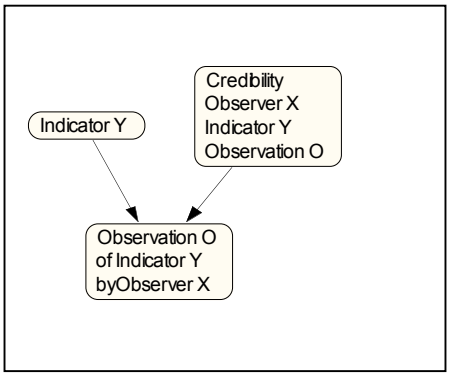
$$ENT(N) = \sum_{i=1}^{n} P(s_i)\log_2(P(s_i))$$

If it is necessary to combine entropy values or standard deviations with other metrics into a total confidence score, it is appropriate to scale the standard deviation or entropy value into a range from 0.0 to 1.0, where 1.0 is a large (maximum) dispersion, and 0.0 is a small (minimum) dispersion. This scaling can be accomplished by the following:

Entropy score = Ent(N)/Log_2(n) where n is the number of states in the node

Standard Deviation score = (Std Dev)/(.5 * (max value - min value))

Observational error is automatically managed by the inference engine as long as the equivalent of likelihood ratios are entered as observations. In the case of sensors making observations, if the error model for the sensor is reflected in observational error then we need take no further actions. If a sensor's error model and observation are reported separately or in the case of a human observer, both the accuracy of the observation made and the credibility of the sensor are required. For a fuller discussion of credibility and how to measure it see Schum (1994).



The observational error cannot be removed from the dispersion for the Node of Interest in a sound manner. However, by setting states for credibility nodes to the 'Perfect' state, we can effectively remove the uncertainty attributable to the lack of credibility for the experts and model sources. Hence we can compare the calculated dispersion measure with expert and model credibility factored in with the dispersion measure without credibility factored in.

3.4.3 Completeness

From a practical point of view, a completeness measure should indicate whether we have enough information to make a *reasonable inference* about the situation at hand. Ideally, the measure or measuring process would also tell us *what information is missing* and *rank order the importance* of obtaining the missing pieces of information.

We define a reasonable inference to be one that is specific enough to determine a course of action. This means that one has to be confident enough to take action and also that additional information will not change the indicated treatment/repair strategy. So a completeness measure may be defined as: *A measure of completeness measures the amount and type of available evidence for a node of interest and whether that evidence is sufficient to act upon the belief for the node of interest.*

Our strategy for measuring completeness is to enumerate the possible observed indicators for the node of interest, note whether an indicator has been observed, and weight each observed indicator by its value of information. Using standard functions, the value of information, $V_{j,n}$, of each observed indicator, *j*, for the node of interest, *n*, can be easily obtained from the constructed network

For the completeness measure we suggest the following ratio:

$$Completeness_{nM} = (V_{1,n} * S_1 + \ldots + V_{m,n} * S_m) / (V_{1,n} + \ldots + V_{m,n})$$

where S_j is set to 1 if node *j* has be observed and it set to zero otherwise and where m is the number of observable nodes in the constructed network.

3.4.4 Taking Actions Given the Confidence Measure for a Node of Interest

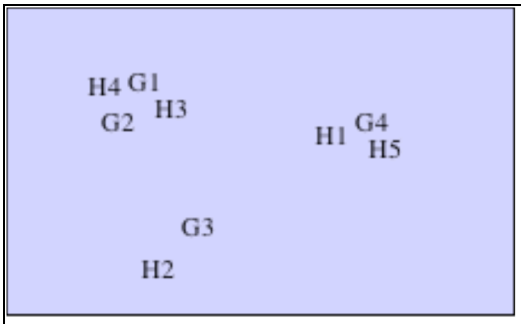
Whatever the value of the other two major terms, if model fit is relatively low, then the end user should work with a model expert to determine the source of the poor fit. The first thing for the model expert to do is to determine if the data fits a rare case. If the data does not fit a rare case, then the modeling expert and user need to work together to determine what the model is missing.

If the model fit is good, then the user should pay attention only to the dispersion and completeness terms. In these cases, the only way to improve the confidence is to make more observations: either 1) observations of previously unobserved indicators or 2) additional observations of previously observed indicators. The first case will cause the completeness term to increase and, to the extent that these additional uncertain observations agree with previous observations, the dispersion term should also increase. The second case will improve the dispersion term to the degree that new information agrees with the old. Unless the completeness is relatively high, attention should first be paid to the completeness term for more observations may significantly alter the dispersion term.

