USE OF DOMAIN KNOWLEDGE MODELS TO RECOGNIZE COOPERATIVE FORCE ACTIVITIES

June 2001

Edward J. Wright
Suzanne M. Mahoney, Ph.D.
Kathryn Blackmond Laskey, Ph.D.
Information Extraction and Transport, Inc.
1911 North Fort Myer Drive
Suite 600
Arlington, VA, 22153

ABSTRACT

This paper provides results of experiments on the use of domain knowledge models to recognize military groups, units, and activities from input track data. The group, unit and activity assessments are provided in the form of a situation estimate which is evaluated against ground truth. Our experimental architecture allows us to evaluate the quality of the resulting situation estimate as a function of the quality of the input track data.

A situation assessment integrates and fuses 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 situation 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 have previously presented a method for scoring the quality of the situation estimate. In this paper we extend the previous results to the ability to score estimates of hierarchical groups / units and their activities. We discuss how our approach is instantiated in algorithms and describe results of experiments performed for DARPA's Dynamic Database program.

1. INTRODUCTION

1.1. BACKGROUND

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 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 units and 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).

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.

Our approach for scoring situation estimates 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 both false alarms and misses. Moreover, the relative priority of false alarms versus misses can be tuned by the user.

4) Our previous work (Mahoney, et al, 2000) implemented a situation assessment fidelity measure that was capable of scoring a single level of a group hierarchy. This year's work has extended our knowledge models, used to generate the situation estimate, and the fidelity

scoring measure, to more realistic situations containing hierarchies of military groups, units and activities.

1.2. ORGANIZATION OF THE PAPER

The paper is organized in the following sections: Section 2 describes the operation of Tactically Significant Group, Unit, and activity, Detection and Assessment (TSGUDA) software component, and the role of knowledge domain models in forming a situation estimate. Section 3 describes our experimental architecture and the extensions to our situation assessment scoring to handle hierarchical groups, units, and activities. Section 4 describes results of experimental runs of TSGUDA and evaluation of the quality of situation assessments as a function of the quality in input track data. Section 5 presents conclusions.

2. TACTICAL SITE, GROUP, UNIT, AND ACTIVITY DETECTION AND ASSESSMENT

Bayesian networks (Jensen, 1996), 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 (Mahoney, et al, 2000).

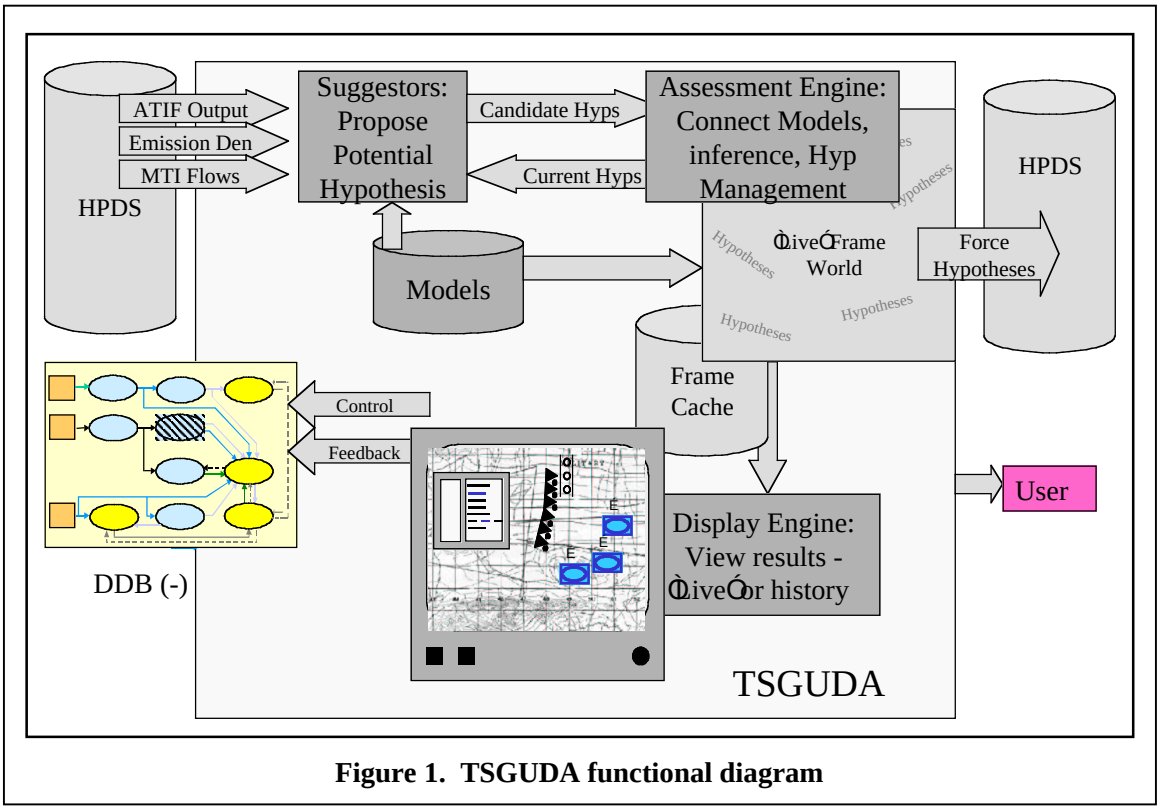

Figure 1. TSGUDA functional diagram

In support of DARPAÕs 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. In operation, TSGUDA builds a situation specific Bayesian Network, from a database of domain knowledge models represented as Bayesian Network fragments (Laskey and Mahoney, 1997). Queries against the situation specific Bayesian Network provide probabilistic assessments of hypotheses that form the situation estimate.

Figure 1 shows the operation of TSGUDA as it is used in the DDB program. Input data is ingested from the High Performance Data Store (HPDS). This data consists of fused track produced by the All source Track and ID Fusion (ATIF) component, as well as MTI flow data and SIGINT Emissions density data. The first functional component of TSGUDA is the ÒsuggestorsÓ They identify possible hypotheses which are passed to the assessment engine. The suggestors use information from the HPDS, the available knowledge models, and the current hypotheses maintained in the assessment engine. The role of the suggestors is to detect, with a high probability of detection (and corresponding high false alarm rate) many possible candidate hypotheses from the data.

