Network-based Truth Maintenance System

Subrata Das and **David Lawless**[1]

Abstract

We present a Network-based Truth Maintenance System (NTMS) for problem solvers based on Bayesian belief network (BN) technology. BN technology has been proven to be effective in various domains, e.g. assessing battlefield situations, such as the enemy's likely point of interdiction. Nodes and links in a BN capture semantic relationships among various domain related concepts. In the absence of firmer knowledge, default assumptions provide the beliefs of some nodes in a BN. Before posting incoming evidence into a BN node, a truth maintenance procedure is invoked to check for information consistency between the node's current expected state and the new observed state. In case of inconsistency, the truth maintenance procedure revises some default assumptions, by isolating those nodes causing inconsistency, via a sensitivity analysis procedure that exploits the strengths of BN causal dependency. We have applied our approach for trustworthy situation assessment in the context of a military Stability and Support Operation (SASO) scenario.

1 INTRODUCTION

We begin by presenting a small example that shows how inconsistency detection can potentially help make a trustworthy situation assessment, and the valuable contribution a suitable Truth Maintenance System (TMS) (Doyle, 1979; Forbus and de Kleer, 1993) can make in that context.

Suppose a knowledge base captures a simple causal rule between sensors and objects, which states that the presence of an enemy vehicle at a particular location will cause generation of signals from various sensors placed at that location. Now suppose that such a signal is received from a sensor covering the particular area. In the absence of any other knowledge, we assume by default that our sensors are functional. The inference engine concludes that an enemy vehicle is present in the area and adds this information to the active knowledge base. Subsequently, more reliable information is received, in the form of images of the area, which show conclusively there are no enemy vehicles in the area. This information would then be added to the knowledge base. Now, however, the added report is inconsistent with the earlier report suggesting the presence of an enemy vehicle. Consequently, it will be necessary to revise some of the earlier beliefs. The truth maintenance procedure should identify the incorrect default assumption that the sensor is functional and revise this assumption in the active knowledge base, preventing any further use of the evidence produced by the faulty sensor.

Our proposed Network-based Truth Maintenance System (NTMS) is specifically geared for any problem solver based on Bayesian belief network (BN) technology (Pearl, 1986; Lauritzen and Spiegelhalter, 1988). BN technology has been proven to be very effective, and we have applied it in a variety of domains for decision aiding, including battlefield situation assessment (Das et al, 2002), and spacecraft health determination (Das and Grecu, 2000), as it possesses a variety of theoretical and practical advantages relative to other approaches in dealing with the issues of uncertain inferencing, computational tractability, and causal and diagnostic reasoning. The nodes of a BN denote the variables representing concepts such as vehicle, sensor, and report, and the links denote causal relationships between the variables. The three-step procedure for our approach to truth maintenance consists of inconsistency detection via distance measure, inconsistency isolation via network sensitivity analysis, and inconsistency recovery via the adjustment of default assumptions.

Researchers have proposed other types of truth maintenance systems (TMS) over the years (Forbus and de Kleer, 1993): 1) Belief Maintenance Systems (BMS) (Falkenhainer, 1988; Ramoni, 1994); 2) Justification-Based Truth Maintenance Systems (JTMS); 3) Assumption-Based Truth Maintenance Systems (ATMS) (de Kleer, 1986); 4) Logic-Based Truth Maintenance Systems (LTMS). But our proposed NTMS offers several advantages for its 1) graphical representation that is more general than simple sentences in Boolean logic; 2) computational tractability of evidence propagation which avoids logical theorem proving; and 3) generalized reasoning which supports both deductive and abductive reasoning.

The rest of the paper is organized as follows. Some background in BN technology and network sensitivity analysis is presented in Section 2. The proposed truth maintenance approach is discussed in Section 3, including an application to trustworthy situation assessment in the context of an implemented SASO scenario. We conclude in Section 4 with some remarks on our future research and development plans.

