Australian Government
Department of Defence
Defence Science and
Technology Organisation

# Evaluation of the Effectiveness of Machine-based Situation Assessment – Preliminary Work

*David M. Lingard[1] and Dale A. Lambert[2]*

**[1] Intelligence, Surveillance & Reconnaissance Division**
**[2] Command, Control, Communications & Intelligence Division**
**Defence Science and Technology Organisation**

DSTO-TN-0836

## ABSTRACT

The Information Fusion Panel within The Technical Cooperation Program (TTCP) is developing algorithms to perform machine-based situation assessment to assist human operators in complex situations. This report proposes a technique to measure the effectiveness of these algorithms in a simulation environment where ground truth is well-defined. In addition, this report models the situation assessment algorithms abstractly using random inference networks, and examines how errors (damage) spread through the inference networks. This models deficiencies in the object assessment as input to the situation assessment algorithm.

**APPROVED FOR PUBLIC RELEASE**

*Published by*

*Intelligence, Surveillance & Reconnaissance Division*
*DSTO Defence Science and Technology Organisation*
*PO Box 1500*
*Edinburgh, South Australia 5111 Australia*

*Telephone: (08) 8259 5555*
*Fax: (08) 8259 6567*

*AR-014-253*
*August 2008*

# Evaluation of the Effectiveness of Machine-based Situation Assessment – Preliminary Work

## Executive Summary

The Information Fusion Panel within The Technical Cooperation Panel (TTCP) is developing algorithms for machine-based situation assessment to assist the human operator in complex situations. This report describes a technique that could be employed to evaluate the effectiveness of these algorithms in a simulation environment where the ground truth is well-defined. The Information Fusion Panel can demonstrate and test the algorithms by simulating information fusion in the fictional "North Atlantis" scenario.

This report contributes directly to fulfilment of a TTCP milestone.

In the TTCP experiments the ground truth and situation assessment will be described using the same ontological framework. However this report also briefly discusses the more general case where different ontological frameworks are employed for the ground truth and situation assessment.

The technique for evaluating the algorithms involves a process of aligning the situation assessment more closely to the ground truth over a number of iterations. Propositions where the situation assessment starts to diverge from the ground truth are identified, and then the technique tries to "prune" these branches of divergence. The technique utilises the F-value metric described in the scientific literature. The F-value is a set-theoretic measure that facilitates comparison of the output propositions from the situation assessment with those from the ground truth. The evaluation technique seeks to measure how quickly the F-value converges to its maximum value of unity over successive iterations, indicating that the situation assessment has converged to the ground truth.

During the comparison of the situation assessment and ground truth, the matching propositions from the two sets of output propositions are identified. This report discusses techniques for performing partial matching of output propositions from the situation assessment and ground truth, e.g. when they differ only in a single argument. It addresses partial matching for both cases where the differing argument is a discrete variable or a continuous variable.

Further work is required to test this evaluation technique prior to it being employed in TTCP experiments, especially to understand better the conditions under which the F-value converges to unity.

Errors in the input object assessment or the situation assessment algorithms can cause cascading errors through the inference networks. Complex networks have received attention recently in a diverse range of disciplines including the physical sciences, biological sciences, economics and sociology. Examples of complex networks are electricity grids, the Internet, neurons in the human brain, the global economy, and friendship networks. One issue studied has been that of damage spreading through complex networks, e.g. viruses spreading over the Internet.

This report describes how random inference networks were constructed to abstractly model the situation assessment process, and to examine how errors (damage) spread through such networks. The random inference networks consist of: (a) input propositions that model inputs to the situation assessment such as the object assessment, (b) inference rules, and (c) output propositions. This approach has similarity to the random Boolean networks described in the scientific literature that abstractly model gene regulation and control.

The F-value metric was used to measure how a single perturbation spread through a random inference network. The results achieved thus far are only very preliminary, but they show that damage spreading depends on:

(a) the degree of connectedness between the rules and other predicates in the network, and

(b) the type of Boolean functions that are employed in the network.

Further work is needed to produce a more complete set of results, and to leverage off the significant body of research into damage spreading in random Boolean networks.

# Contents

# Acronyms

| | |
|---|---|
| C3I | Command, Control , Communications and Intelligence |
| DSTO | Defence Science and Technology Organisation |
| GT | Ground Truth |
| ISRD | Intelligence, Surveillance and Reconnaissance Division |
| JDL | Joint Directors of Laboratories |
| JVC | Jonker-Volgenant-Castañon |
| NK | Synonym for Random Boolean Network |
| RBN | Random Boolean Network |
| SA | Situation Assessment |
| STAGE | Scenario Toolkit and Generation Environment |
| TTCP | The Technical Cooperation Panel |

# 1. Introduction

The most dominant model of data fusion is the Joint Directors of Laboratories (JDL) model [1, 2]. The three levels of data fusion from that model that are most relevant to this report are: [6, 7]

1. **Level 1. Object assessments** are stored representations of objects. They are usually partitioned into data registration, data association, position attribute estimation, and identification.
2. **Level 2. Situation assessments** are stored representations of relations between objects. Situation assessment fuses the kinematic and temporal characteristics of the data to create a description of the situation in terms of indications of warnings, plans of action, and inferences about the distribution of forces and flow of information.
3. **Level 3. Threat assessments** are stored representations of effects between objects. They assess the threat posed by the enemy being tracked. This may also include an assessment of the friendly forces' ability to engage the enemy effectively.

This report is primarily concerned with algorithms that are developed to perform machine-based situation assessments, thus assisting the human operators in complex military situations. In particular, this report is concerned with how to assess the effectiveness of these algorithms.

In relation to machine-based situation assessment, there needs to be a way to represent the domain of interest in a meaningful way. Accordingly, reference 3 posed the Semantic Challenge that is "What symbols should be used, and how do these symbols acquire meaning?" It is a very significant challenge to develop an **ontological framework** to achieve this goal. A framework named "Mephisto" is being developed (3, 4, 5). Its layers and a sample of the associated concepts are:
**Metaphysical Layer**: exist, time, connect, distance, angle.
**Environmental Layer**: land, sea, air, temperature, weight.
**Functional Layer**: sense, move, attack, destroy.
**Cognitive Layer**: achieve, intend, belief, expect, inform, prefer.
**Social Layer**: group, ally, enemy, possess, authorise.

Examples of the **relations** developed under the Mephisto framework are: (3, 4, 5)

1. **destroyed**(x) meaning that x is destroyed.
2. **can_transform**(z,x,y) meaning that z can transform x into y.
3. **desires**(X, $\alpha$) meaning that the individual X desires that $\alpha$ in order to satisfy an existing intention.

Reference 4 describes an ontology as the systematic specification of the concepts required to describe the domain of interest. It notes that ontological frameworks need to do at least two things:

1. Provide a formal language in which ontologies can be expressed.
2. Provide reasoning capabilities, so that an ontology can be demonstrated to be free of contradictions.

Reference 4 notes that the formal language utilised is normally a **logical language**, and that, in mathematical terms, the logic employed should be sound, complete, decidable and tractable if possible. Logic systems that might be employed include description logics. A **formal logic** is a formal language together with an inference relation that specifies which sentences of that formal language can be inferred from sets of sentences in that formal language. A **formal theory** is a set of sentences expressed in a formal language. When combined with a formal logic, inferences from a formal theory can be made that describe a domain of interest in the world.

The goal of the algorithm for machine-based situation assessment is to determine the most likely state of the domain of interest. For example, reference 6 develops a **modal model** for the state of the domain of interest that comprises a set of **possible worlds**. The maintenance of a probability density function over the possible worlds allows the most likely possible world to be determined.

# 2. TTCP Activity

The Technical Cooperation Panel (TTCP) provides a mechanism for collaboration in Defence science and technology for the participating countries: Australia, the United States, the United Kingdom, Canada and New Zealand. Technical Panel 1 in the Command, Control, Communications and Intelligence (C3I) Group of TTCP is named the Information Fusion Technical Panel. One of their key goals is to develop and test algorithms for machine-based situation assessment.

To achieve this end, they have developed a detailed “North Atlantis” scenario [8]. Figure 1 shows a map of North Atlantis, a fictional continent between Europe and Greenland. The scenario includes: six nations with alliances and hostilities, destroyers, frigates, mine vessels, patrol boats, submarines, merchant ships, whaling and counter whaling vessels, coast guard vessels, tourist vessels, commercial aircraft, surveillance aircraft, military strike aircraft, and military and civilian helicopters.

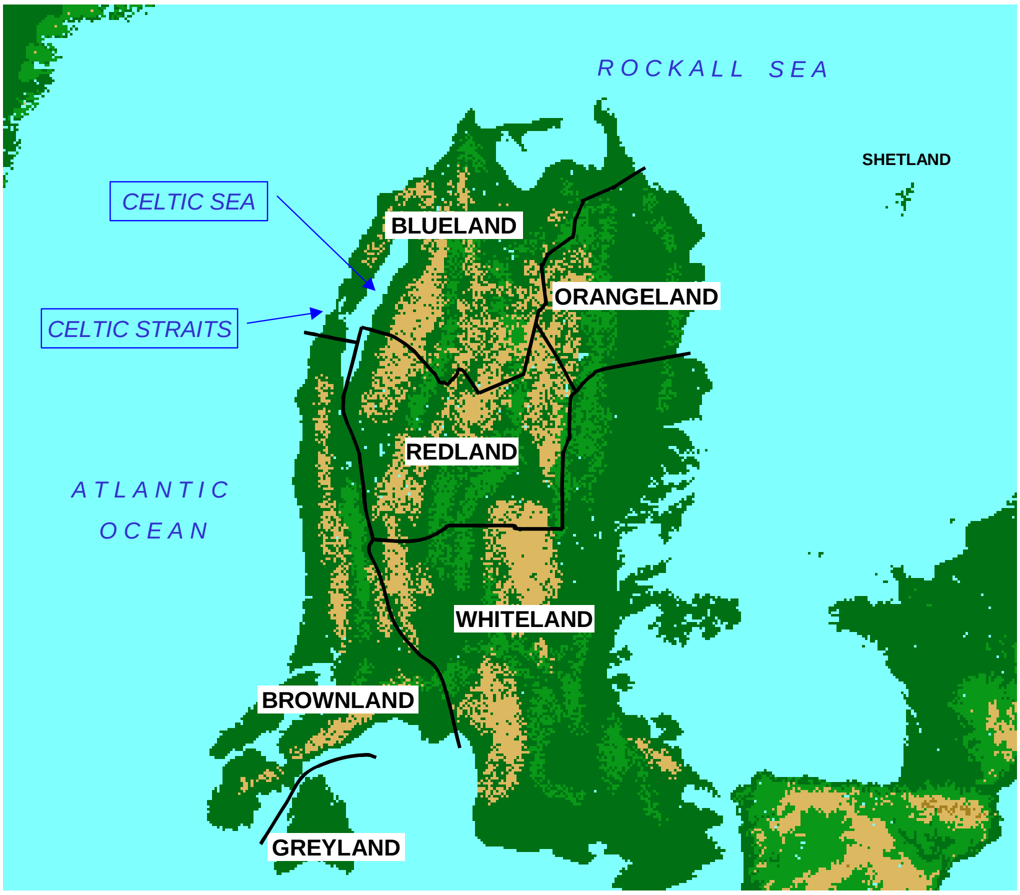


*Figure 1. The fictional continent of North Atlantis*

In 2005, the scenario was instantiated in a simulation environment based on the Scenario Toolkit and Generation Environment (STAGE). Algorithms for object assessment and situation assessment have been exercised in this simulation environment. This report is specifically concerned with how the algorithms for situation assessment will be evaluated in the simulation environment. This report contributes directly to fulfilment of a TTCP milestone.