There are several reasons why the dispersion term may be low. These are 1) the observations are simply imprecise; 2) the sources are not very credible; 3) there is conflict in the data. The first two reasons can be overcome by better sensors and/or sensing conditions while the third may require some analysis. Conflict may reflect either poor sensing or a model that does not explain what is being sensed. When conflict is severe, the model fit should not be high.

4. Example

The methods described in this paper were applied to the example scenario illustrated in Figure 5. The spatial layouts of four ground truth units and five hypotheses are shown, along with the types of the ground truth units and the probability distributions for type and existence for the five hypotheses.



Ground Truth Element	Type
G1	Armor Platoon
G2	Armor Platoon
G3	Mechanized Infantry Platoon
G4	Logistics Platoon

	P(Type)				
Hypothesis	Artillery	Maneuver	Logistics	Other	P(Exist)
H1	0.03	0.04	0.94	0.01	0.88
H2	0.03	0.92	0.02	0.03	0.94
H3	0.92	0.03	0.04	0.01	0.98
H4	0.98	0.01	0.005	0.005	0.97
H5	0.02	0.05	0.88	0.05	0.18

Figure 5 Situation and Hypothesis Scores

We computed match scores as described in Section 3.1. These scores are shown in Table 1. We then computed match probabilities of all match combinations, and aggregated them into probabilities for each ground truth / hypothesis match, along with the probabilities that each hypothesis is a false alarm and each ground truth element is a missed detection. These probabilities are shown in Table 2.

	H1	H2	H3	H4	H5
G1	--	--	0.14	0.16	--
G2	--	--	0.18	0.13	--
G3	--	0.25	--	--	--
G4	0.19	--	--	--	0.44

Table 1 Fidelity Scores for Hypothesis / Ground Truth Matches

	H1	H2	H3	H4	H5	MD
G1	0	0	59.8%	37.2%	0	3.0%
G2	0	0	37.0%	59.9%	0	3.1%
G3	0	94.2%	0	0	0	5.8%
G4	75.1%	0	0	0	21.7%	3.2%
FA	24.9%	5.8%	3.3%	2.9%	78.3%	

Table 2: Assignment Probabilities for Ground Truth / Hypothesis Pairs

As discussed above in Section 3, the overall evaluation of a situation involves aggregating the individual element fidelity scores into an overall score. We can form probability-weighted averages for any metric the decision maker wishes to see. Probably the most commonly used situation evaluation metrics are the number of false alarms and the number of missed detections. The expected number of false alarms, where the expectation is taken over all match combinations meeting the distance gate, is the sum of the probabilities in the FA row of Table 2, or 1.15. Similarly, the expected number of missed detections is the sum of the final column of Table 2, or 0.15. Notice that the difference between the number of false alarms and the number of missed detections must be equal to 1.0, which is the number of excess hypotheses over the number of ground truth elements. If there were more ground truth elements than hypotheses, the number of missed detections would have to exceed the number of false alarms by the number by which the number of ground hypothesized units exceeded the number of hypothesized elements.

Typically, we use a ROC curve to evaluate performance and calibrate settings of a single-entity fusion system. The ROC curve plots the probability of a missed detection against the probability of a false alarm, when it can be assumed that there is no association error in matching hypotheses to ground truth units.

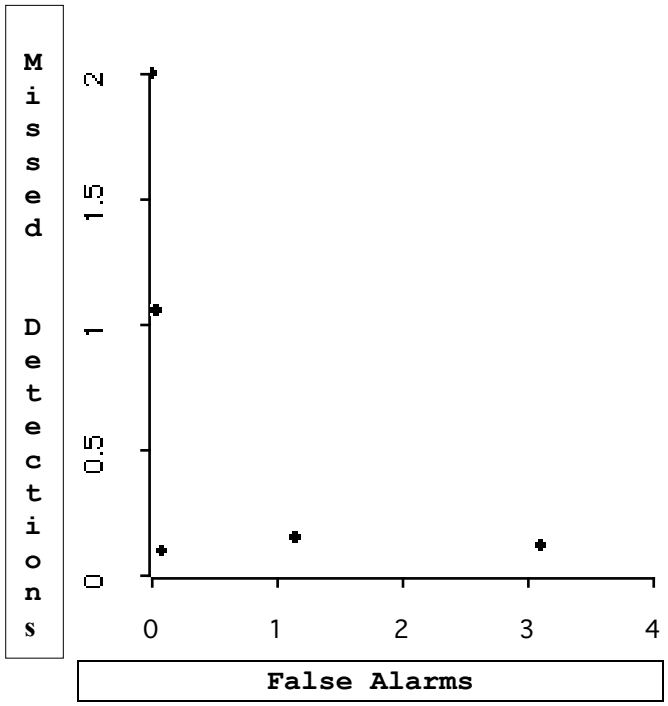


Figure 6 Expected Missed Detects versus Expected False Alarms for Example Scenario