2 BACKGROUND

2.1 Bayesian Belief Networks (BN)

A Bayesian belief network (Pearl, 1988; Jensen, 1996) is a graphical, probabilistic knowledge representation of a collection of variables describing some domain. The nodes of the belief network denote the variables and the links denote causal relationships between the variables. The topology encodes the *qualitative* knowledge about the domain. Conditional probability tables (CPTs) encode the *quantitative* details (strengths) of the causal relationships. The belief network of Figure 1 encodes the relationships over the domain consisting of the binary variables, *Injury, Rain*, *Game, Transport*, *Electricity*, and *Commentary*; its topology captures the commonsense knowledge that:

1. *Rain* causes *Transport* disruption
2. *Rain* causes *Electricity* failure

[1] Charles River Analytics, Inc., 625 Mount Auburn St., Cambridge, MA 02138, USA. Email: {sdas, dlawless}@cra.com

3. *Game* causes running *Commentary* on the radio
4. *Injury* and *Rain* prevent *Game* from being played

As shown in Figure 1, the CPT specifies the probability of each possible value of the child variable conditioned on each possible combination of parent variable values. For example, the probability of getting sunburn given that clouds are present is 0.1, whereas the probability of getting sunburn given clear skies is 0.7.

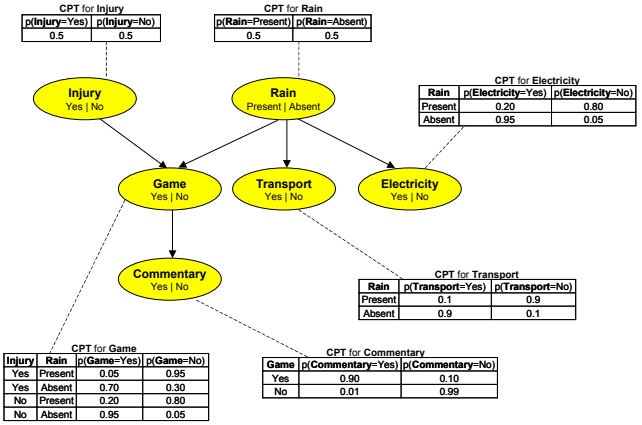

Figure 1: Simple Bayesian Belief Network (BN)

The structure of a belief network encodes other information as well. Specifically, the lack of links between certain variables represents a lack of direct causal influence, that is, they indicate conditional independence relations. This belief network encodes many independence relations, for example,

1. $Electricity \perp Transport \mid Rain$
2. $Commentary \perp \{ Rain, Electricity \} \mid Game$

where '$\perp$' is read 'is independent of' and '$\mid$' is read 'given.' Once the value of *Rain* is known, the value of *Transport* adds no further information about *Electricity*. Similar conditional independence assertions hold for other variables.

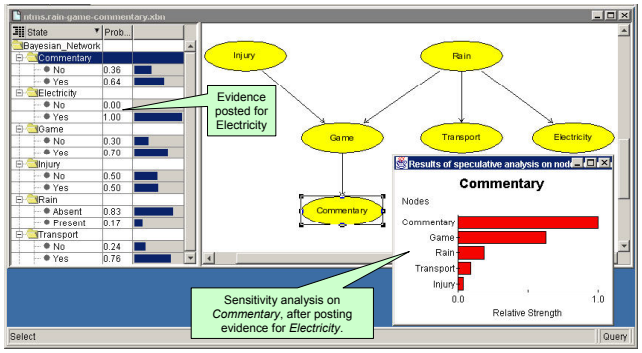

Figure 2: Software model for Figure 1

Figure 2 shows an implementation of the network in Figure 1 on in-house software. The panel on the right hand side (of the larger child window) is the actual network of Figure 1, and the panel on the left hand side shows the current beliefs of the nodes. Evidence for *Electricity* has been posted to the network.

When new evidence is posted to a variable in a belief network, that variable updates its own belief vector and then sends out messages indicating updated predictive and diagnostic support vectors to its children and parent nodes respectively. The messages are used by the other nodes, which update their own belief vectors, and also propagate their own updated support vectors. In the case of polytrees, the separation of evidence yields a propagation algorithm (Pearl, 1988) in which update messages need only be passed in one direction between any two nodes after the posting of evidence. The algorithm has been extended to the more general case of directed acyclic graphs (DAGs), e.g. see (Jensen, 1996).