A **complete** formal theory could in principle be developed for the **ground truth** of the scenario. This represents an omniscient view of the scenario in the simulation environment. It is complete in the sense that the truth or falsity of any atomic proposition[1] that is part of the description of the situation in the scenario can be inferred from the formal theory. Thus the formal theory would in principle allow the inference of a full set of atomic propositions to describe the situation in the scenario from a ground truth perspective. Examples of atomic propositions could be:

- Target 6 is a tuna fishing vessel.
- John Brown is an associate of Simon Black.

[1] An atomic proposition is a proposition that uses a single predicate. According to Reference 9, a predicate is a functor (function object) that gives a Boolean 'yes/no' answer to a question about an object.

- Orangeland adjoins Redland.

where the formal language has 'tuna fishing vessel', 'associate of' and 'adjoins' as predicates. In practice a complete representative subset of interest from the formal theory can be used to provide a symbolic expression of the ground truth. This becomes the standard to test the situation assessments against.

In the simulation environment, a given algorithm for machine-based situation assessment will also produce a set of atomic propositions to describe the situation in the scenario. At this stage, to simplify the process, it will be assumed that the information that the situation assessment algorithm receives from the object assessment is perfectly accurate and fully comprehensive information. Even so, the Situation Assessment (SA) algorithm may be imperfect, and so the SA propositions may diverge from the set of Ground Truth (GT) propositions. To assess the performance of the SA algorithms, there needs to be a method for comparing the set of SA propositions with the set of GT propositions. One aim of this report is to describe such a methodology that could be employed when the simulation experiments are performed.

# 3. Metrics

There are a few studies in the literature where the ground truth is compared with the output from an algorithm performing data fusion, in order to evaluate the effectiveness of the algorithm. Reference 10 focuses on Level 1 data fusion (object assessment) and Level 2 data fusion (situation assessment) in a traditional military scenario. Reference 10 presents a set of metrics for comparison of the algorithm output with the ground truth. Of great interest to the authors of this report is reference 11 where the focus is on Level 3 data fusion (threat assessment) in the context of a terrorist threat against national interests.

Reference 11 describes the employment of three metrics:

1. Recall
2. Precision
3. F-value

These can be defined with reference to the Venn diagram in Figure 2. The left-hand oval refers to the set of GT propositions (*GT*), and the right-hand oval refers to the set of SA propositions (*SA*). If the SA algorithm operates perfectly, then the two ovals will overlap perfectly. However, in general the intersection set will be a strict subset of the GT set and the SA set. Also, in general there will be propositions in the GT that aren't in the SA (*GT – SA*), and propositions in the SA that aren't in the GT (*SA – GT*).

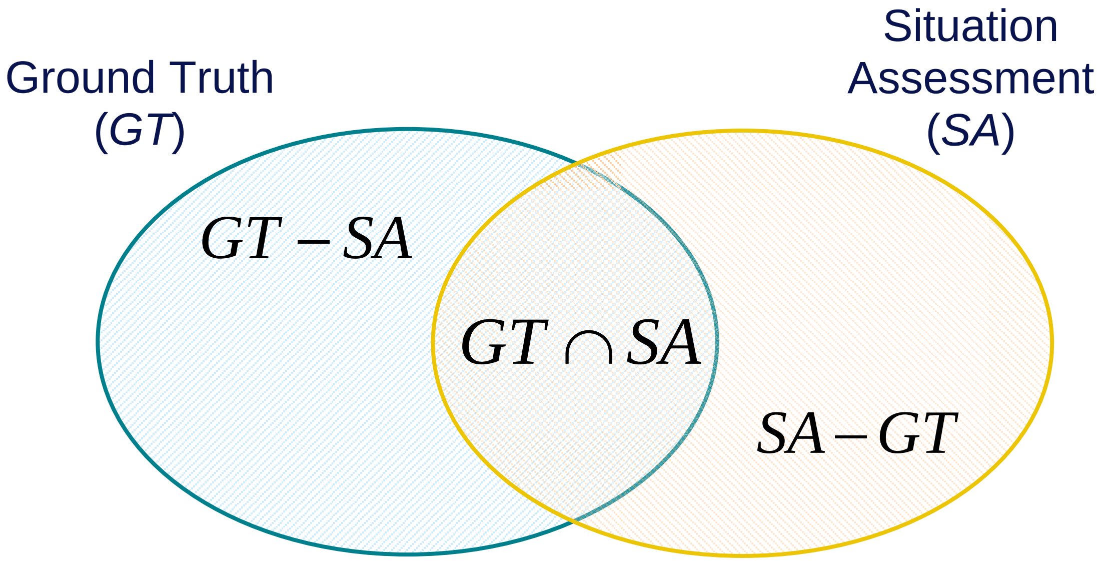


*Figure 2. Venn diagram of the comparison of the ground truth with the situation assessment*

The Recall is defined: $\text{Recall}\,(R) = \dfrac{|GT \cap SA|}{|GT|}$ (1)

The Precision is defined: $\text{Precision}\,(P) = \dfrac{|GT \cap SA|}{|SA|}$ (2)

Where $|X|$ indicates the cardinality of the set $X$.

Note that if the sets GT and SA are disjoint, then this indicates complete failure of the SA algorithm, and both the Recall and Precision are zero. On the other hand, if the SA algorithm operates perfectly, then the Recall and Precision both equal one.

It is often convenient to combine the Recall and Precision into a single metric. This is achieved using the F-value that is the geometric mean of the Recall and Precision:

$$\text{F - value} = \frac{2RP}{R+P} \qquad (3)$$

As with the Recall and Precision, the F-value is between zero and one, with a value of one indicating perfect performance of the SA algorithm.

Another similarity metric described in the literature is the Tversky similarity: [12]

$$\text{Tversky Similarity} = \frac{|GT \cap SA|}{|GT \cap SA| + \alpha|GT - SA| + \beta|SA - GT|} \tag{4}$$

Where $\alpha$ and $\beta$ are between zero and one, and are chosen to bias in favour of either $|GT - SA|$ or $|SA - GT|$.

From inspection of equations 1 to 4, it is apparent that:

1. Recall corresponds to the Tversky Similarity with $\alpha = 1$ and $\beta = 0$.
2. Precision corresponds to the Tversky Similarity with $\alpha = 0$ and $\beta = 1$.
3. F-value corresponds to the Tversky Similarity with $\alpha = 0.5$ and $\beta = 0.5$.

Thus the Recall / Precision / F-value are closely related to the Tversky Similarity.

Another possible global measure of the effectiveness of the SA algorithm can be derived from the probability density function over the possible worlds described by reference 6 (discussed in Section 1). The ground truth could be matched across the possible worlds in the situation assessment. The possible world with the closest match could be noted, along with its associated probability $p_1$. This could be compared with the probability of the most likely possible world $p_2 \geq p_1$. Such a comparison could yield a global measure of effectiveness.

# 4. Ontology Matching

For the simulation experiments to be run under the auspices of TTCP, the ground truth and situation assessment will be defined within the same ontological framework. However, the more general case is where the ground truth and situation assessment are defined within different ontological frameworks. To gain a better understanding of a how the comparison of ground truth and situation assessment might be performed in this more general case, the literature on ontology matching was consulted. The specific context was to imagine the ground truth as Ontology A being matched to the situation assessment as a different Ontology B. Definition of the matching function would facilitate a detailed comparison of the atomic propositions from the ground truth and situation assessment.

Broadly speaking, there are three types of ontology matching:

1. Terminological matching.
2. Learning-based matching.
3. Structural-based matching.

With terminological matching, the aim is to determine the quality-of-match between Concept A from Ontology A and Concept B from Ontology B. Concepts with a high quality-of-match are candidates for matching. One approach is to look at the **similarity of the strings** that comprise the names of the concepts using measures such as n-grams [13, 14], string edit distance [14], and longest common sub-sequence [15]. For example, if Concept A is *Defence* and Concept B is *Defense*, the string similarity is very high. Another approach is to examine the **linguistic affinity** between concepts [16, 14, 17]. For example, if Concept A is *drink* and Concept B is *beverage*, they have a high linguistic affinity because they are synonyms.

With learning-based matching [18, 19, 14, 15, 20, 13], often classification algorithms are employed such as the naïve Bayes classification algorithm, and complete-link hierarchical clustering. Concepts that belong to the same class or cluster are candidates for matching. For example, one approach is to obtain a corpus of documents relevant to Ontology A and another corpus of documents relevant to Ontology B. A document classifier is trained using the concepts in Ontology A and documents relevant to Ontology A, and is then exercised on the documents relevant to Ontology B. Similarly, a second document classifier is trained using the concepts in Ontology B and documents relevant to Ontology B, and is then exercised on the documents relevant to Ontology A. The net result is a measure of how much the documents associated with Concept A and Concept B tend to overlap, and the magnitude of this measure indicates whether the two concepts are candidates for matching [14]. Various other metrics have been defined in the literature to measure the similarity between concepts, such as the Jaccard Co-efficient [19], and the Cosine Measure [20].

With structural-based matching, the neighbourhoods of Concept A and Concept B are examined to determine how similar they are. Where they are quite similar, Concept A and Concept B are candidates for matching. One approach is to measure the **contextual affinity** between Concept A and Concept B [16]. It is determined whether the **attributes** of Concepts A and B are well-matched in terms of (a) linguistic affinity (discussed above), (b) compatibility of data types, and (c) whether the attributes are mandatory or optional.[2] It is also determined whether the relations in the neighbourhoods of Concepts A and B are well-matched in terms of (a) linguistic affinity, and (b) being of the same type, e.g. same-as, is-a, part-of.

An alternative approach to performing structural-based matching is **relaxation labelling** [19]. First a different type of matcher performs coarse matching, and then relaxation labelling is used to refine this over a series of iterations. It relies on measuring how well features in the neighbourhood of Concept A match with features in the neighbourhood of Concept B.