The candidate hypotheses are sent to the assessment engine, that is responsible for building and maintaining the situation specific Bayesian Network. Hypotheses in the situation estimate are represented by nodes, or collections of nodes, in the Bayesian Network. The current Bayesian Network can be queried at any time to provide an assessment of any hypotheses. The assessment engine is also capable of performing hypotheses management, by periodically evaluating all, or a specific subset of the probabilistic hypotheses and either eliminating them or declaring them to be true.

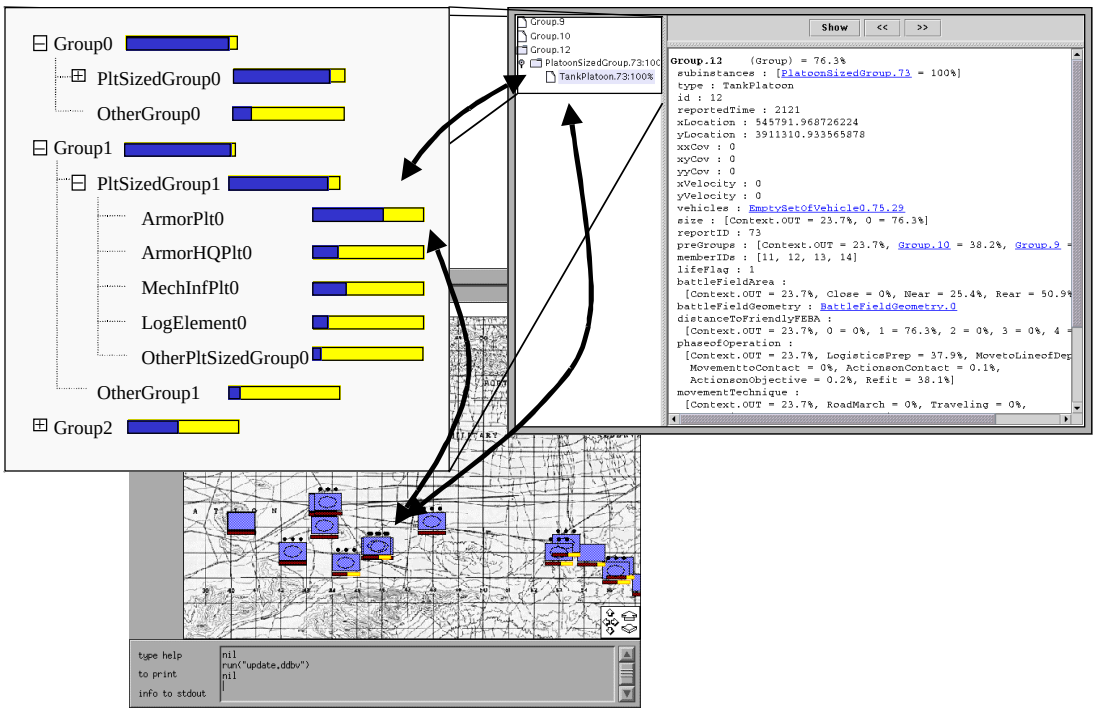

Figure 2. Visualization of TSGUDA results

The situation specific Bayesian Network is maintained by the assessment engine. It changes dynamically over time, as suggesters present new candidate hypotheses, and as the hypothesis management functions prune the network. There is also a capability to store the current state of the network in the "frame cache". This provides the capability to store the history of the situation estimate as it evolves over time, and a future capability to "backtrack" to an earlier state if the system discovers that it is diverged too far from reality.

The situation estimate, or force hypotheses, generated by TSGUDA is stored in the HPDS, and is also available to the user in the form of a symbolic map display. In the future, the results may also provide information to support feedback and control to other DDB components.

Figure 2 shows an example of the types of information in the TSGUDA situation estimate. The unit icons on the map represent the location and unit type of the hypothesized units. The unit type symbol shown is the highest assessed type probability from a range of alternatives maintained by the system. The unit symbol also included a bar underneath it that displays the assessed probability of existence of the unit. The unit icon can be linked to a textual display of other hypotheses related to the unit, and to a visualization of the hypotheses tree for the unit type.

2.1. DOMAIN KNOWLEDGE MODELS

Domain knowledge is used in two places in the TSGUDA system. First, domain knowledge is represented as Bayesian Network fragments. These fragments are used to construct the situation specific Bayesian Network. Second, domain nowledge is compiled into rapidly executing suggestors that nominate hypotheses for consideration by the assessment module. The suggestors may also use Bayesian Network models or influence diagrams, or may use domain knowledge encoded in cruder hueristics.

2.1.1. Bayesian Network Domain Models for Military Situation Assessment

Military situation assessment is the process of reasoning about battlefield entities of interest given evidence about those entities. Some questions about battlefield entities of interest include:

1) Are there any entities of interest present?
2) If so, what type of entities are they?
3) What are these entities doing?
4) Are the entities of interest acting cooperatively with one another?
5) Is there a significant activity (i.e. an attack) occurring and if so, where is it?

The evidence about entities of interest is frequently at the level of individual vehicles. At this level, an entity's activity is typically described in the simplest of terms (e.g. moving or stationary) and entity type information is usually partial. Location and time are also reported. Notably, all the evidence is uncertain.

The assessment process that uses this evidence is a hierarchical one. Given military domain knowledge an intelligence analyst can infer a great deal about the situation represented by the evidence. First, one can infer the presence of a group of interest from a set of reports about vehicles. A group is defined to be a set of vehicles operating together over a period of time. From the simple activities and locations of the vehicles in the group, one may also suggest the group's formation and infer whether the group is moving or not. From these basic inferences, an analyst

can infer more complex group activities. For example, the movement technique for the group is one of: RoadMarch, Traveling, Traveling Overwatch, Bounding Overwatch, Assembly, Defensive Posture, or Combat Movement. In addition, the type of the group can be inferred from the types of vehicles that are presumed to be members of the group. In turn, groups themselves may be clustered into higher level groups and the type of higher level group can be inferred from the types of the clustered groups. Furthermore, by examining the distribution of groups and their activities across the battlefield, one can infer whether a significant activity, (e.g., an attack) is about to occur or is in process.

TSGUDA performs all of the above tasks to reason from vehicle level ÒreportsÓ to platoon and company sized groups and their activities to whether or not a significant activity is taking place and if so, where it is occurring. The domain knowledge that supports these inferences is of two flavors: Generic domain knowledge and specific hypothesis management knowledge.