2.2 Network Sensitivity Analysis

To isolate groups of nodes that may be causing inconsistency, we apply a technique known as *sensitivity analysis* to belief networks. Sensitivity analysis (Jensen, 1996) helps determine which evidence is most relevant to the state of a particular network node. Our approach is as follows. Suppose we have a set of battlefield reports, constituting evidence *e*, which has already been posted to the appropriate nodes of a BN, for assessing a battlefield situation. Now, incoming evidence causes inconsistency at a particular node, and we need to determine which subsets of evidence *e* are relevant to, and support or contradict, the state of the inconsistent node. In other words, we find those evidentiary nodes that have the most effect upon the state of the inconsistent node.

To illustrate the concept of sensitivity analysis on the BN in Figure 1, we have a set *e* of evidence, consisting of *eI*, *eT*, and *eE*, representing respectively no injury, no transport disruption, and the presence of electricity. Symbolically, we have:

eI: $Injury = No$, $\neg eI$: $Injury = Yes$
eT: $Transport = Yes$, $\neg eT$: $Transport = No$
eE: $Electricity = Yes$, $\neg eE$: $Electricity = No$

Initially, with no evidence posted, we have:

$p(Commentary = Yes) = 0.43$

When we post the full set *e*, the network shows that

$p(Commentary = Yes \mid eI, eT, eE) = 0.84$

suggesting a high likelihood of finding a running radio commentary on the game. But actual report suggests no radio commentary at all, i.e. we receive evidence indicating a low likelihood of commentary, e.g. we might receive $p(Commentary = Yes) = 0.2$. We therefore have a serious discrepancy between the network's predicted value (0.84) and the actual value of *Commentary* (0.2).

Using sensitivity analysis, we will try to find a minimal subset of the evidence causing the discrepancy, and therefore to be retracted, to bring the state of the node *Commentary* to neutral. This simple approach lets us optimize over a state space of just 2**3 = 8 subsets.

One possibility is to focus on all evidence, and retract it; this results in the probability noted above, i.e.:

$p(Commentary = Yes) = 0.43$ or
$p(Commentary = No) = 0.57$

Another possibility is to retract none of our evidence; this was also calculated above, i.e.:

$p(Commentary = Yes \mid eI, eT, eE) = 0.84$ or
$p(Commentary = No \mid eI, eT, eE) = 0.16$

The sensitivity results of retracting one piece of evidence at a time are the following:

$p(Commentary = No \mid eI, eT) = 0.21$ (retracting *eE*)
$p(Commentary = No \mid eI, eE) = 0.26$ (retracting *eT*)
$p(Commentary = No \mid eT, eE) = 0.27$ (retracting *eI*)

Therefore, none of the above three yields a neutral state for the node *Commentary*. Retracting two pieces of evidence at a time gives the following sensitivity results:

$p(Commentary = No \mid eI) = 0.48$ (retracting *eT*, *eE*)

p(*Commentary* = *No* | *eT*) = 0.32 (retracting *eI*, *eE*)
p(*Commentary* = *No* | *eE*) = 0.36 (retracting *eI*, *eT*)

Therefore, after checking all possibilities, the sensitivity analysis identifies the evidence subset {*eT*, *eE*}; retracting this subset brings the state of *Commentary* to almost neutral.

This kind of search for the right subset that accounts for almost all the change is the central theme of sensitivity analysis. Note that not only can we retract our evidence, it can also be further revised. For example, after retracting *eE*, one could postulate a new value for electricity, e.g. *Electricity* = *No*. This would result as p(*Commentary* = *No* | *eI*, *eT*, ¬*eE*) = 0.58, bringing the node commentary close to a neutral state. We may also consider using finer granularity in our adjustments, at the expense of further enlarging the search space.

Formally, the measure of sensitivity analysis at a node is the variance of its belief, that is, expected change squared of the beliefs of the node, taken over all of its states, due to a finding at other nodes (Jensen, 1996). To illustrate our in-house implementation of sensitivity analysis, Figure 2 also shows the results of the sensitivity analysis carried out on the network, where *Commentary* is the query variable, and *Game*, *Rain*, *Injury*, and *Transport* are the finding variables.

Note that sensitivity analysis is highly context dependent, in that its results vary significantly with the evidence that has been posted. Moreover, evidence in which we are more confident (e.g. p(*X*=*Yes*)=0.9 for a binary variable *X*) has less effect upon the target node, hence will show up near the bottom of the list.