# 5. Exploring Employment of the Metrics

## 5.1 Simple Scenario

To explore the employment of the metrics discussed in Section 3, a simple scenario was developed, and coded using the logic programming language Prolog. This included a set of input propositions, along with rules to infer conclusions. Below is the full set of output propositions for the **ground truth**, including the conclusions derived from the rules, along with some brief explanation. First the activity of the vessel named the "Ironhorse" and its

[2] As an example to help explain the terminology, two concepts might be **student** and **person**. An attribute of **student** and **person** might be *age*. A relation involving **student** and **person** might be **student <u>is-a</u> person**, where **<u>is-a</u>** is a binary relation.

captain Samuel White is described. Even though Samuel White is associated with the Brownland military, the GT inference doesn't conclude that the Ironhorse is a threat.

- commercial_vessel(ironhorse) – The Ironhorse is a commercial vessel.
- captain_of_vessel(samuel_white,ironhorse) – Samuel White is the captain of the Ironhorse.
- recent_communication(samuel_white,daniel_smith) – Samuel White has recently been communicating with Daniel Smith.
- secret_service_of(brownland,daniel_smith) – Daniel Smith is a member of the Secret Service of Brownland.
- associated_with_military_of(brownland,samuel_white) – Thus Samuel White is associated with the military of Brownland.
- associated_with_military_of(brownland,ironhorse) – Thus the Ironhorse is associated with the military of Brownland.
- in_maritime_region_of(blueland,ironhorse) – The Ironhorse is in the territorial waters of Blueland.
- last_port_of_call(nectarville,ironhorse) – The last port-of-call of the Ironhorse was Nectarville.
- port(nectarville,brownland) – Nectarville is a port in Brownland.
- previous_country_visited(brownland,ironhorse) – Thus the previous country visited by the Ironhorse was Brownland.
- diplomatic_climate(blueland,brownland,normal) – The diplomatic climate between Blueland and Brownland is normal.

Second the activity of the vessel named the "Masked Avenger" and its captain Andrew Brown is described. Andrew Brown is associated with the Orangeland military, and in this case the inference does conclude that the Masked Avenger is a threat.

- commercial_vessel(masked_avenger) – The "Masked Avenger" is a commercial vessel.
- captain_of_vessel(andrew_brown,masked_avenger) – Andrew Brown is the captain of the Masked Avenger.
- recent_communication(andrew_brown,thomas_jones) – Andrew Brown has recently been communicating with Thomas Jones.
- secret_service_of(orangeland,thomas_jones) – Thomas Jones is a member of the Secret Service of Orangeland.
- associated_with_military_of(orangeland,andrew_brown) – Thus Andrew Brown is associated with the military of Orangeland.
- associated_with_military_of(orangeland,masked_avenger) – Thus the Masked Avenger is associated with the military of Orangeland.
- in_maritime_region_of(blueland,masked_avenger) – The Masked Avenger is in the territorial waters of Blueland.
- last_port_of_call(jorvik,masked_avenger) – The last port-of-call of the Masked Avenger was Jorvik.
- port(jorvik,orangeland) – Jorvik is a port in Orangeland.
- previous_country_visited(orangeland,masked_avenger) – Thus the previous country visited by the Masked Avenger was Orangeland.

- diplomatic_climate(blueland,orangeland,very_tense) – The diplomatic climate between Blueland and Orangeland is very tense.
- hostile_intent_towards(blueland,masked_avenger,orangeland) – Thus the Masked Avenger has hostile intent towards Blueland.
- risk_of_dangerous_cargo(masked_avenger,blueland,high) – Thus there is a high risk that the Masked Avenger is carrying cargo that is dangerous from a Blueland perspective.
- potential_threat(masked_avenger,blueland,high) – Thus a high level of threat is posed by the Masked Avenger to Blueland interests.

Figure 3 shows an example of one of the GT rules in Prolog format. The output predicate is "hostile_intent_towards", and the input predicates are on the right-hand-side. When X is "blueland", Y is "masked_avenger", and Z is "orangeland", all the input propositions are TRUE, and consequently the output proposition is also TRUE, because each comma between two propositions in the rule indicates logical AND.[3] All these propositions appear in the list of propositions above, except the "ok" predicate that is discussed below.

```
hostile_intent_towards(X, Y, Z) :- associated_with_military_of(Z, Y),
                                   diplomatic_climate(X, Z, very_tense),
                                   in_maritime_region_of(X, Y),
                                   ok(hostile_intent_towards(X, Y, Z)).
```

*Figure 3. An example of one of the GT rules in Prolog format*

In forming the **situation assessment**, the same set of input propositions and rules were employed, but a few errors were deliberately introduced into the rules to model real-life deficiencies in SA algorithms. For example, Figure 4 shows the SA rule corresponding to the GT rule in Figure 3. Inspection of both figures shows that the "diplomatic_climate" predicate has been deleted from the SA rule.

```
hostile_intent_towards(X, Y, Z) :- associated_with_military_of(Z, Y),
                                   in_maritime_region_of(X, Y),
                                   ok(hostile_intent_towards(X, Y, Z)).
```

*Figure 4. The SA rule corresponding to the GT rule in Figure 3*

## 5.2 Calculation of the Metrics

Prolog was used to obtain a complete listing of: (a) the output propositions for the ground truth (given in Section 5.1), and (b) the output propositions for the situation assessment. Within Prolog, the metrics described in Section 3 were calculated; refer to Table 1 for the results. The overall comparison between the ground truth and situation assessment yielded an F-value of 0.91.

[3] hostile_intent_towards is the predicate, and hostile_intent_towards(X,Y,Z) and hostile_intent_towards(blueland, masked_avenger, orangeland) are propositions.

*Table 1. The metrics calculated in Prolog for the simple scenario*

| **Metric / Parameter** | **Value** |
|---|---|
| $\|GT\|$ | 25 |
| $\|SA\|$ | 28 |
| $\|GT \cap SA\|$ | 24 |
| Recall | 0.96 |
| Precision | 0.86 |
| F-value | 0.91 |

The scoring method described above effectively weights all of the output propositions equally when the F-value is calculated. If the output propositions were somehow ranked in order of perceived importance, then some type of weighted sum could be devised to calculate quantities mirroring $|GT|$, $|SA|$ and $|GT \cap SA|$ where more important propositions would contribute more compared to propositions with lesser importance. For example, taking the simple scenario described in Section 5.1, potential_threat(masked_avenger,blueland, high) may be perceived as more important than commercial_vessel(masked_avenger). In fact, reference 11 (discussed in Section 3) employs weighted sums to calculate quantities mirroring $|GT|$, $|SA|$ and $|GT \cap SA|$. The employment of weightings could be a future extension to the methodology described in this report.

## 5.3 False Positives and Negatives

**False positives** are the propositions that are in the situation assessment, but not in the ground truth (i.e. in *SA – GT* with reference to Figure 2). They represent spurious information inserted by the situation assessment. For the simple scenario described above, there were five false positives:

1. **hostile_intent_towards(blueland,ironhorse,brownland)**
2. risk_of_dangerous_cargo(ironhorse,blueland,medium)
3. potential_threat(ironhorse,blueland,medium)
4. **risk_of_dangerous_cargo(masked_avenger,blueland,medium)**
5. potential_threat(masked_avenger,blueland,medium)

The false positives in boldface are special cases where the input propositions to the rule were TRUE in the situation assessment and ground truth, but the output proposition was TRUE in the situation assessment and FALSE in the ground truth. This indicates some type of corruption in the rule in the situation assessment. For example, the first false positive (hostile_intent_towards) can be seen to be a direct consequence of the error introduced in Figure 4, that is, deleting the predicate diplomatic_climate. On the other hand, the false positives not in boldface have at least one input proposition to the rule that is TRUE in the situation assessment, but FALSE in the ground truth. Thus it isn't surprising that the output proposition is TRUE in the situation assessment but FALSE in the ground truth. The false positives in boldface can be viewed as pointers to where the situation assessment's underlying theory begins to diverge from the ground truth. For convenience they are termed **root false positives**.

**False negatives** are the propositions that are in the ground truth, but not in the situation assessment (i.e. in *GT – SA* with reference to Figure 2). They represent information not captured by the situation assessment. For the simple scenario described above, there were two false negatives:

- **risk_of_dangerous_cargo(masked_avenger,blueland,high)**
- potential_threat(masked_avenger,blueland,high)

Using a similar argument to above, the false negative in boldface is a **root false negative**, and indicates where the situation assessment's underlying theory starts to diverge from the ground truth.

## 5.4 Partial Matching

One approach to matching propositions between the ground truth and situation assessment is to maintain that if they are identical there is a match, and if they are not identical there is no match. A less "black-and-white" approach is to allow partial matching – refer to the two examples in Table 2 from the simple scenario described above.

*Table 2. Examples of partial matching from the simple scenario*

| | **Ground Truth Proposition** | **Situation Assessment Proposition** | **Quality-of-Match** |
|---|---|---|---|
| 1 | risk_of_dangerous_cargo(masked_avenger, blueland, high) | risk_of_dangerous_cargo(masked_avenger, blueland, medium) | 0.5 |
| 2 | potential_threat(masked_avenger, blueland, high) | potential_threat(masked_avenger, blueland, medium) | 0.5 |

The quality-of-match value must be between zero (no match) and one (perfect match). In the case of the partial matches in Table 2, the only difference is that the third argument is "high" in the ground truth, but the situation assessment assigns "medium" to the third argument. The Prolog code searches for cases where the only difference between the propositions is in a single argument to detect a partial match. For example, where that difference is high versus medium as in Table 2, the code automatically assigns a quality of match of 0.5

Another example, this time hypothetical, might be the proposition commercial_vessel(target$_6$) in the ground truth versus fishing_vessel(target$_6$) in the situation assessment. The Prolog code would detect that both propositions refer to the same target, and may assign a partial match with a quality-of-match of 0.3 based on the perceived difference between "commercial vessel" and "fishing vessel". In general, the quality-of-match values are based on the professional judgements of the personnel who configure the system for comparing the SA with the GT.

The total of the quality-of-match scores from the partial matches contribute to the intersection between the ground truth and situation assessment propositions, that is $|GT \cap SA|$ in Figure 2. For example, perfect matches contribute 23 points to the score for $|GT \cap SA|$ in Table 1. The partial matches in Table 2 also contribute one point (2 x 0.5) to give the final score of 24.

In Table 2, partial matches are assigned when only one difference between the GT and SA propositions is detected. Although it hasn't been implemented, a more complex scheme could be employed that caters for partial matches with more than one difference.