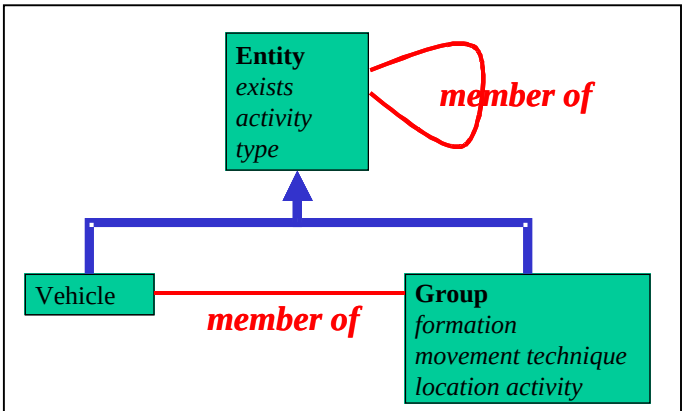

Figure 3. Entity Frames

The *generic domain knowledge* describes entities of interest and their relationships with one another. These generic domain models may be composed in a variety of ways to represent any situation of interest. We use frames to represent the entities of interest. Each frame specifies the relevant attributes and relationships of the entity. The frames are arranged in conceptual hierarchies. See Figure 3. This figure shows that Vehicles and Groups are both members of a conceptual hierarchy with a common parent, Entity. They both inherit the attributes of exists, activity, and type from Entity. The member of relationship is shown in red. In general, an Entity is a member of another Entity and specifically a Vehicle is a member of a Group. Not shown are the member of relationships among different types of groups.

Because the knowledge about these entities is uncertain, each attribute is represented by a set of mutually exclusive and exhaustive possible states. The value of an attribute is represented by a probability distribution across the possible states. Relationships are also uncertain. At any one time, a Vehicle is a member of exactly one Group. However, due to uncertainties and incompleteness of evidence, we may be uncertain about which Group a Vehicle belongs to or even if it is a member of a Group. This is called *reference uncertainty*. Like attributes, the uncertainty is represented by a probability distribution across a set of possible states, in this case a list of Groups to which the vehicle may belong.

We use Bayesian network fragments to represent the relationships among the frames. See Figure 4. This represents the probabilistic relationship

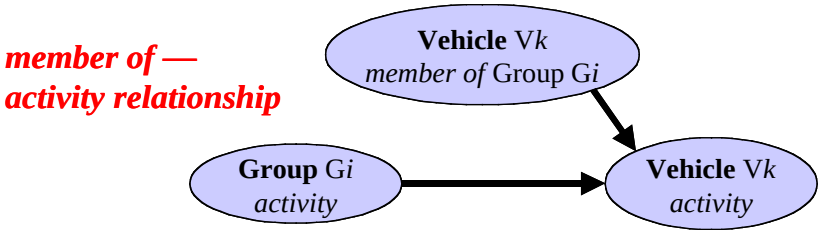

Figure 4. Network Fragment

between Group G*i*'s activity and Vehicle V*k*'s activity depending upon whether or not Vehicle V*k* is a member of Group G*i*.

Of special interest is the *exists* attribute. This attribute represents the uncertainty about whether a hypothesized entity of interest is actually a false alarm. We call this *existence uncertainty.*

In addition to frames for the entities of interest, TSGUDA's knowledge base contains report frames that represent evidence from outside sources such as ATIF and from special TSGUDA algorithms, such as group clustering. TSGUDA also contains an Attack Reasoning frame designed to infer whether or not an attack is occurring from the distribution of Groups across the battlefield.

A *situation estimate* consists of instances of frames and their probabilistic relationships that have been linked into a Bayesian network. Probabilistic inference generates posterior belief in uncertain attributes of the entity and attack reasoning frames given evidence and given the hypothesis management decisions made by TSGUDA.

Specific hypothesis management knowledge is used by TSGUDA to construct a situation (Laskey, et al, 2001). Hypothesis decisions are of four types:

- *existence hypotheses:* Given evidence, an instance of an entity may be instantiated. Once instantiated, an entity may be removed from the situation if the posterior belief for the ***exists*** attribute is too low. Or, if the belief for ***exists*** is extremely high, the instance may be made certain.
- *type hypotheses:* Because evidence is usually partial, there is often uncertainty about the type of entity that has been instantiated. With additional evidence, the type may be narrowed to one or more of the subtypes of the entity. For example, we may be "sure" that a particular vehicle is a tank while remaining uncertain about the precise type of tank. In that case we can subtype the vehicle as a tank.

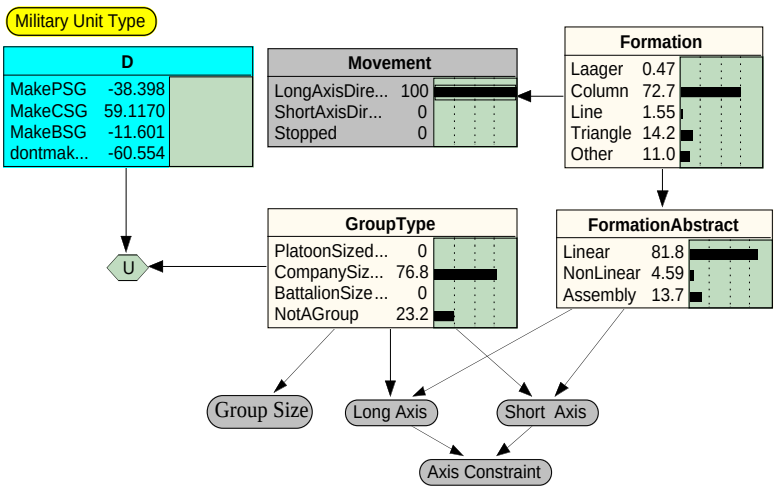

Figure 5. Group Type Given Numbers of Vehicles and Bounding Box Dimensions

- *association hypotheses:* Association hypotheses represent uncertainty about the relationships among entities, or between entities and reports. Examples are hypotheses about which if any of several previously observed entities gave rise to a reported vehicle track, or which of several hypothesized groups contains a given vehicle.

- *identity hypotheses:* When two hypothesized instances are very similar, they may correspond to the same entity. For example, if a track is lost and then observed again, the previously hypothesized vehicle instance may still be represented along with a newly hypothesized instance. Alternatively, there may be several different platoon hypotheses sharing many of the same vehicles. In this case, duplicate hypotheses may be eliminated or combined with other hypotheses.