2.3 Truth Maintenance via Sensitivity Analysis

A belief network helps make predictions and decisions based on available observations and some default assumptions. For example, in the context of the example network of Figure 1, one can predict the status of the game based on observations such as transportation status, player injury, electricity supply, and so on. All the desired evidence may not be available when one needs to make predictions and decisions, therefore one must make some default assumptions. The *a priori* probabilities of some variable states can guide construction of the default assumption set.

For example, usually there is no player injury and one can assume the probability distribution (*Yes* = 0.1, *No* = 0.9) for *Injury* as a default. Note that unlike a traditional logic-based truth maintenance system (Reiter, 1980), default assumptions in our environment are probabilistic. With the default assumptions for *Injury* and *Rain* as (Yes = 0.1, No = 0.9), and with real evidence indicating certainty of the electricity supply (Yes = 1.0, No = 0.0), the network infers the probability distribution of *Game* is (Yes = 0.91, No = 0.09). Therefore, one can predict that the game is almost certainly going to be held. Consequently, there is a high likelihood of running commentary, which is reflected in the probability distribution (*Yes* = 0.82, *No* = 0.18) of *Commentary*.

Suppose, however, that no radio commentary is heard during the scheduled time of the game. This evidence is inconsistent with the current state of the node *Commentary* in the network, i.e. there is substantial difference (e.g. can be measured by means of Euclidean distance) between the current predicted state and the observed evidence. Instead of simply posting the contradictory evidence and letting the beliefs propagate as per usual, *we interpret this situation to mean that our default assumptions need revision*; we attempt to revise the set of defaults to be more consistent with the evidence observed. In the context of our example, i.e. we to revise the default assumptions for variables *Injury* and *Rain* so as to be more consistent with the fact that there is no running commentary.

We will use sensitivity analysis on the network with hypothesis *h* as "*Commentary* = *No*". We will try to make simple, minimal revisions to our default assumptions, which will bring the state of the node *Commentary* to a neutral level (almost equal distribution of the states Yes and No) suitable for propagating evidence. It is important to note that we are not trying to match the incoming value for *Commentary* at this point, i.e. we are not trying to 'explain' the discrepancy by adjusting our default assumptions to match reality. Rather, we are attempting only to revise the default assumptions so that the discrepancy is tolerable, and a neutral level of *Commentary* will suffice as it indicates that we merely don't know its current value. For simplicity, we will use low granularity in revising our defaults. Moreover, we want to make the minimal number of revisions to our default assumptions that will accomplish this, as we subscribe to the heuristic that simpler reasons for problems are more likely to be the correct reasons.

At this point the total evidence *e* posted to the network consists of the default assumptions for *Injury* and *Rain*, and the observed evidence for *Electricity*:

Injury: (*Yes* = 0.1, *No* = 0.9)
Rain: (*Present* = 0.1, *Absent* = 0.9)
Electricity: (*Yes* = 1.0, No = 0.0)

This gives us 2**2 = 4 options for revising the two default assumptions in *e*:

1. The default or "do nothing" option of not making any revisions. This we of course reject immediately because it's equivalent to accepting the inconsistent state.
2. e_1 : revise *Injury* to (*Yes* = 0.9, *No* = 0.1), i.e. a high probability of injury
3. e_2 : revise *Rain* to (*Present* = 0.9, *Absent* = 0.1), i.e. a high probability of rain
4. e_{12} : revise both *Injury* and *Rain* to (*Yes* = 0.9, *No* = 0.1), i.e. high probabilities of both injury and rain

In the process of revising our default assumptions we want to ensure that the revision is minimal. For example, the last option e_{12} subsumes both e_1 and e_2. Therefore, if we can bring down the probability distribution of the variable *Commentary* to a level consistent with the observed value by either e_1 or e_2 then we should not pursue e_{12}. We have the following results:

p(*Commentary* = *No* | e_1) = 0.36
p(*Commentary* = *No* | e_2) = 0.60
p(*Commentary* = *No* | e_{12}) = 0.73

Clearly, e_1 alone cannot bring the network to a consistent state, whereas e_2 is enough if the evidence of no running commentary is close to 0.6, and e_{12} may be appropriate for higher certainty of no commentary. Note the sensitivity analysis results shown in Figure 2 reflects the first two revision options that the node *Rain* has higher influence on the node *Commentary* than the node *Injury*.