Similarly, we can run the same scenario with different detection thresholds, and corresponding changes in the missed detection and false alarm probabilities in the match algorithm. This yields a comparison of the expected number of missed detects versus false alarms in the system output, where the expectation is taken over all possible assignments of hypotheses to ground truth elements, as described above. For example, in our scenario, if the threshold for declaring hypotheses were tightened, H5 would no longer appear as a hypothesis. If the threshold were tightened still further, H2 would be dropped, followed by H1. Figure 6 shows a plot of the expected number of missed detects versus the expected number of false alarms as the threshold is varied. To evaluate the operating characteristics of a situation assessment

system, then, we would repeat this analysis across different runs of the system to determine appropriate thresholds for declaring hypotheses.

The quality of a situation assessment involves more than just whether a ground truth element is detected. We can also consider costs associated with errors in type, activity, or other features of the ground truth element.

5. Summary and Conclusions

We have proposed a theoretically sound and computationally practical approach to evaluating the quality of situation assessments. With this approach we can evaluate situation models against ground truth for test and evaluation purposes. In addition, we can compute confidence measures for evaluating models at run time when no ground truth data is available. We have described our approach and demonstrated its practicality with an example.

Acknowledgements

Work for this paper was performed under government contract number, F33615-98-C-1314, Alphatech subcontract number 98036-7488. The authors wish to give thanks to the IET developers of TSGUDA: Bruce D'Ambrosio, Masami Takikawa, Dan Upper, Rui Yang, Scott Johnston, and Steve Langs. We also thank Gary Patton of Veridian for domain expertise and Tod Levitt of IET for technical guidance. Finally, we thank Otto Kessler of DARPA for providing the necessary vision to challenge us to develop a metric for measuring the quality of a situation estimate.

References

Alphatech (1998) *Evaluation Metrics for Dynamic Multiuser Information Fusion.*

Antony, Richard T. (1995) *Principles of Data Fusion Automation,* Artech House, Inc.

Balci, O. (1998) Verification, Validation, and Accreditation, in *Proceedings of the 1998 Winter Simulation Conference*. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Mainivannan, eds.

D'Ambrosio, B. (1995) Local Expression Languages for Probabilistic Dependence. In *International Journal of Approximate Reasoning,* P. P. Bonissone (ed), North-Holland, New York. pp.61-81.

Hodges, James S. and James A. Dewar. (1992) "Is It You or Your Model Talking? A Framework for Model Validation." Rand Publication Series, R-4114-AF/A/OSD.

Kleijnen, J. P .C. (1994) "Sensitivity Analysis Versus Uncertainty Analysis: When to Use What?" in *Predictability and Nonlinear Modeling in Natural Sciences and Economics* edited by J. Grasman and G. van Straten. Kluwer Academic Publishers, 1994.

Laskey, K.B. (1991) Conflict and Surprise: Heuristics for Model Revision, in *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference,* Morgan Kaufmann Publishers, San Mateo, CA. pp. 197-204.

Laskey, K.B. (1995) *Model Uncertainty: Theory and Practical Implications*, *IEEE Transactions in Systems, Man and Cybernetics*.

Li, Z. and D'Ambrosio, B., 1994, Efficient inference in Bayes networks as a combinatorial optimization problem. *International Journal of Approximate Reasoning* 11:55-81.

Mahoney, Suzanne (1999) *Network Fragments*, Ph.D. Dissertation, George Mason University, Fairfax, VA.

Mahoney, S.M. and K.B. Laskey (1998) Constructing Situation-Specific Belief Networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference*, G. Cooper and S. Moral (eds.), San Francisco, CA: Morgan Kaufmann. pp. 370-378.

Neopolitan, R.E. (1990) *Probabilistic Reasoning in Expert Systems*, New York, John Wiley & Sons, Inc.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA, Morgan Kaufmann.

Van Horn, R. L. (1971) "Validation of Simulation Results," *Management Science*. **17**: 5.

Waltz, E. and Llinas, J., (1990) *MultiSensor Data Fusion*, Artech House, Inc., Norwood, MA.

Wright, E. (1997) Information Integration for Remote Sensing, In *Proceedings 1997 ISSSR Conference*, San Diego, CA.

Yin, R. K.. (1994) *Case Study Research Design and Methods*. Sage Publications.