The examples of partial matching discussed thus far tend to be more of a discrete nature, for example the value-of-interest being either "low", "medium" or "high". There will also need to be matching involving continuous variables such as distance and time. Reference 11 provides one means of achieving this by employing the sigmoid function. A hypothetical example is the GT proposition referring to a range range(51000,1200) indicating that the range is 51000m. The partially matching SA proposition might be range(51400,1200) indicating a range of 51400m. Thus the situation assessment is in error by 400m. The second argument in the "range" predicate indicates the range error (in this example 1200m) that corresponds to a quality-of-match of 0.5 – refer to this range error as the "nominal error" for convenience. For an arbitrary range error, the ratio of the nominal error to the actual error is calculated (1200 / 400 = 3 in our example), and then the sigmoid function in Figure 5 is consulted. Reading from the graph gives a quality-of-match of 0.75 for our example.

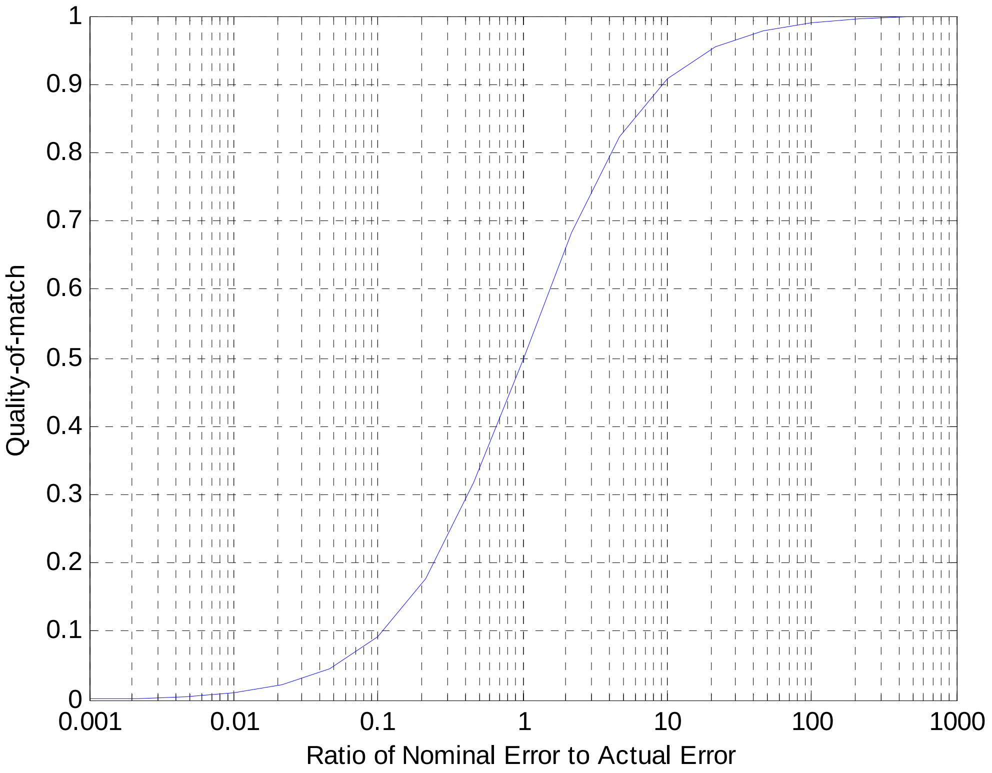


*Figure 5. Sigmoid function used to calculate the quality-of-match with continuous variables*

Inspection of the sigmoid function in Figure 5 confirms that very large errors will yield a quality-of-match approaching 0, and very small errors will yield a quality-of-match approaching 1. This general approach only requires one parameter (nominal error) to specify the quality-of-match for arbitrary errors, and having the nominal error as an argument in the predicate means that it can be varied as required on a case-by-case basis.

The approach described above employs a fixed sigmoid function in Figure 5. A more complex approach would be to include a third argument in the "range" predicate that is a scaling parameter that either compresses or stretches the sigmoid curve along the axis entitled "Ratio of Nominal Error to Actual Error". This would provide additional flexibility in specifying the quality-of-match involving continuous variables, however there would need to be some principled means of judging what value to give to this scaling parameter in each specific context.

For more complex scenarios there may be ambiguity about which partial matches to form. For example, a given proposition from the ground truth may potentially be matched with ten different propositions from the situation assessment. In general a linear assignment algorithm will be required to decide which partial matches to form. The optimisation goal might be to maximise the total of all the quality-of-match values from the full set of partial matches. Candidate assignment algorithms are:

1. The auction algorithm [21, 24]. The authors of this DSTO Technical Note have access to C code for this algorithm.[4]
2. The Jonker-Volgenant-Castañon (JVC) algorithm [22, 23, 24].

## 5.5 Iterative Correction Process

As discussed in Section 5.3, the root false positives and root false negatives can be viewed as pointers to where the situation assessment starts to diverge from the ground truth. Thinking in terms of networks of propositions, one may view the root false positives and negatives as the start of branches of propositions in the situation assessment and ground truth where they diverge away from each other. This study seeks to test the thesis that, once identified, the root false positives and negatives can be corrected, thus pruning the branches of divergence.

For example, from Section 5.3, one of the root false positives was:

**1. hostile_intent_towards(blueland,ironhorse,brownland)**

In Prolog, the approach taken is to introduce the "ok" predicate shown in Figure 4. For the specific root false positive listed above, the ok predicate takes the form shown in Figure 6.

[4] Supplied by Dr Jason Williams (DSTO).

```
ok(hostile_intent_towards(X, Y, Z)) :-
                    (X, Y, Z) \== (blueland, ironhorse, brownland).
```

*Figure 6. The "ok" predicate employed in the situation assessment to correct root false positives*

The role of the ok predicate is to allow only those combinations of X, Y and Z that are "OKAY", that is for the specific case of hostile_intent_towards, the combination X = blueland, Y = ironhorse, Z = brownland is disallowed in order to correct the root false positive.[5] The ok predicate in Figure 6 could be expanded with other combinations of X, Y and Z if there was a requirement to correct other root false positives related to hostile_intent_towards. The second root false positive listed in Section 5.3 [risk_of_dangerous_cargo(masked_avenger,blueland,medium)] would also require a similar employment of the ok predicate.

It is a simpler matter to correct the root false negatives in Prolog. For example, to correct the root false negative in Section 5.3, it is suffice to include the following proposition in the situation assessment, effectively inserting the missing information:

- risk_of_dangerous_cargo(masked_avenger,blueland,high)

Once the root false positives and negatives have been corrected in Prolog, the Recall, Precision and F-value can be calculated again in a second iteration. The new results are shown in Table 3. Note that in this case, the corrections have resulted in a significant improvement to the F-value, and the numbers of false positives and false negatives have been significantly reduced. In fact, for the second iteration there are no false positives, and only one false negative (that is also a root false negative):

- **potential_threat(masked_avenger,blueland,high)**

*Table 3. The metrics calculated in Prolog for the second iteration*

| **Metric / Parameter** | **Value** |
|---|---|
| $\|GT\|$ | 25 |
| $\|SA\|$ | 24 |
| $\|GT \cap SA\|$ | 24 |
| Recall | 0.96 |
| Precision | 1 |
| F-value | 0.98 |

Now this root false negative can be corrected, and then the Recall, Precision and F-value can be calculated in a third iteration. This time the ground truth and situation assessment match perfectly, and the Recall, Precision and F-value are all one. Figure 7 shows the increase in the F-value over the iterations of (re-)calculation and correction.

[5] In Prolog, T1 \== T2 is TRUE when the two terms T1 and T2 are not literally identical.

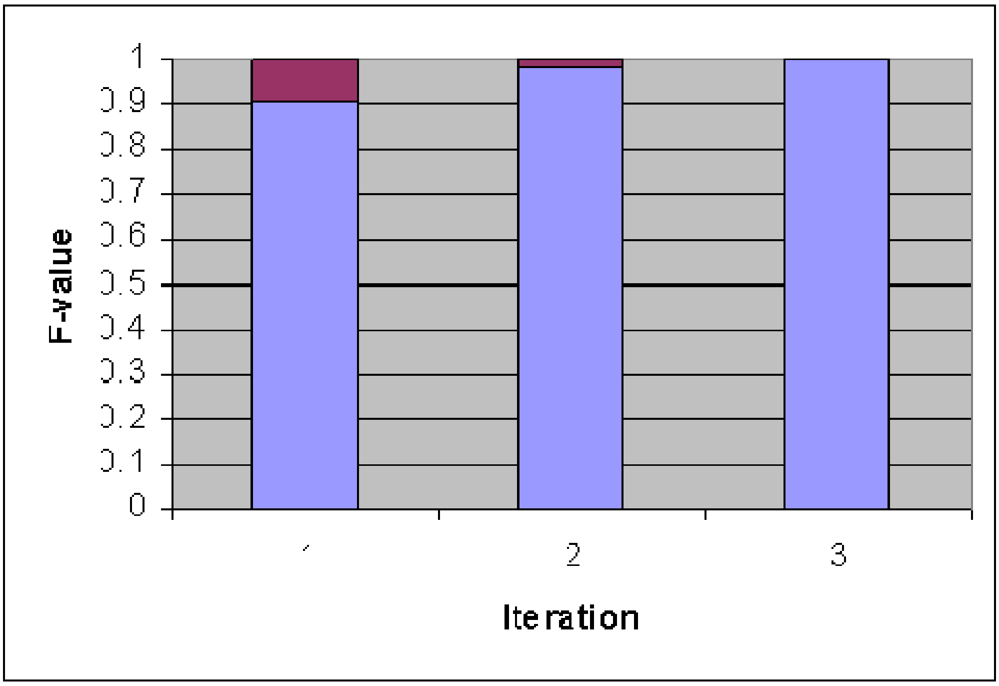


*Figure 7. The increase in the F-value over the iterations of (re-)calculation and correction*

It is the thesis of this report that the integrated area in maroon in Figure 7 is a measure of the effectiveness of the situation assessment algorithm because it provides a measure of how difficult it is to correct the situation assessment. This measure will be zero if the algorithm performs perfectly, and the situation assessment perfectly matches the ground truth in the first iteration. The measure will be quite large if the F-value is significantly less than one in the first iteration, and it only very slowly converges to a value of one. The value of the measure could be transformed using a non-linear function to yield a more intuitive result where one indicates perfect performance, and zero indicates complete failure.

For the simple scenario above, the F-value reached unity in three iterations. However, in the general case the authors are currently uncertain about the convergence properties of the F-value. Thinking in terms of networks of propositions, it is uncertain whether correction could result in the creation of new branches of divergence between the situation assessment and ground truth. Possibilities include (a) the F-value being oscillatory, and (b) the F-value not converging to unity.

One simple result is the condition under which the F-value is strictly increasing from one iteration to the next:

$$d|GT \cap SA| > \frac{F}{2} d|SA| \qquad (5)$$

in the limit of infinitesimal changes, where $|SA|$ and $|GT \cap SA|$ are defined in Section 3.