Note we can generalize the algorithm to cases where the variables are not boolean, using our sensitivity analysis procedure to approximate a uniform belief distribution. We can also use finer granularity in our search (i.e. check more than just absolute belief levels) for more accurate results, at the expense of more computation time.

3 NTMS FOR SITUATION AWARENESS

3.1 Belief Networks for Situation Awareness

We have applied our truth maintenance approach to military scenarios, for maintaining consistent BN states so as to ensure trustworthy situation assessments, such as "Enemy will interdict at NAI 2". This kind of assessment is based on observation of low-level enemy activities such as communication and reconnaissance. Each belief network is constructed to assess a specific high-level situation in the form of the commander's priority intelligence requirement (PIR). Before posting incoming evidence at a belief network node, a truth maintenance procedure is invoked to detect information inconsistency between the node's current state and the state of the evidence to be posted. In the case of inconsistency, the truth maintenance procedure isolates the nodes that are causing inconsistency, based on the causal network dependency. The proposed NTMS thus incrementally maintains only consistent BN states.

BNs related to a SASO (Stability And Support Operation) scenario were constructed in a knowledge elicitation session with our subject matter experts. These belief networks were designed to answer the PIRs as described above. A portion of the BN used to answer the last PIR, related to the enemy's interdiction at a specified location by its special police force (SPF) and is shown below in Figure 3.

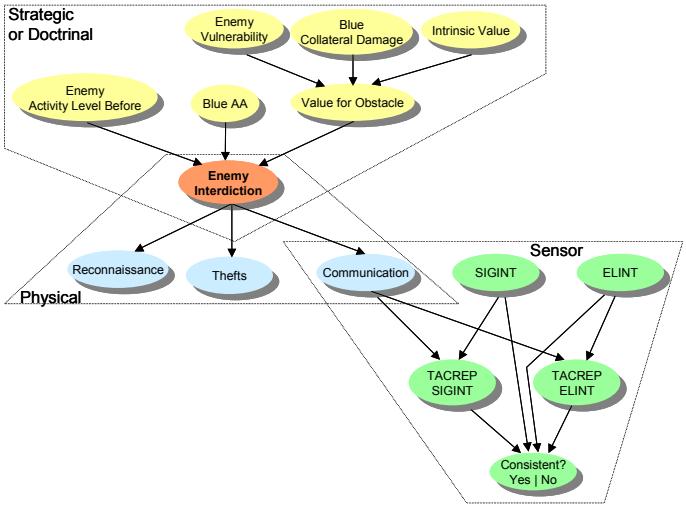

Figure 3: Belief net for Enemy Interdiction

The upper half of the BN of Figure 3 represents doctrinal knowledge. The nodes and links encode information allowing one to infer where the enemy would interdict according to its doctrine, which represents background knowledge that makes an event or a situation likely whenever a set of general criteria is satisfied. In effect, doctrinal knowledge does not include observations or intelligence reports about the site per se.

For example, a location considered by the enemy adequate to place an obstacle, and therefore provides high-value for the obstacle, if it has low vulnerability, high collateral damage, and high intrinsic value. This is encoded by the portion of the BN in Figure 3 that includes the top three nodes and their common child. Also, a location where enemy activity level has been observed earlier, and is a likely avenue of approach (AA) for the Blue forces, and provides a high value for an obstacle, is likely to be used for interdiction by the enemy. This is encoded by the portion of the BN in Figure 3 that includes the three nodes of the second row from the top and their common child.

The lower right part of the BN in Figure 3 is focused on observations or INTEL reports. Observations of reconnaissance, theft, and communication activities at a certain location are strong indications of a likely interdiction location. When new evidence is posted to the *Recon* node based on intelligence reports, the likelihood of enemy interdiction at that location will increase, which in turn will increase the likelihood of the enemy's theft and communication activities at that location.