In the TSGUDA architecture, hypothesis management decisions are often made by *suggestors*. Some suggestors simply pass on evidence from outside processes such as ATIF. Other suggestors, are themselves special algorithms. A group cluster suggester hypothesizes groups (existence hypotheses) and their associated vehicles (association hypotheses). Other suggestors use influence diagrams to make decisions. Figure 5. shows the influence diagram used to determine whether a cluster of vehicles on the ground qualifies as a Platoon-Sized (PSG), Company-Sized (CSG) or Battalion-Sized (BSG) group. It also gives a probability distribution for the group formation. Note that one of the decisions shown in the blue box is *not* to hypothesize a group.

2.1.2. CLUSTER SUGGESTORS

An important suggestor in the DDB application is the cluster suggestor that suggests candidate hypotheses about the existence and membership of clusters of ground vehicles.

Tracking clusters is qualitatively different from tracking vehicles, and it introduces a different kind of uncertainty. Individuals can join groups and depart from them, and whole groups can merge and split. Even if we had perfect vehicle tracks, we could still have doubts about whether or not two clusters with overlapping but not identical membership should be described as the same group. Our current code uses simple, but brittle criteria. For example, a cluster in the current data is considered a continuation of a cluster track if its membership differs from the most recent membership of the cluster track only by the arrival and departure of solitary vehicles. Otherwise the cluster is used to initiate a new cluster track. Unfortunately, this criteria produces many spurious cluster tracks. For example, if two groups briefly move close enough to appear as a single cluster and then separate again, the number of clusters being tracked increases by three: 1)˚the initial cluster tracks are terminated, 2)˚a larger, short-lived cluster track is created and then terminated, and 3)˚two new small cluster tracks are created.

Future work will implement more sophisticated cluster reasoning. Two main changes have been identified to make the cluster suggestor more robust. First, when we associate a current cluster with an old cluster track, we need to consider cluster tracks which were not updated in the previous update cycle. This is analogous to coasting a vehicle track. Second, we need to use more flexible criteria concerning cluster track membership. For instance, if a cluster of three vehicles joins a cluster of 12 vehicles, we should treat the merged cluster as a continuation of the larger cluster track, but if clusters of size 5 and 6 merge, we should probably create a new cluster track. We will probably do this initially using quantitative heuristics such as a measure of membership similarity. We may instead use a principled Bayesian network model. Another possibility is to estimate a degree of association between pairs of vehicles, based on how often they have been in close proximity.

3. EXPERIMENTAL ARCHITECTURE AND SCORING

The ability to generate a situation estimate has little value unless there is some confidence that the estimate is accurate enough to be useful to a decision maker. An evaluation can be done subjectively by visually comparing a situation estimate with known ground truth. But a subjective evaluation is time consuming to perform and is of little value in quantifying the effects of small changes in domain models, suggestor logic, or hypothesis management. In addition, there are only a very few ground truth data sets available to the DDB program. Based on these concerns we have developed an experimental architecture that allows us to systematically evaluate TSGUDA and its components. The experimental architecture is shown in figure 6.

3.1. TSGUDA EXPERIMENTAL ARCHITECTURE

Because of the limited real world ground truth, and the need to evaluate TSGUDA performance against a variety of scenarios, we have developed a capability to generate simulated scenarios. These simulations define the types, membership, and activities of a hierarchy of military units, and then generate each specific vehicle track. The vehicle tracks are then processed to simulate the TSGUDA inputs — normally received from ATIF. The processing may be Òerror freeÓ to provide ground truth data to TSGUDA, or it may simulate the types of errors characteristic of real TSGUDA input data. We have implemented error models for probability of vehicle type ID, probability of detection, probability of correct association (of a vehicle track at one time step with the correct track at the next time step), and a false alarm rate.

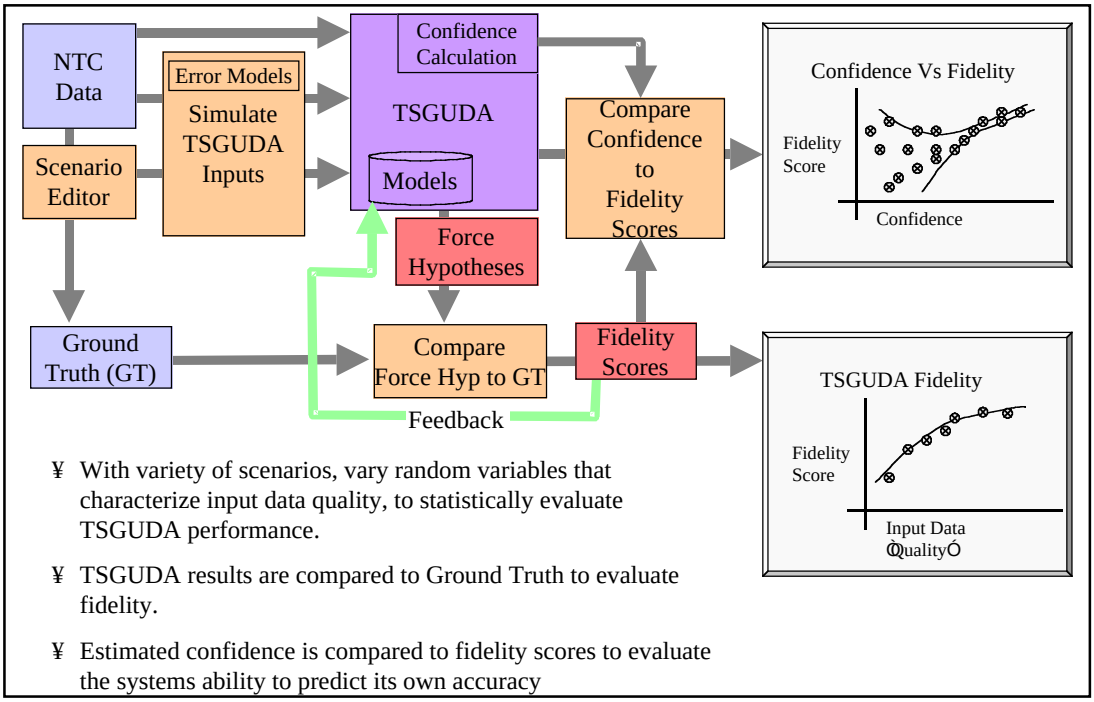

Figure 6. Experimental architecture.

The simulated ground truth, with simulated error models applied, is then input to TSGUDA, which generates a situation estimate in the form of force hypotheses. These force hypotheses are

compared to the original ground truth force hypotheses to generate a fidelity score. The fidelity scoring process is described in section 3.2 below. The results of the fidelity score can be used in a feedback process to tune the models, suggestor logic, and hypothesis management logic.