### 5.6 Further work

Further work is required to investigate the convergence properties of the F-value in Figure 7 in the general case. The question to be answered is: Under what conditions will the F-value converge to unity? Two avenues have been identified to pursue this question:

1. Examination of the theory of non-monotonic logics that are relevant to the iterative correction process described in Section 5.5.
2. Experimentation with the random inference networks discussed in Section 6, specifically, examining what type of convergence behaviour they exhibit.

A further extension to the methodology described above for comparing the situation assessment with the ground truth could be to weight the output propositions according to their importance when calculating the Recall, Precision and F-value. This was discussed in Section 5.2.

## 6. Random Inference Networks

### 6.1 Background

Reference 25 notes that **complex networks** are being studied across many fields of science. Complex networks can be modelled as structures consisting of nodes or vertices connected by links or edges. Examples include:

1. The Internet is a network of routers or domains.
2. The World Wide Web is a network of websites.
3. The electrical power grid can be described as a network.
4. The global economy is a network of national economies, that in turn are networks of markets, that in turn are networks of producers and consumers.
5. In nature, food webs can be described using networks.
6. An organisation is a network of people.
7. In the social domain there are friendship networks.
8. The human brain is a network of neurons.
9. In the human body metabolic pathways can be described using networks.

Reference 25 discusses some of the different types of networks that have been identified:

1. **Exponential networks** where the number of links per node is fairly uniform across the network, e.g. a roadmap showing highways in the U.S. where the nodes are cities.
2. **Scale-free networks** where a few nodes have a large number of connections, but most nodes have only a few connections, e.g. an airline routing map in the U.S. where the nodes are airports.
3. **Small-world networks** where there is local clustering, and the average path length between two randomly selected nodes is low (the so-called “small world” effect), e.g. friendship networks.

Scale-free networks such as the Internet have the property that they are robust against random failures of nodes or links. However, they are fragile against intentional attacks aimed at key nodes that have a large number of connections.

Of key interest to this report, damage can spread across networks, e.g. diseases across social networks, viruses across the Internet, and failures can cascade across the power grid. The networks of interest in this report are the inference networks used to form the situation assessment. Errors can occur in such an inference network, and this can then spread as damage across the inference network.

Random Boolean Networks (RBN), also known as NK networks, were pioneered as simplified models of gene regulation and control [26, 27]. In such a network there are N nodes that are connected using directed links to form various cyclic pathways. For each node there are exactly K incoming links that control the state of the node. An example is shown diagrammatically in Figure 8. The state of each node is either zero or one, and this is controlled by the K incoming links via a random Boolean function.

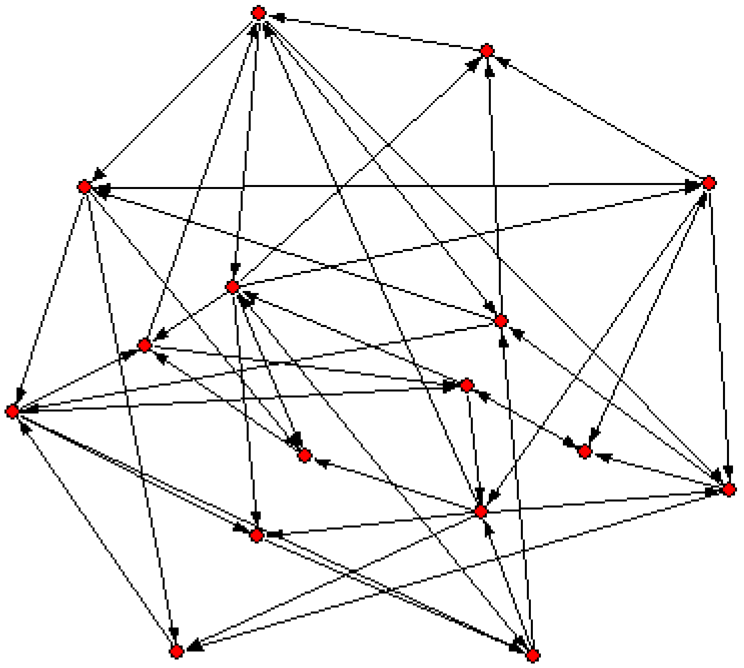


*Figure 8. Example of an NK network with N = 16 and K = 3, taken from reference 28.*

For example, consider hypothetically that K is two, and node A is controlled by node B and node C. Simulation can be used to study the dynamics of the NK networks. At a given step in the simulation, let the state of node A be zero, node B one, and node C zero. Let the random Boolean function be logical OR in this specific case. Then when the state of all the nodes is synchronously updated to proceed to the next step in the simulation, the state of node A will change to one since (node B) OR (node C) is one. Likewise the states of nodes B and C will be updated depending on their particular controlling inputs, and similarly for all the remaining nodes in the network. The simulation is initialised by randomly assigning each node a value of zero or one. Then the dynamics of the network is studied by studying the states of the nodes as the simulation progresses through its successive steps. Reference 29 has a Java applet that can be used to observe the dynamics of an NK network.

Damage spreading has been studied for NK networks by switching (perturbing) the initial state of a randomly chosen node, and determining how this impacts the dynamics of the network (e.g. reference 30). The nature of the damage spreading is dependent on K:

1. **Frozen regime**. For K = 1, the initial perturbation dies out quickly, after which both the original and perturbed network display the same pattern.
2. **Critical regime**. For K = 2, the effect of the perturbation tends to persist, but it only impacts part of the network.
3. **Chaotic regime**. For K = 3 or greater, the perturbation causes extensive changes to the network dynamics.

Researchers currently believe that gene regulation networks operate in or near the critical regime, because evolution requires that there must be sensitivity to perturbations and mutations, but not the very high sensitivity of the chaotic regime.

## 6.2 Creation of Networks

One aim of this report was to examine damage spreading in the inference networks used for situation assessment in a similar manner to how damage spreading has been studied in NK networks, as discussed in Section 6.1. The process employed is summarised in Figure 9. Random inference networks were created to study damage spreading, one for the ground truth, and another for the situation assessment. The only difference between the networks when they were initialised was a single perturbation applied to the SA network. The aim was to measure how this perturbation spread throughout the SA network by comparing the final states of the SA network and GT network.

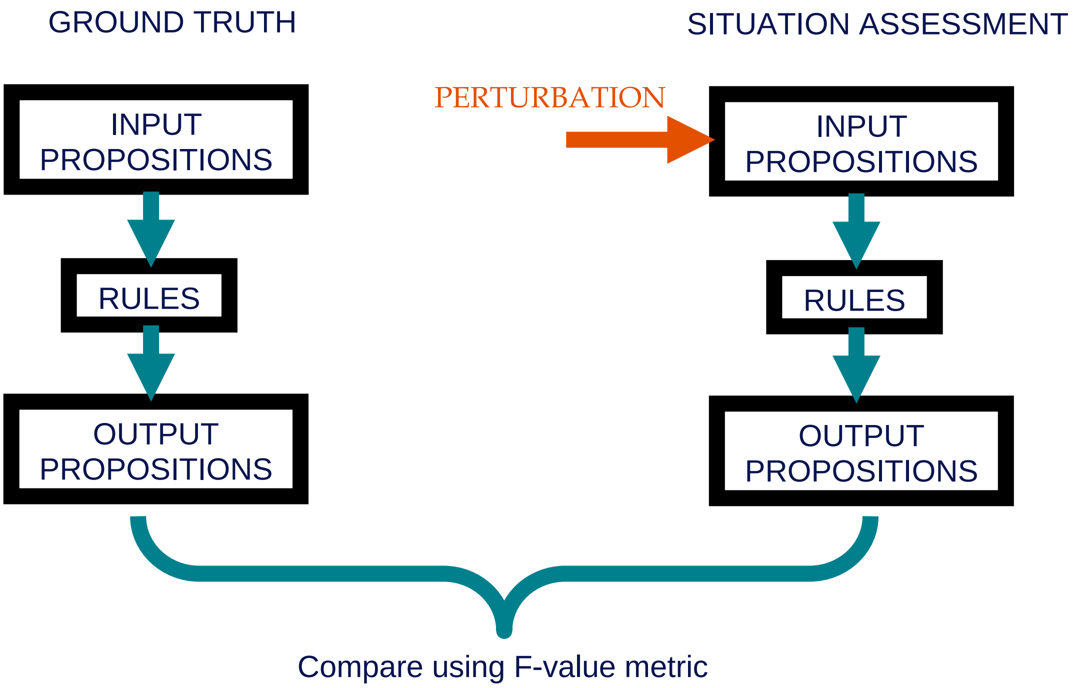
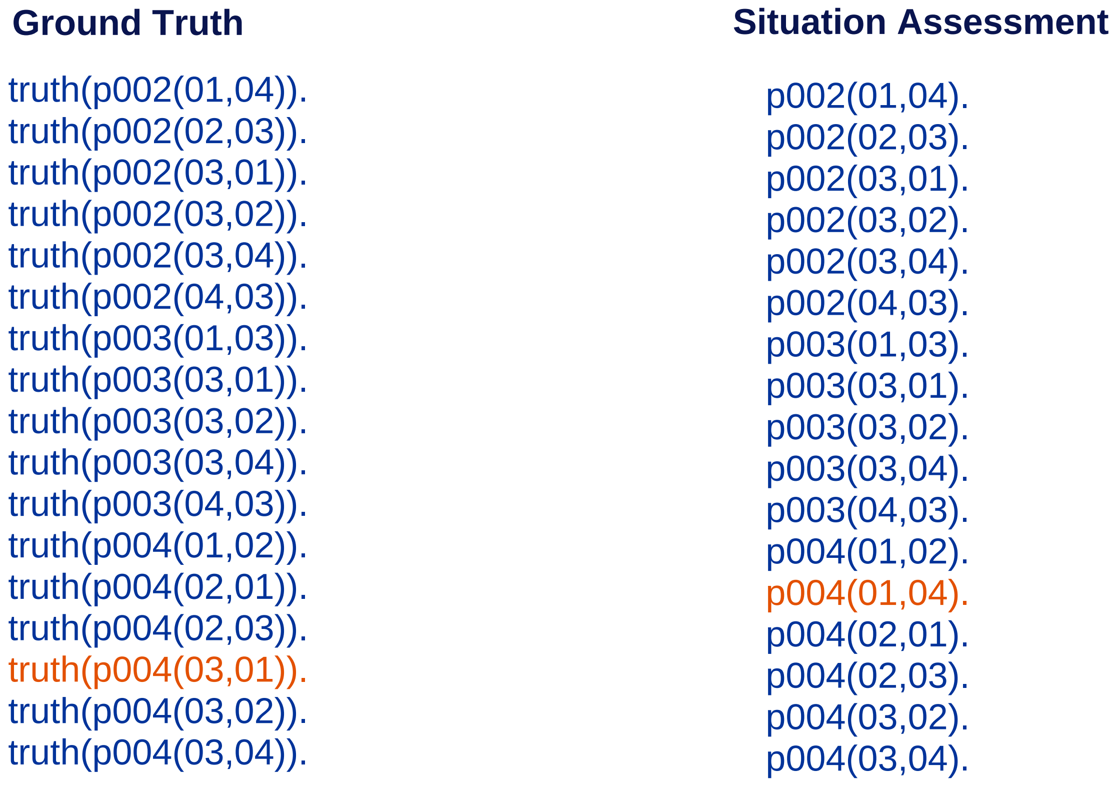


*Figure 9. The process employed in studying damage spreading in inference networks*

**Input propositions** were created that modelled the input that the situation assessment would receive from sources like the object assessment. Figure 10 shows some input propositions extracted from a GT random inference network, and the corresponding SA network (in Prolog format). Note that the two sets of input propositions match perfectly apart from the mismatching propositions in orange caused by the perturbation to the SA network that was applied randomly. This models deficiencies in the object assessment as input to the situation assessment algorithm. In this case four input predicates were employed (p001, p002, p003 and p004),[6] and four targets were employed (01, 02, 03 and 04) to form the input propositions. Each predicate formed a binary relation involving two targets as arguments. The two arguments of an input proposition were not allowed to be identical. The predicate "truth" differentiates the GT input propositions from the SA input propositions. The input propositions were created randomly with duplication of propositions disallowed.