Enemy activity is detectable by various types of sensors, including signal intelligence (SIGINT) and electronic intelligence (ELINT) sensors. Enemy activity at a location with a sensor produces a tactical report (TACREP). The portion of the BN labeled 'Sensor' in Figure 3 shows that both SIGINT and ELINT produce reports upon detection of the enemy's communication activities. If all sensors are functional, and the enemy is truly present, then reports should be produced by each sensor at the location. If the enemy is absent then no sensor should produce reports. The CPT associated with the node labeled 'Consistent?' is constructed to reflect this relationship between the two nodes TACREP SIGINT and TACREP ELINT, in a manner similar to user-defined integrity constraints in databases (Das, 1992). For example, high likelihood for a SIGINT report and low likelihood for an ELINT report together cause a possible inconsistency. Evidence sent by battlefield intelligence sources is propagated through the network of Figure 3 to answer the PIR.

3.2 Belief Revision and Truth Maintenance

We illustrate truth maintenance and consistency checking, focusing on the portion of the network in Figure 3 labeled 'Sensor'.

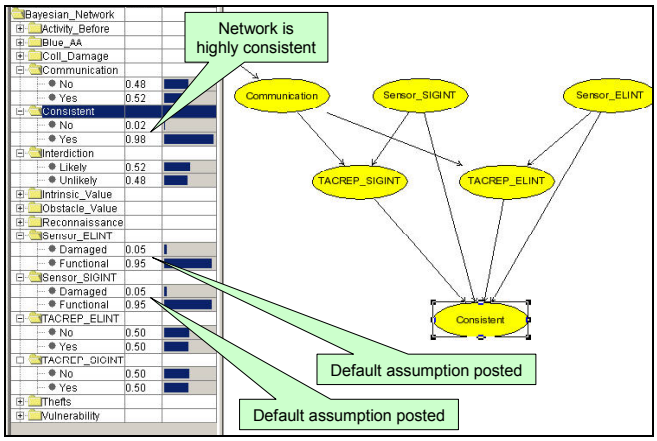

Figure 4: Network with sensor defaults posted

The SIGINT and ELINT sensors are by default assumed functional, so we post evidence:

Sensor_SIGINT: (*Functional* = 0.95, *Damaged* =0.05)

Sensor_ELINT: (*Functional* = 0.95, *Damaged* = 0.05)

Figure 4 shows the state of the network after the propagation of this evidence. Note the current belief state (*Yes*=0.98, *No*=0.02) of *Consistent* assures consistency amongst the sensor reports. Now suppose we receive a tactical report generated by the SIGINT

sensor, confirming enemy communication activity at the location. This results in the propagation of evidence:

TACREP_SIGINT: (*Yes*=0.99, *No*=0.01).

However, for whatever reason (it's late, or lost, or something else), we do not receive a tactical report from the ELINT sensor, resulting in the propagation of evidence

TACREP_ELINT: (*Yes*=0.01, *No*=0.99).

Figure 5 shows the resulting network state, a concern due to the changed belief state of *Consistent* from (*Yes*=0.98, *No*=0.02) to (*Yes*=0.79, *No*=0.21). Our goal now is to raise the belief of the state *Yes* of the node *Consistent* to a level above 0.9. We perform sensitivity analysis with respect to *Consistent*, to isolate nodes that have high influence on it.

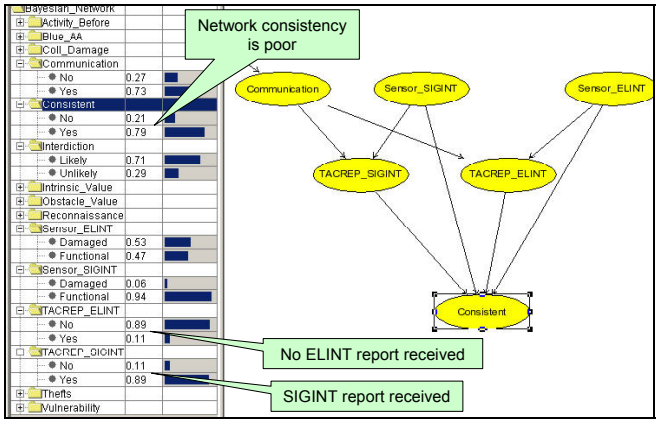

Figure 5: Network inconsistency on sensor evidence

Figure 6 shows sensitivity analysis on the node *Consistent*. Of the top three nodes, only the state of *Sensor_ELINT* was assumed by default, suggesting its revision.