It is understood that a fidelity measure can only be used when ground truth is available, so will be of no use in a real military situation where a TSGUDA like situation estimation capability is needed. Figure 6 also shows a confidence calculation. The confidence measure is a metric developed by the system based on the quantity and believed quality of the input data, consistency between the available evidence, and the fit to existing models. The confidence measure will provide an estimate of the quality of the situation estimate independent of ground truth. The theory for the confidence calculation has been developed (Mahoney, et al, 2000), but it has not been implemented.

3.2. MEASURING SITUATION ESTIMATE RESULTS

In previous work (Mahoney, et al, 2000) we demonstrated the capability to measure the fidelity of a situation estimate to ground truth. However, the previous experiments evaluated only a single level of a force hierarchy. This section describes the extensions to the fidelity calculation to handle hierarchical groups, units, and activities.

The elements to be evaluated include:

1) Hypotheses about entities of interest and their attributes;
2) An indication of which reports and/or lower level situation elements are associated with each hypothesized entity of interest;
3) Inferences about features of the entities of interest. Relevant features may include:
    - Type of entity (e.g., maneuver company, engineer platoon);
    - Location of entity;
    - Composition of entity (e.g., number of elements of each type);
    - Activity of entity.

Evaluating the quality of a situation estimate in comparison with a ground truth scenario requires first associating situation hypotheses with the ground truth elements that gave rise to them and then evaluating how well the situation hypotheses represent the ground truth elements that they represent. In particular, a situation estimate is a faithful representation of a ground truth scenario to the extent that:

1) Most ground truth elements are associated with situation hypotheses (there are few *missed detections*);
2) Most situation hypotheses have exactly one ground truth element associated with them (there are few *false alarms*);
3) The features of interest (e.g., type, location, composition, activity) of the ground truth element are faithfully represented in the situation hypothesis.

Because there is uncertainty associated with a situation estimate, we do not expect an exact match with ground truth. A *scoring rule* is used to evaluate the degree to which a situation estimate meets the above criteria. The scoring rule is applied to *legal matches* of situation estimates to

ground truth elements. A legal match $\mu = (M, G_u, H_u)$ consists of a set M of *matched pairs*, a set G_u of *unmatched ground truth elements*, and a set H_u of *unmatched hypotheses*. Each matched pair $m=(g,h)\in M$ consists of a ground truth element and a situation hypothesis. Each ground truth element in the situation is listed exactly once either as a member of a matched pair or as an unmatched ground truth element. Each hypothesis in the situation estimate is listed exactly once as either a member of a matched pair or as an unmatched hypothesis. Each legal match μ is scored as follows:

- Each unmatched ground truth element $g\in G_u$ is assigned a *missed detect loss* λ_{md}.
- Each unmatched hypothesis $h\in H_u$ is assigned a *false alarm loss* λ_{fa}.
- Each matched ground truth element/hypothesis pair $m=(g,h)\in M$ receives a *location loss* equal to

$$\lambda_{loc}(m) = w_{loc}((x_g-x_h)^2-(y_g-y_h)^2),$$

  where w_l is a weighting factor for location, (x_g,y_g) is the location of the ground truth element in UTM coordinates, and (x_h,y_h) is the location of the hypothesized element in UTM coordinates. For vehicles the actual or reported vehicle location is used; for composite elements, the centroid is used.

- Each matched ground truth element/hypothesis pair $m=(g,h)$ receives a *type loss* equal to $\lambda_{typ}(m)$

$$\lambda_{typ}(m) = \sum_t \lambda(t,g)\pi_h(t)$$

  where $\lambda(t,g)$ is a loss associated with classifying an entity of the type of ground truth element g as type t, and π_{th} is the probability assigned by the situation hypothesis to type t. The loss $\lambda(t,g)$ is assumed to be zero if t is the correct type of ground truth element g, and positive for types not equal to the type of g.

- The total loss for the legal match is then given by:

$$\lambda_{tot}(\mu) = n_{md}\lambda_{md} + n_{fa}\lambda_{fa} + \sum_m \lambda_{loc}(m) + \lambda_{typ}(m)$$

  where m ranges over the matched ground truth element / hypothesis pairs, n_{md} is the total number of unmatched ground truth elements, and n_{fa} is the total number of unmatched hypotheses.

In our experiments we did not score situations with respect to activity estimates, although adding this capability is a straightforward extension to our method.

The approach described above scores legal matches of situation hypotheses to ground truth elements. However, there typically exist multiple ways to assign ground truth elements to hypotheses. We could simply score situations by computing the score of the highest-scoring legal match, but this would overstate the situation quality by ignoring uncertainty over matches of situations to ground truth elements. Moreover, finding the best match may be computationally very demanding for large scenarios. Instead, we used a sampling approach called Metropolis-Hastings sampling. Metropolis-Hastings sampling has its origin in statistical physics and has been widely applied to difficult problems in statistics and optimization. The basic assumption is

that lower scoring matches are more probable than higher scoring matches. We assume that the probability of a legal match is given by the Boltzmann distribution:

$$\pi(\mu)\frac{1}{Z}\exp\{-\lambda_{tot}(\mu)/\beta\},$$

where Z is a normalization constant ensuring that the probabilities add to 1, and β is a tuning parameter which (given appropriately normalized units) is analogous to temperature in physical systems. When β is near zero, only scores $\lambda_{tot}(\mu)$ very near the minimum score have any appreciable probability. As β increases, matches with higher loss score have an increasing probability of being accepted. We choose a value of β heuristically, seeking to prevent the algorithm from becoming "stuck" in a local minimum but not to proliferate very poor matches.

We sample from the Boltzmann distribution $\pi(\mu)$ using the Metropolis-Hastings algorithm:

- Begin with $M=\varnothing$ and G_u and H_u equal to the set of all ground truth elements and hypotheses, respectively. Call this legal match μ_0. Let its score be denoted by $\lambda_{tot}(\mu_0)$.
- Given the current legal match μ_i randomly perform one of the following operations:
  - Add a matched pair (g,h) to M, where $g\in G_u$ and $h\in H_u$;
  - For two pairs $(g,h)\in M$ and $(g',h')\in M$, switch hypotheses to obtain (g,h') and (g',h);
  - Remove a matched pair (g,h) from M, add g to G_u and h to H_u.