[6] More concrete examples of predicates are given in Section 5.1.



Ground Truth

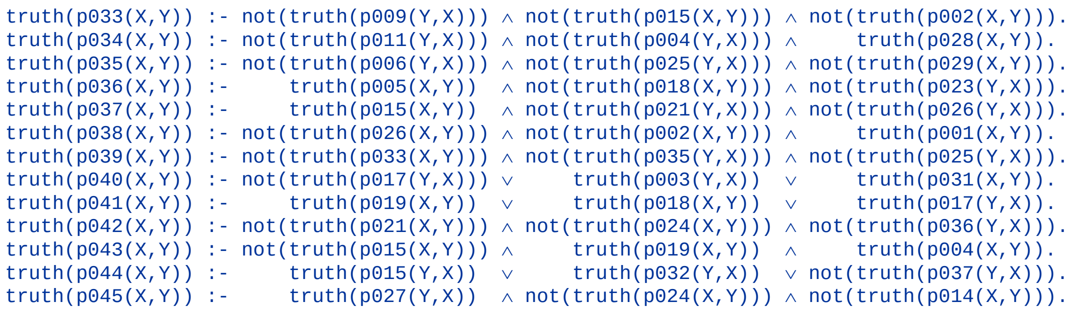
```
truth(p002(01,04)).
truth(p002(02,03)).
truth(p002(03,01)).
truth(p002(03,02)).
truth(p002(03,04)).
truth(p002(04,03)).
truth(p003(01,03)).
truth(p003(03,01)).
truth(p003(03,02)).
truth(p003(03,04)).
truth(p003(04,03)).
truth(p004(01,02)).
truth(p004(02,01)).
truth(p004(02,03)).
truth(p004(03,01)).
truth(p004(03,02)).
truth(p004(03,04)).
```

Situation Assessment

```
p002(01,04).
p002(02,03).
p002(03,01).
p002(03,02).
p002(03,04).
p002(04,03).
p003(01,03).
p003(03,01).
p003(03,02).
p003(03,04).
p003(04,03).
p004(01,02).
p004(01,04).
p004(02,01).
p004(02,03).
p004(03,02).
p004(03,04).
```

*Figure 10. Input propositions extracted from a GT inference network and corresponding SA network*

Referring to Figure 9, **inference rules** were also created randomly. The **same** rules were employed for both the GT network and SA network. Figure 11 shows an extract of rules from a GT random inference network (hence the "truth" predicate is used). The rules are shown in Prolog format with one minor difference: ∧ (instead of ,) is used as shorthand for logical AND, and ∨ (instead of ;) is used as shorthand for logical OR.



```
truth(p033(X,Y)) :- not(truth(p009(Y,X))) ∧ not(truth(p015(X,Y))) ∧ not(truth(p002(X,Y))).
truth(p034(X,Y)) :- not(truth(p011(Y,X))) ∧ not(truth(p004(Y,X))) ∧     truth(p028(X,Y)).
truth(p035(X,Y)) :- not(truth(p006(Y,X))) ∧ not(truth(p025(Y,X))) ∧ not(truth(p029(X,Y))).
truth(p036(X,Y)) :-     truth(p005(X,Y))  ∧ not(truth(p018(X,Y))) ∧ not(truth(p023(Y,X))).
truth(p037(X,Y)) :-     truth(p015(X,Y))  ∧ not(truth(p021(Y,X))) ∧ not(truth(p026(Y,X))).
truth(p038(X,Y)) :- not(truth(p026(X,Y))) ∧ not(truth(p002(X,Y))) ∧     truth(p001(X,Y)).
truth(p039(X,Y)) :- not(truth(p033(X,Y))) ∧ not(truth(p035(Y,X))) ∧ not(truth(p025(Y,X))).
truth(p040(X,Y)) :- not(truth(p017(Y,X))) ∨     truth(p003(Y,X))  ∨     truth(p031(X,Y)).
truth(p041(X,Y)) :-     truth(p019(X,Y))  ∨     truth(p018(X,Y))  ∨     truth(p017(Y,X)).
truth(p042(X,Y)) :- not(truth(p021(X,Y))) ∧ not(truth(p024(X,Y))) ∧ not(truth(p036(Y,X))).
truth(p043(X,Y)) :- not(truth(p015(X,Y))) ∧     truth(p019(X,Y))  ∧     truth(p004(X,Y)).
truth(p044(X,Y)) :-     truth(p015(Y,X))  ∨     truth(p032(Y,X))  ∨ not(truth(p037(Y,X))).
truth(p045(X,Y)) :-     truth(p027(Y,X))  ∧ not(truth(p024(X,Y))) ∧ not(truth(p014(X,Y))).
```

*Figure 11. An extract of rules from a GT random inference network*

In Figure 11, the extracted rules are numbered sequentially from 33 to 45. As each rule was created, it was randomly linked back to three predicates already created: either rules or else predicates used to create the input propositions. For example, when rule 35 was created, there were already in place predicates numbered from 1 to 4 corresponding to the input propositions, and rules numbered from 5 to 34. Three different predicates between 1 and 34 were chosen at random; in this case they were p006, p025 and p029. (As an aside, p006, p025 and p025 would have been disallowed since the predicates had to be different.) All combinations of three different predicates between p001 and p034 were equally likely to be chosen, i.e. there wasn't any preferential selection. A similar approach to growing directed networks is described in reference 31.

In Figure 11, all the rules and input predicates to the rules have an arity of two, i.e. each has two arguments. With the input predicates to the rules, the order of the arguments was swapped according to the toss of an unbiased coin. For example, with p035 the first two input predicates had their arguments swapped (Y,X), whereas the third input predicate was not swapped (X,Y). Negation was applied to each input predicate according to the toss of an unbiased coin. For example, with p035 negation ("not") was applied to each input predicate (this occurrence has a probability of 1/8). Whether a rule employed logical AND (∧) or logical OR (∨) was governed by the toss of a coin that could be **biased** either in favour of logical AND or logical OR. For the rules in Figure 11 the coin was biased in favour of logical AND, and, as a specific example, p035 employed logical AND.

There are 256 possible different Boolean functions that can relate a rule to three input predicates. As a specific example, the Boolean function for p035 is shown in Table 4. In this case, all the input predicates must evaluate to FALSE in order for the rule to evaluate to TRUE. With reference to the final column in Table 4, the 256 possible Boolean functions correspond to the possible permutations of the eight data rows, where each row can have the value TRUE or FALSE.

*Table 4. The Boolean function for rule p035 in Figure 11*

| Input Predicate 1 | Input Predicate 2 | Input Predicate 3 | Rule |
|---|---|---|---|
| TRUE | TRUE | TRUE | FALSE |
| TRUE | TRUE | FALSE | FALSE |
| TRUE | FALSE | TRUE | FALSE |
| TRUE | FALSE | FALSE | FALSE |
| FALSE | TRUE | TRUE | FALSE |
| FALSE | TRUE | FALSE | FALSE |
| FALSE | FALSE | TRUE | FALSE |
| FALSE | FALSE | FALSE | TRUE |

The simple scheme for rule creation employed in this study can only create a subset of the 256 possible Boolean functions; in fact, only 16 different functions can be created. The 16 functions have a similar form to the final column in Table 4:

1. If logical AND is chosen for the rule then all rows show FALSE except one row that shows TRUE; or
2. If logical OR is chosen for the rule then all rows show TRUE except one row that shows FALSE.

The specific row that shows the exception depends upon how negation ("not") is applied to the three input predicates in the rule; note that there are eight possible options for this corresponding to the eight data rows in Table 4. For example, since negation is applied to each input predicate of rule p035 in Figure 11, each input predicate must evaluate to FALSE in order for the rule to evaluate to TRUE, and hence the final row in Table 4 shows the "exception" (TRUE). Table 5 shows the 16 Boolean functions employed in this study. The eighth function was employed for p035 – refer to Table 4. All 16 functions were equally likely to be chosen for a given rule if the AND-OR bias was set to 0.5. The first eight functions were more prevalent compared with functions 9 to 16 if logical AND was favoured, and conversely functions 9 to 16 were more prevalent if logical OR was favoured.

*Table 5. The 16 Boolean functions employed in this study*

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T | F | F | F | F | F | F | F | F | T | T | T | T | T | T | T |
| F | T | F | F | F | F | F | F | T | F | T | T | T | T | T | T |
| F | F | T | F | F | F | F | F | T | T | F | T | T | T | T | T |
| F | F | F | T | F | F | F | F | T | T | T | F | T | T | T | T |
| F | F | F | F | T | F | F | F | T | T | T | T | F | T | T | T |
| F | F | F | F | F | T | F | F | T | T | T | T | T | F | T | T |
| F | F | F | F | F | F | T | F | T | T | T | T | T | T | F | T |
| F | F | F | F | F | F | F | T | T | T | T | T | T | T | T | F |

Reference 32 refers to the functions shown in Table 5 as canalysing Boolean functions. This means that by holding one of the inputs in a certain state, the output is fixed. For example, with reference to Table 4, when any of the inputs is held at TRUE, then the output must be FALSE. Only a subset of the 256 possible Boolean functions are canalysing, and the 16 Boolean functions used in this study are a subset of the full set of canalysing Boolean functions. The question then arises as to whether employing this specific subset of canalysing Boolean functions (shown in Table 5) has biased the analysis in any way. This question will be addressed further in Section 6.3.

Returning to Figure 9, the input propositions and inference rules were processed in Prolog to produce two sets of **output propositions**, one for the ground truth and one for the situation assessment. Possible examples of output propositions for the situation assessment are:

- p003(03,04)
- p039(02,01)
- p060(04,02)

As an example of applying a rule (referring to Figure 11), p035(01,03) would evaluate to TRUE and be an output proposition for the ground truth if p006(03,01), p025(03,01) and p029(01,03) all evaluated to FALSE in the ground truth.