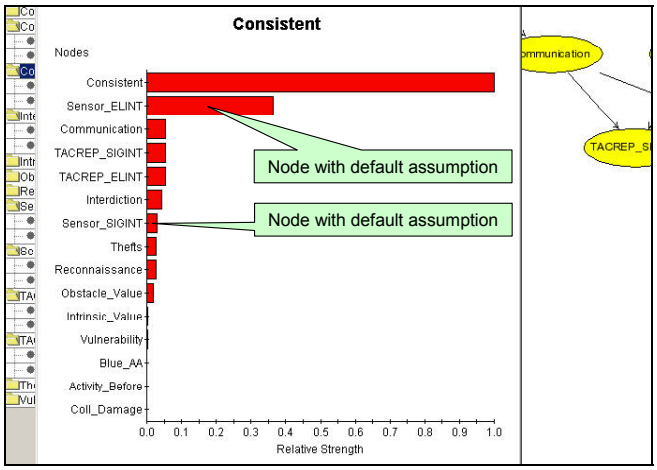

Figure 6: Sensitivity analysis for node *Consistent*

There are now two ways to revise the default assumptions:

1. The state of the ELINT sensor is unknown, so retract its default assumption, to get a belief state of:

   Sensor_ELINT: (*Functional* = 0.5, *Damaged* = 0.5)

2. The state of the ELINT sensor is damaged, so reverse the initial default assumption, i.e. post:

   Sensor_ELINT: (*Functional* = 0.1, *Damaged* = 0.9)

After revising the default assumption for the new belief that the sensor is damaged, the beliefs of *Consistent* become (*Yes*=0.98, *No*=0.02), suggesting excellent consistency in the network.

4 CONCLUSION

We have presented a probabilistic approach to truth maintenance, specifically for problem solvers based on BN technology. Knowledge base updates are translated into evidence and propagated into the networks for the purpose of detection, isolation, and recovery from inconsistency. The underlying truth maintenance procedure uses a BN sensitivity analysis exploiting strengths of the causal dependencies. We demonstrated our approach for situation assessment in the context of a SASO scenario.

Our follow-on work focuses on all three sub-areas of truth maintenance. For inconsistency detection, we are developing a more precise metric that is context sensitive. For isolation, we are developing an improved sensitivity analysis scheme that considers the influence of combinations of nodes on the target. Finally, for recovery, we are developing a minimal way of readjusting default assumptions before posting incoming evidence.

5 REFERENCES

[1]. Das, S. (1992). Deductive Databases and Logic Programming. Addison-Wesley.

[2]. Das, S., Gonsalves, P., and Grey, R., "Situation Assessment via Belief Networks", Proceedings of the Fifth International Conference on Information Fusion, Maryland, USA, 2002.

[3]. Das, S., and Grecu, D., "COGENT: Cognitive Agent to Amplify Human Perception and Cognition", Proceedings of the Fourth International Conference on Autonomous Agents, Barcelona, Spain, 2000.

[4]. De Kleer, J., "An Assumption-based TMS", Artificial Intelligence, 28:127-162, 1986.

[5]. Doyle, J., "Truth Maintenance Systems", Artificial Intelligence, 12(3): 231-272, 1979.

[6]. Falkenhainer, B., "Towards a General-Purpose Belief Maintenance System", Uncertainty in Artificial Intelligence 2. North-Holland, 1988.

[7]. Forbus, K. D. and de Kleer, J. (1993). "Building Problem Solvers", MIT Press.

[8]. Jensen, F.V. (1996). An Introduction to Bayesian Networks. Springer-Verlag.

[9]. Lauritzen, S. L. and D. J. Spiegelhalter (1988). "Local Computation with Probability in Graphical Structures and Applications to Expert Systems." *Journal of the Royal Statistical Society B* **50**(2).

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

[11]. Ramoni, M., "Belief Maintenance in Bayesian Networks", Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI94). Morgan-Kaufmann.

[12]. Reiter, R. (1980). "A logic for default reasoning". Artificial Intelligence, Vol. 13, pp. 81-132.