  In sampling, we apply a distance filter to ensure that extremely improbable match pairs are not added to M.
- Step 2) results in a candidate new legal match μ'. Decide whether to accept the new match according to the probabilistic acceptance rule:
  - If $\lambda_{tot}(\mu')> \lambda_{tot}(\mu_i)$, set $\mu_{i+1} = \mu'$
  - Otherwise, set $\mu_{i+1} = \mu'$ with probability p and $\mu_{i+1} = \mu_i$ with probability 1-p, where
  - $p=\exp\{\lambda_{tot}(\mu_i)-\lambda_{tot}(\mu')\}$.

It is well known (e.g., Gilks, et al., 1996) that this rule produces a sample from the distribution $\pi(\mu)$. We applied the standard technique of discarding a "burn in" sample and then keeping only every n^{th} observation in order to achieve approximately uncorrelated samples. We achieved good results from sampling every 60^{th} observation and discarding the first two batches of 60 observations.

The algorithm generates numbers that are samples from the probability distribution that is our estimate of the loss (in utility) that a user would experience by using this situation estimate. The actual loss depends on user selected values in the loss matrices that define the losses for the "errors" of false alarms, missed detections, miss-classifications, location errors, etc. for each level of the hierarchy. It is possible to equally weight all "errors" in order to produce a user and application independent fidelity score. It is also possible for the user who has a specific interest in particular unit types, or unit activities, to weight the loss matrices to more heavily penalize certain errors. For example, if the user is primarily interested in armor units, the loss for misclassification, missed detection, and false alarm of an armor unit should be much higher than for other types of units. These loss values have no effect on the operation of TSGUDA, they are only used for evaluating the situation estimates generated by TSGUDA.

3.2.1. EXAMPLE

An example that illustrates the application of the fidelity scoring is shown in Figure 3. The ground truth is shown as a hierarchy consisting of a Armor Heavy Company Team, consisting of a HQs Plt, two Armor Platoons, and a Mechanized Infantry Platoon. There is also a separate Engineer Platoon. Two example force level hypotheses are also shown. In the first, all platoons

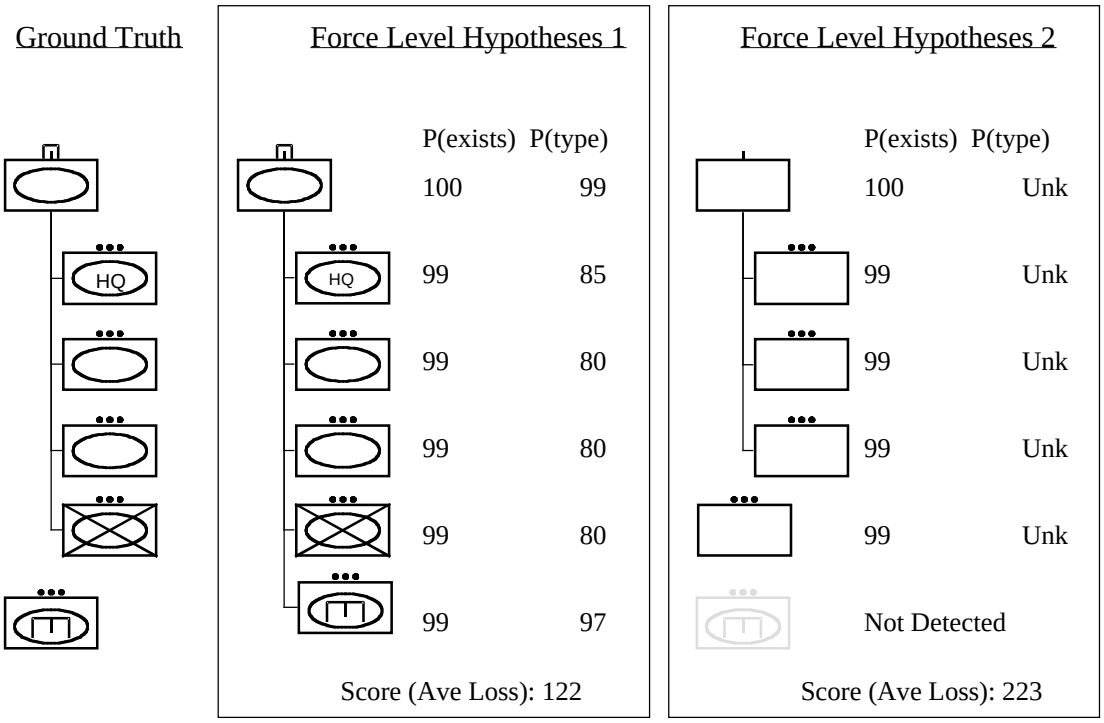

Figure 7. Ground truth force level hypotheses with two example situation hypotheses.

have been detected and identified with high probabilities. The differences between the ground truth and the hypotheses is in the hypothesized membership of the Engineer Platoon in the Company, and in the vehicles hypothesized as members of the Engineer Platoon (not shown). The score (average loss from the fidelity scoring algorithm) is 122.

The second force level hypotheses contains a company of unknown type, consisting of three platoons of unknown type, and one additional platoon. The separate Engineer platoon was not detected in this example. The score (average loss from the fidelity scoring algorithm) for this example is 223.

4. RESULTS

We have performed hundreds of experiments over multiple scenarios. This section provides representative results from one set of experiments on one scenario that is representative of the results from the other scenarios.

These results all use the same loss matrices in the fidelity scoring function. The loss values were chosen to more severely penalize errors (miss detections, false alarms, and miss classifications) in

armor units. Once the loss values are defined, it is the changes in average total loss that indicates the quality of a given situation estimate.

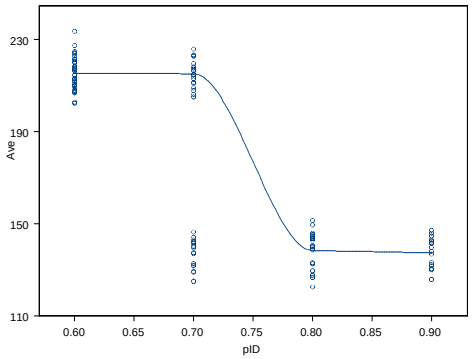

Figure 8. Fidelity (Ave Loss) as a function of pID.