There was a need to determine the level of damage spreading caused by the single perturbation to the situation assessment described above. This was achieved by comparison of the two sets of output propositions through calculation of the F-value metric described in Section 3. This resulted in a normalised measure of the damage spreading, with a value closer to unity indicating lesser damage, and a value closer to zero indicating greater damage. The authors note that some other studies employ absolute measures of damage spreading, e.g. reference 30, however the authors believe that a normalised measure is more appropriate when comparing damage spreading across random inference networks of different size.

There are various parameters that describe the structure of the random inference networks. The baseline set of parameters employed in this study are listed in Table 6. For certain parameters, the dependence of the F-value was tested by varying the parameter value from the baseline and recalculating the F-value. The preliminary results of this experimentation are described in Section 6.3.

*Table 6. Baseline set of parameters used to create the random inference networks*

| Parameter | Value | Comments |
|---|---|---|
| Arity of the predicates (number of arguments) | 2 | All predicates had the same arity. |
| The number of predicates used to create the input propositions | 4 | p001, p002, p003, p004 |
| The number of targets used to create the input propositions | 4 | 01, 02, 03, 04 |
| Number of input propositions created | 24 | |
| The number of input predicates for each rule | 3 | |
| Bias in favour of using logical AND during rule creation | 0.5 | Bias must be between 0 and 1 |
| Number of rules created | 64 | |
| Number of independent networks created | 100 | |

Referring to Table 6:

- The number of input propositions randomly created (24) corresponded to half the total number of possible input propositions (48). (Prolog assumed that the 24 "missing" input propositions were FALSE.)
- Regards the bias:
    - A value of 0 corresponded to exclusive use of logical OR.
    - A value of 0.5 corresponded to using an unbiased coin to choose between logical AND and logical OR when creating a rule.
    - A value of 1 corresponded to exclusive use of logical AND.
- The F-value was averaged over 100 independent random inference networks to reduce the variance of the results.

## 6.3 Preliminary Results

The results obtained thus far are very preliminary. Figure 12 shows the variation of the average F-value with the AND-OR bias. The error bars show the standard error. Damage spreading appears to be significantly worse as logical AND becomes more prevalent in the random inference network (i.e. rules 1 to 8 in Table 5 are more prevalent). Figure 13 shows the average number of output propositions for the GT random inference network versus the AND-OR bias. The number of output propositions varies linearly with the bias, with a significantly greater number of output propositions as logical OR becomes more prevalent (i.e. rules 9 to 16 in Table 5 are more prevalent). These results make sense given that:

1. For random inputs, logical OR is seven times more likely to produce a TRUE output compared with logical AND. Note in Figure 13 that the maximum number of output propositions (when bias = 0) is roughly seven times greater than the minimum (when bias = 1).
2. When the output of logical AND is TRUE, it will switch to FALSE if any of its inputs change.
3. When the output of logical OR is TRUE, this output is more robust to changes in the inputs. Thus rules based on logical OR can be expected to be more robust to perturbations.

Figure 14 shows the variation of the average F-value with the number of input predicates per rule that is effectively the connectivity of the random inference networks. (The AND-OR bias was set to the baseline value of 0.5 for these results, i.e. neither logical AND or logical OR was favoured during rule creation.) It is interesting that the damage spreading is worse as the number of input predicates per rule decreases from three to one. The random inference networks discussed in this report are very similar to the NK networks discussed in Section 6.1 and references 26 to 32. However with the NK networks the damage spreading is worse as the connectivity parameter K increases from one to three; this is opposite to the result shown in Figure 14 for the random inference networks.

This apparent anomaly is explained in reference 32 where it is noted that NK networks with the connectivity parameter K > 2 can be driven into the ordered regime when canalysing Boolean functions are employed instead of randomly selecting from the full set of possible Boolean functions. If the full set of 256 possible Boolean functions were employed with the random inference networks, then the damage spreading should become worse as the number of input predicates increases from one to three. However the opposite result was obtained because the analysis used a subset of canalysing Boolean functions.

The key point from Figure 12 and Figure 14 is that the damage spreading in the random inference networks is dependent on both the connectivity and the type of Boolean functions employed, e.g. whether the functions are canalysing. The key question is whether the study of NK networks that has been pursued for almost 40 years can be leveraged to better understand damage spreading in random inference networks and more generally inference networks used for situation assessment.

Figure 15 shows the average number of output propositions for the GT random inference network versus the number of input predicates per rule. The number of output propositions appears to be independent of the number of input predicates per rule.

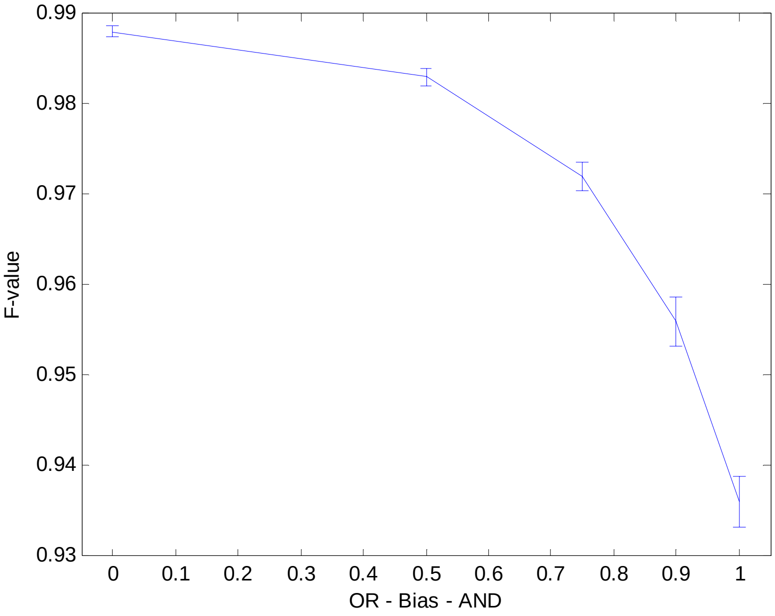


*Figure 12. The variation of the average F-value with the AND-OR bias*

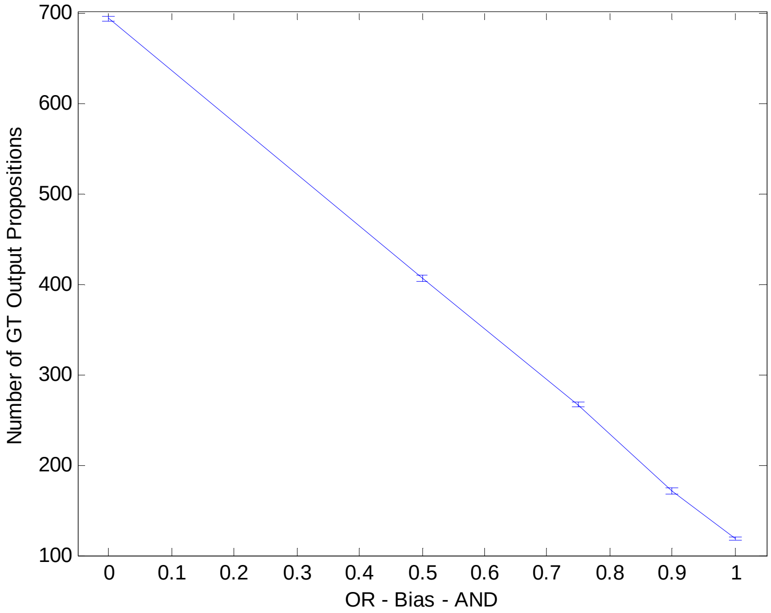


*Figure 13. The average number of output propositions for the GT random inference network versus the AND-OR bias*

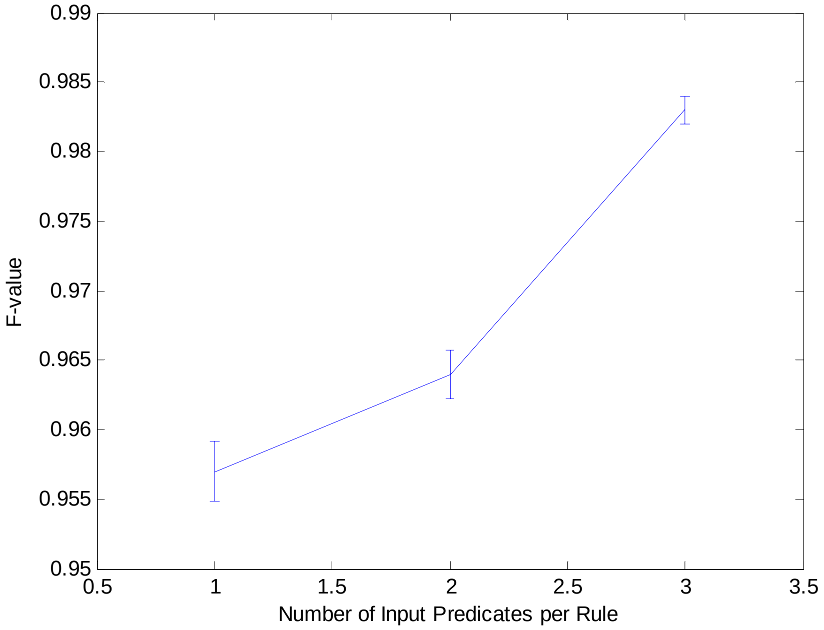


*Figure 14. The variation of the average F-value with the number of input predicates per rule*

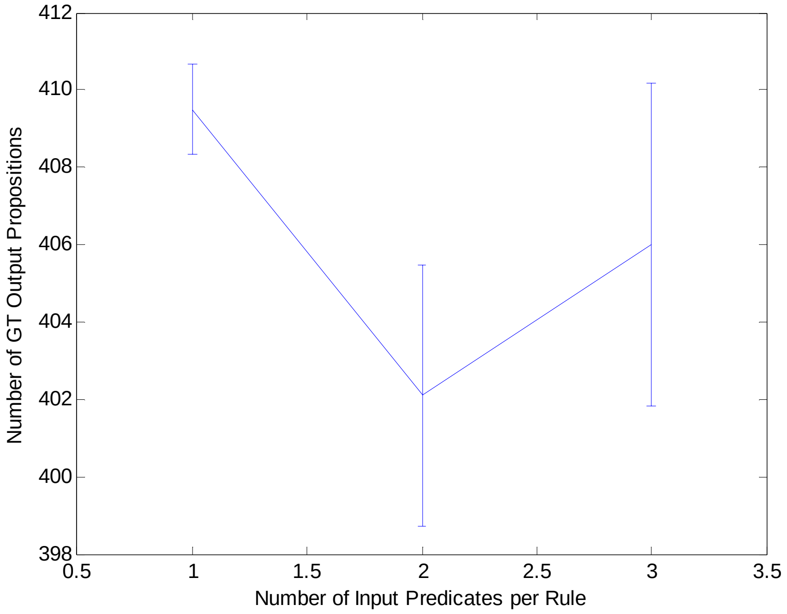


*Figure 15. The average number of output propositions for the GT random inference network versus the number of input predicates per rule*

### 6.4 Further work

The research literature on damage spreading in NK networks should be consulted to determine whether lessons learnt in that domain can lead to better understanding of damage spreading in inference networks used for situation assessment.

The random inference networks described in Section 6.2 should be further developed so that the full set of 256 possible Boolean functions is available for rule creation. This will allow confirmation that damage spreading in random inference networks with all Boolean functions available is similar to that in NK networks, i.e. damage spreading is worse with increasing connectivity.

A more extensive investigation of the variation of the F-value with the parameters in Table 6 should be undertaken in order to better understand when damage spreading is better or worse.

The analysis could be extended by considering perturbations applied to the rule-sets of the random inference networks, and not just the input propositions. This is akin to deficiencies in the inference rules used for machine-based situation assessment.

# 7. Conclusions

Under the umbrella of TTCP, algorithms are being developed for machine-based situation assessment to assist the human operator in complex situations. This report has discussed a technique that could be employed to evaluate the effectiveness of these algorithms in a simulation environment where the ground truth is well-defined. The technique involves an iterative process of aligning the situation assessment more closely to the ground truth, and is based on the F-value metric described in the scientific literature. This report also discusses techniques for performing partial matching of propositions when comparing the output propositions from the situation assessment and ground truth. Further work is required to test this evaluation technique prior to it being employed in TTCP experiments.

Errors in the input object assessment or the situation assessment algorithms can cause cascading errors through the inference networks. Complex networks have received attention recently in a diverse range of disciplines including the physical sciences, biological sciences, economics and sociology. One issue studied has been that of damage spreading through complex networks. This report has described how random inference networks were constructed to abstractly model the situation assessment process, and to examine how errors (damage) spread through such networks. The F-value metric was used to measure how a single perturbation spread through a random inference network. The results achieved thus far are only very preliminary, but they show that damage spreading depends on: (a) the degree of connectedness between the rules and other predicates in the network, and (b) the type of Boolean functions that are employed in the network. Further work is needed to produce a more complete set of results, and to leverage off the significant body of research into damage spreading in NK networks.

# 8. Acknowledgements

The authors gratefully acknowledge discussions with Martin Oxenham and Chris Nowak regards Sections 1 to 5. David Lingard gratefully acknowledges discussions with Anne-Marie Grisogono, Matthew Berryman, Alex Ryan and Vanja Radenovic regards Section 6.

# 9. References

1. Steinberg A. N., C. L. Bowman and F. E. White 1998, *Revisions to the JDL Data Fusion Model*, The Joint NATO/IRIS Conference, Quebec.
2. Llinas J., C. Bowman, G. Rogova, A. Steinberg, E. Waltz and F. White 2004, *Revisiting the JDL Data Fusion Model II*, Proceedings of the 7th International Conference on Information Fusion, Stockholm, Sweden.
3. Lambert, D.A. 2003, *Grand Challenges of Information Fusion*, Proceedings of the 6th International Conference on Information Fusion, Cairns Australia, pp. 213 – 219.
4. Nowak, C. 2003, *On Ontologies for High-level Information Fusion*, Proceedings of the 6th International Conference on Information Fusion, Cairns, Australia, pp. 657 – 664.
5. Lambert, D.A. and C. Nowak 2008, *The Mephisto Conceptual Framework*, DSTO Technical Report, in preparation.
6. Lambert, D.A. 2007, *STDF Model Based Maritime Situation Assessments*, The 10th International Conference on Information Fusion, Quebec, Canada.
7. Smith, D. and S. Singh 2006, *Approaches to Multisensor Data Fusion in Target Tracking: A Survey*, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 12.
8. Blanchette, M. 2004, *Military Strikes in Atlantis – A Baseline Scenario for Coalition Situation Analysis*, Technical Memorandum, Defence R&D Canada – Valcartier.
9. http://appliedwavelets.net/Documents/Terminology.html, Predicate entry, accessed 11th August 2008.
10. Mahoney, S. M., K. B. Laskey, E. Wright, K.C. Ng 2000, *Measuring Performance for Situation Assessment*, Information Extraction and Transport, Inc., Arlington, VA 22209.
11. Schrag, R.C., M. Takikawa, P. Goger, and J. Eilbert 2007, *Performance Evaluation for Automated Threat Detection*, submitted to JAIF.
12. Wang, J., Z. Ding, C. Jiang 2006, *GAOM: Genetic Algorithm based Ontology Matching*, 2006 IEEE Asia-Pacific Conference on Services Computing., South China Univ. of Technol., Guangzhou, Guangdong, China, IEEE Comput. Soc.
13. van Elst, L. and M. Kiesel 2004, *Generating and integrating evidence for ontology mappings*, Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, EKAW 2004, Proceedings (Lecture Notes in Artificial Intelligence Vol. 3257), The Univ. of Southampton , Springer-Verlag, Berlin, Germany.
14. Lambrix, P. and H. Tan 2006, *SAMBO - A system for aligning and merging biomedical ontologies*, Web Semantics, Vol. 4, No. 3, pp 196-206.
15. Lyttleton, O., D. Sinclair, D. Tracey 2005, *Mediating between Heterogeneous Ontologies Using Schema Matching Techniques*, Proceedings of the 2005 IEEE International Conference on

Information Reuse and Integration, Las Vegas, NV, USA, IEEE Syst., Man and Cybernetics Soc.

16. Castano, S., A. Ferrara, S. Montanelli, G. Racca 2004, *Matching techniques for resource discovery in distributed systems using heterogeneous ontology descriptions*, Proceedings ITCC 2004, International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, IEEE Comput. Soc.
17. Wang, J., F. Ali, R. Appaneravanda 2005, *A Web service for efficient ontology comparison*, Proceedings 2005 IEEE International Conference on Web Service, Orlando, FL, USA, IEEE Computer Society.
18. Wachter, T., A. Wobst, M. Schroeder, H. Tan, P. Lambrix 2006, *A Corpus-driven Approach for Design, Evolution and Alignment of Ontologies*, Proceedings of the 2006 Winter Simulation Conference, IEEE.
19. Doan, A., J. Madhavan, R. Dhamankar, P. Domingos, A. Halevy 2003, *Learning to match ontologies on the Semantic Web*, The VLDB Journal, Vol. 12, pp 303–319.
20. Richardson, B. and L. Mazlack 2005, *Approximate Metrics For Autonomous Semantic Web Ontology Merging*, Proceedings of the IEEE International Conference on Fuzzy Systems, Reno, NV, USA, IEEE Neural Networks Soc.
21. Bertsekas, D.P. 1992, *Auction Algorithms for Network Flow Problems: A Tutorial Introduction*, Computational Optimization and Applications, Vol. 1, pp. 7-66, available at http://web.mit.edu/dimitrib/www/Auction_Survey.pdf
22. Jonker, R. and A. Volgenant 1987, *A Shortest Augmenting Path Algorithms for Dense and Sparse Linear Assignment Problems*, Computing, Vol. 39, pp. 325-340.
23. Drummond, O.E., D.A. Castanon, and M.S. Bellovin 1990, *Comparison of 2-D Assignment Algorithms for Sparse, Rectangular, Floating Point, Cost Matrices*, Journal of the SDI Panels on Tracking, Institute for Defense Analyses, Alexandria, VA, Issue No. 4, pp 4-81 to 4-97, Dec. 15.
24. Bar-Shalom, Y. and W.D. Blair (Editors) 2000, *Multitarget – Multisensor Tracking, Volume 3: Applications and Advances*, Artech House, Inc., Norwood MA U.S.A.
25. Wang, X.F. and G. Chen 2003, *Complex Networks: Small-World, Scale-Free and Beyond*, IEEE Circuits and Systems Magazine, First Quarter, page 6.
26. Kauffman, S.A. 1969, *Metabolic stability and epigenesis in randomly constructed genetic nets*, Journal of Theoretical Biology, Volume 22, page 437.
27. Kauffman, S.A. 1993, *The Origins of Order: Self-organisation and Selection in Evolution*, Oxford University Press, New York.
28. http://fias.uni-frankfurt.de/~willadsen/RBN/ accessed April 2008.
29. http://www-users.cs.york.ac.uk/susan/cyc/n/nk.htm accessed April 2008.
30. Rohlf, T., N. Gulbahce, C. Teuscher 2007, *Damage Spreading and Criticality in Finite Random Dynamical Networks*, Physical Review Letters, Volume 99, 248701.
31. Yuan, B., and B-H Wang 2007, *Growing directed networks: organisation and dynamics,* New Journal of Physics, Volume 9, 282.
32. Kauffman, S.A. 2003, *Understanding genetic regulatory networks*, International Journal of Astrobiology, Volume 2, Issue Number 2, pp. 131–139.

Page classification: UNCLASSIFIED

# DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
# DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

Evaluation of the Effectiveness of Machine-based Situation Assessment – Preliminary Work

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

| | |
|---|---|
| Document | (U) |
| Title | (U) |
| Abstract | (U) |

4. AUTHOR(S)

David M. Lingard and Dale A. Lambert

5. CORPORATE AUTHOR

DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

| | | | |
|---|---|---|---|
| 6a. DSTO NUMBER<br>DSTO-TN-0836 | 6b. AR NUMBER<br>AR-014-253 | 6c. TYPE OF REPORT<br>Technical Note | 7. DOCUMENT DATE<br>August 2008 |

| | | | | |
|---|---|---|---|---|
| 8. FILE NUMBER<br>2008/1094363/1 | 9. TASK NUMBER<br>07/251 | 10. TASK SPONSOR<br>CDS | 11. NO. OF PAGES<br>29 | 12. NO. OF REFERENCES<br>32 |

13. URL on the World Wide Web

http://www.dsto.defence.gov.au/corporate/reports/DSTO-TN-0836.pdf

14. RELEASE AUTHORITY

Chief, Intelligence, Surveillance & Reconnaissance Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

16. DELIBERATE ANNOUNCEMENT

No limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS

Complex environments
Algorithms
Machine theory

19. ABSTRACT

The Information Fusion Panel within The Technical Cooperation Program (TTCP) is developing algorithms to perform machine-based situation assessment to assist human operators in complex situations. This report proposes a technique to measure the effectiveness of these algorithms in a simulation environment where ground truth is well-defined. In addition, this report models the situation assessment algorithms abstractly using random inference networks, and examines how errors (damage) spread through the inference networks. This models deficiencies in the object assessment as input to the situation assessment algorithm.

Page classification: UNCLASSIFIED