4.1. FIDELITY OF SITUATION ESTIMATES

The simulated scenario involved a US armor heavy Company Team and a separate Engineer Platoon. The Armor Heavy Company Team consisted of a Tank Company HQs, 2 Tank Platoons, and a Mechanized Infantry Platoon. The scenario involved Engineer activity to clear obstacles and then the leapfrog movement of the Tank Heavy Company Team forward for an attack at an area of the National Training Center (NTC). From this scenario, ground truth vehicle tracks were generated. The experiment consisted of repeated runs of the TSGUDA input simulator (with error models), generation of a situation estimate by TSGUDA, and scoring of the situation estimate by comparison to the original ground

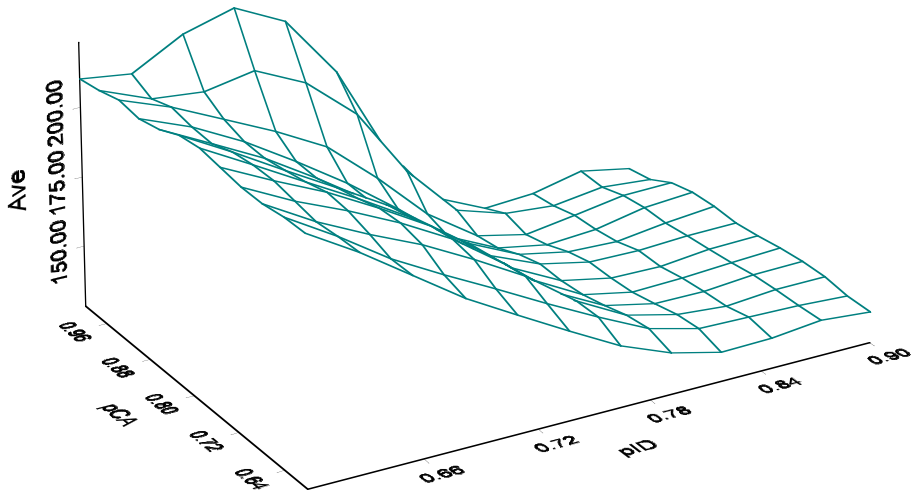

Figure 9. Average Loss as a function of p(ID) and p(CA)

truth. In this experiment, 146 situation estimates were generated from simulated data generated from the following variations in error model parameters:

Parameter	Variation	Description
P(ID)	1.0 - .6	Probability of correct ID
P(CA)	1.0 - .6	Probability of Correct Association (from one time step to the next)
D(CA)	100 — 40	Distance for Correct Association (only tracks separated by less than this distance may be incorrectly associated)

Figure 8 is a plot of the fidelity score (average loss) against pID. Figure 9 shows the response surface for the fidelity score (average loss) as a function of p(ID) and p(CA). The results here are similar to the results from last year: the quality of the situation estimate is better when p(ID) is high, and is relatively insensitive to p(CA). The insensitivity to the p(CA) depends on the distance over which missassociation errors can occur. In this scenario, the platoons of the company were usually separated far enough apart that any missassociation that did occur, still occurred within the same hypothesized platoon.

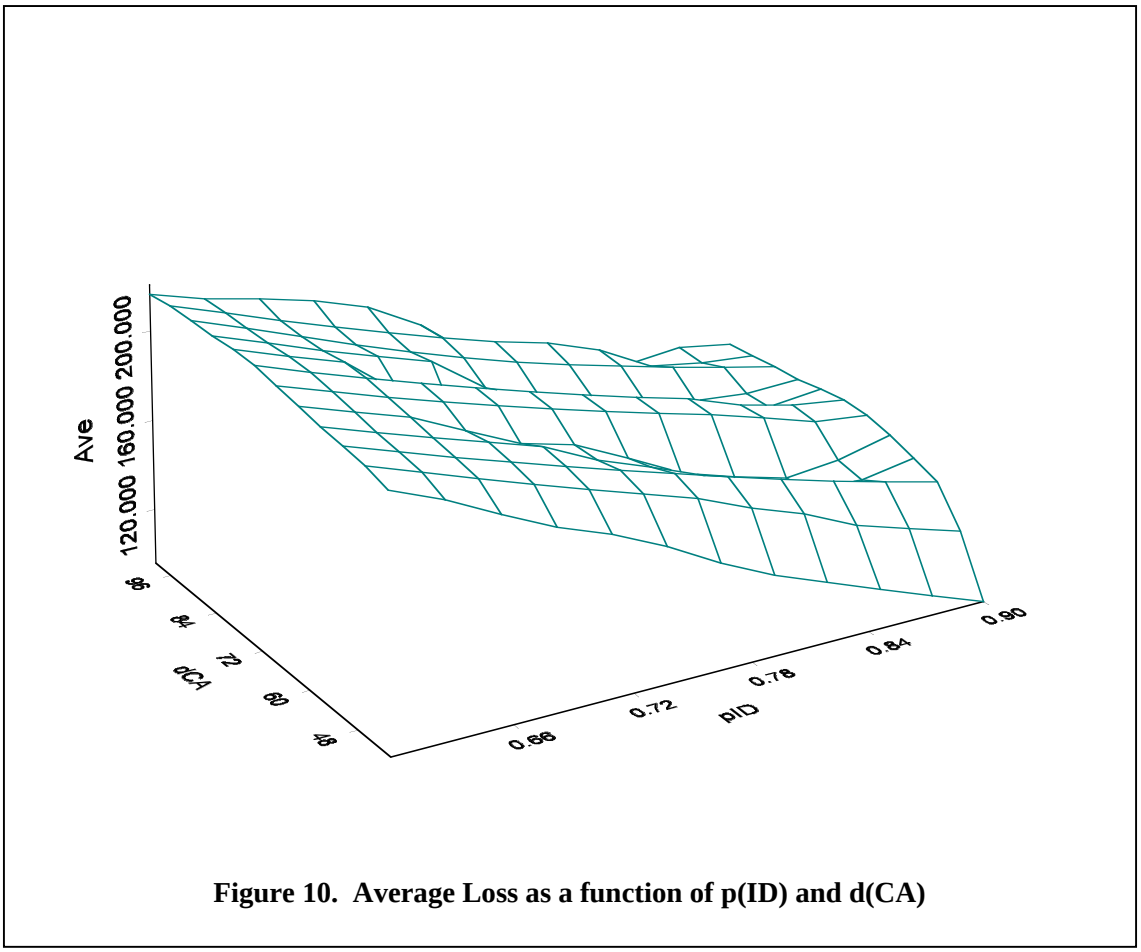

Figure 10. Average Loss as a function of p(ID) and d(CA)

Figure 10 shows the response surface for the fidelity score (average loss) as a function of p(ID) and d(CA). This figure again shows the relationship between p(ID) and the fidelity (average

loss), but also shows a smaller dependence between the fidelity and the correct association distance, dCA. If we assume that the distance over which missassociation errors occur, depends on the DDB system registration accuracy, then this improvement in the situation estimate based on decreasing the missasociation distance, is evidence that improvements in registration accuracy will improve the situation estimate.

Figure 10. shows another view of the relationship between these variables and the situation estimate fidelity score. The four response surfaces are generated for different values of pCA. It shows that for higher values of pCA (upper right graph) the fidelity is relatively insensitive to dCA. For smaller values of pCA there is an improvement in fidelity as dCA decreases, and the improvement from increasing pID has greater effect.

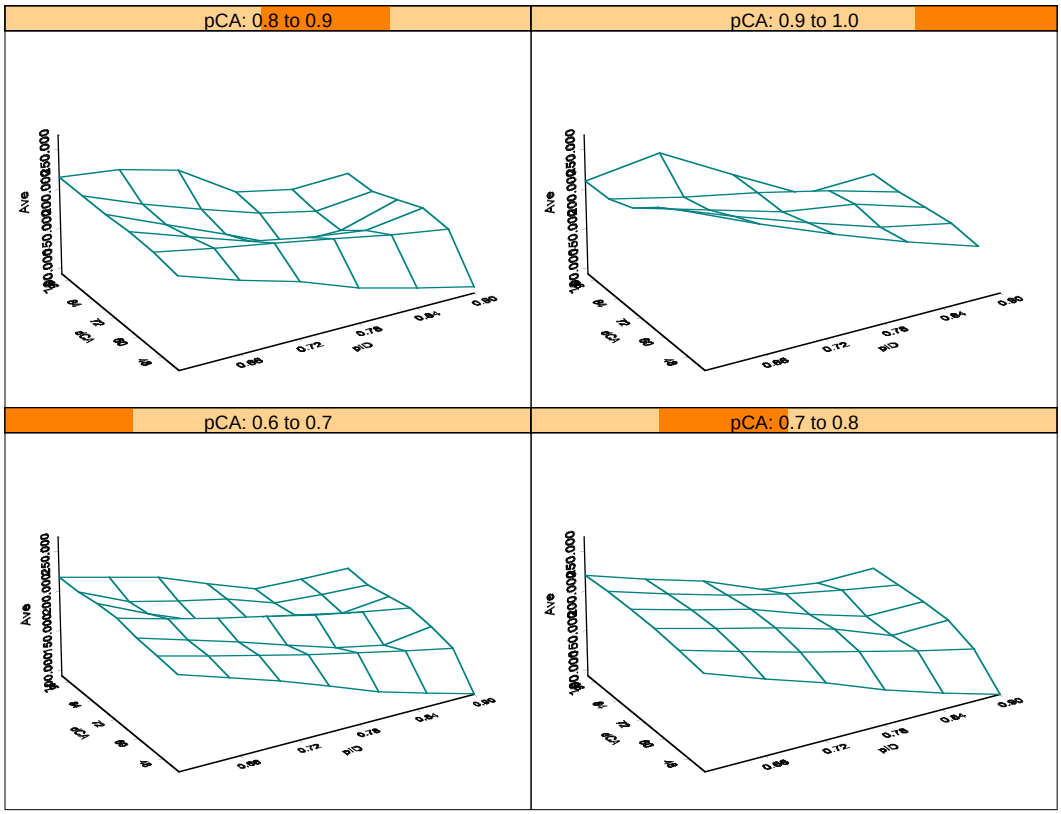

Figure 11. Response surfaces for Fidelity (Ave Loss) as a function of pID and dCA for different values of pCA.

In all these examples, fidelity score behaves in a reasonable way in response to variations in the quality of input track data.

4.2. COMPUTATIONAL COMPLEXITY

The extension of the knowledge domain models to reason about hierarchical groups, units and activities did result in more complex situation specific Bayesian Networks. We were able to accommodate the additional complexity in the models used for the relatively small scenarios

exercised in the experiments. We are investigating additional modeling choices and computational techniques that will allow us to extend the capabilities provided by these models to much larger scenarios.

5. CONCLUSIONS

The experiments described here demonstrate that TSGUDA is capable of robustly identifying groups, units, and activities hierarchies from the cooperative activities of members of a group. In addition, the extended fidelity metric provides a way to quantify the quality of the situation estimates provided by TSGUDA. This result builds on previous work, and shows that it is possible to extend situation inferences up the group / unit hierarchy.

The ability to measure situation estimates provides feedback to the domain knowledge modelers and developers to improve models and algorithms in TSGUDA. This is particularly important as the knowledge models become more complex, in order to model more complex unit and activity hierarchies. It is now feasible to envision an automated feedback loop that could automatically learn optimum models from a set of scenarios.

The capability to measure the situation estimate also provides a capability to develop an empirical model of the TSGUDA performance, that could be used in the DDB system model, and that could be used in tradeoff analysis to determine how to allocate resources between TSGUDA and other components of DDB, in order to optimize the overall situation estimate.

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 DÕAmbrosio, Masami Takikawa, and Dan Upper. We also thank Tod Levitt of IET for technical guidance. Finally, we thank Otto Kessler of DARPA for providing the necessary vision to challenge us to developing and extending a metric for measuring the quality of a situation estimate.

6. REFERENCES

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

Gilks, W. R., S. Richardson, and Spiegelhalter, D. (eds) (1996). *Markov Chain Monte Carlo in Practice*. London: Chapman & Hall.

Jensen, F.V., 1997, An Introduction to Bayesian Networks, Springer-Verlag, NY

Laskey, K. B. and S. M. Mahoney (1997) Network Fragments: Representing Knowledge for Constructing Probabilistic Models. In Geiger, D. and Shenoy, P. (eds) *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, San Francisco, CA: Morgan Kaufmann.

Laskey, Kathryn Blackmond, Suzanne M. Mahoney and Ed Wright, (2001) Hypothesis Management in Situation-Specific Network Construction. To appear in *Uncertainty in Artificial Intelligence, Proceedings of the Seventeenth Conference.* San Francisco, CA, Morgan Kaufmann

Mahoney S.M., Laskey K.B., Wright E., and Ng K.C., (2000) ÒMeasuring Performance for Situation Assessment,Ó*Proc. 2000 MSS National Symposium on Sensor and Data Fusion*, San Antonio, TX, June 2000.

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

Neapolitan, R. E. (1990) Probabilistic Reasoning in Expert Systems: Theory and Algorithms, John Wiley and Sons, Inc., New York.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA.