Detecting Deception by Analysis of Competing Hypotheses

Christopher Elsaesser Frank J. Stech
The MITRE Corporation
7515 Colshire Drive
McLean, Virginia 22102-7508

Abstract

This paper describes the central component of a system to assist intelligence analysts detect deception. We describe how deceptions exploit cognitive limits and biases and review prior work on processes that can help people recognize organized deceptions. Our process is based on Heuer's Analysis of Competing Hypotheses, which we automate by generating state-based plans and converting them to Bayesian belief networks. Our decision aid uses a concept from Bayesian classification to identify distinguishing evidence that a deceiver must hide and a counter-deceiver must uncover. We illustrate the process with one of the most important deceptions of the 20th Century.

1 INTRODUCTION

Deception is ubiquitous, ranging from the common (magic, financial fraud and scams) to the famous (e.g., D-Day; Indian nuclear tests [CIA 1998]). Complex stratagems, even though they have many opportunities to fail, often fool even those on guard against deception. Nevertheless, we think it is possible to construct a system to help intelligence analysts detect deceptions. This paper describes the central component of such a system.

We begin by noting how deceptions exploit cognitive limits and biases. Next we describe prior work that guides our approach. Our model is based on a process called Analysis of Competing Hypotheses (ACH). Section 4 describes how we automate ACH by generating state-based plans which represent alternate hypotheses and convert the plans into belief networks. Using a modified type of Bayesian classification, our decision aid identifies distinguishing evidence that a deceiver must hide and a counter-deceiver must uncover. We describe how a prototype system identified the key to one of the most important deceptions of the 20th Century. We conclude by listing areas for further research.

2 WHY DECEPTION WORKS

> *There is no need to sally forth, for it remains true that those things which make us human are, curiously enough, always close at hand. … we have met the enemy, and not only may he be ours, he may be us.*
> – Walt Kelly (1913-1973)

Effective deceptions exploit reasoning errors, cognitive limitations, and concomitant biases. The most important of these are:

- Reasoning from evidence to hypotheses
- Failure to entertain a deception hypothesis
- Biased estimates of probabilities
- Failure to consider false positive rates of evidence

The first two involve considering too few alternative hypotheses due to incomplete generation or premature pruning (which may involve misestimates of probabilities). The sources and effects of biases arising from mental estimates of probabilities are well known [Gilovich 2002]. We are particularly concerned with bias due to tunnel vision: making conclusions that support preconceptions, and "mirror imaging": assuming an adversary is likely to choose a course of action that appeals to the observer.

To recognize deception one must consider many alternatives and overcome biases that lead to inappropriately weighing evidence that seems to support one of only a few alternatives. The next section reviews work related to these aims.

3 RELATED WORK

A major contributor to susceptibility to deception is biased interpretation of observations [Dawes 2001]. Several techniques to reduce bias in probabilistic assessments have been investigated [Elsaesser 1989]. The most promising method is to require a subject to perform and document a systematic analysis of evidence [Fischhoff 1982]. But when not carefully applied, this process can sometimes make one more susceptible to deception.

Dragoni, et al. [Dragoni 1996] used a Bayesian approach in a decision aid for judicial proceedings to help assess witness deception. Abduction is used to eliminate information of low credibility and find maximally consistent subsets of evidence. Dragoni's technique helps in cases of common crime, where extensive coordination of deceptive testimony is unlikely. The technique is less useful against coordinated deception where plotters ensure that the evidence of many controlled sources will be confirmed with supposedly objectively (or even actually) verifiable information.

Johnson et al. [Johnson 2001] observed forensic accountants while they examined questionable business records. Protocol analysis indicated accountants who were best able to detect fraudulent information in financial statements used four processes:

- *Activation:* detect inconsistencies between expectations and observations of the environment.
- *Detection:* produce hypotheses about possible deceptive manipulations of the environment and adjust the assessments of evidence to reflect possible deception tactics.
- *Editing:* edit the initial hypotheses based on the deceptive manipulations and re-assess observations.
- *Reevaluation:* decide on appropriate actions to test the deception hypotheses.

Recent suggestions for training intelligence analysts to detect deception are consistent with Johnson's cognitive model. Whaley & Busby's Congruity Theory & Ombudsman Method [Whaley 2002] identifies information that must be collected to reveal inconsistencies and other cues to deception. R.V. Jones's Theory of Spoof Unmasking [Jones 1995] describes how to check the validity of evidence, highlight inconsistencies, and develop deception hypotheses. Heuer's Analysis of Competing Hypotheses (ACH) [Heuer 1999] specifies how to consider inconsistent and anomalous information, develop competing hypotheses (including deception), and test hypotheses in a manner that reduces susceptibility to cognitive limits and biases. ACH is the basis of the decision aid we report in this paper. ACH consists of the following steps:

1. Identify the possible hypotheses to be considered.
2. List the significant evidence and assumptions for and against each hypothesis.
3. Draw tentative conclusions about the relative likelihood of each hypothesis.
4. Analyze sensitivity of the conclusion to critical items of evidence.
5. Identify future observations that would confirm one of the hypotheses or eliminate others.

Summarizing prior work, we know how and why deception succeeds and procedures to detect deception. Since teaching intelligence analysts the procedures does not seem to produce consistently effective deception detectors, a decision support system seems necessary. The next section describes a prototype of a key part of such a system.

4. AUTOMATING ACH

This section describes how we use state-based planning and Bayesian belief networks to automate Heuer's Analysis of Competing Hypotheses (ACH) in an attempt to overcome cognitive limitations and biases that make people susceptible to deception.

4.1 HYPOTHESIS GENERATION VIA AUTOMATED PLANNING

Step one of ACH is to develop alternate hypotheses about an adversary's course of action. This is intended to help the subject consider alternate explanations of evidence and avoid prematurely making conclusions based on a few salient observations or preconceptions. We use a domain independent task decomposition planning system called Adversarial Planner (AP) [Applegate 1990] to automate as much hypothesis generation as is practical. Here we describe the parts of AP that relate directly to ACH and conversion to Bayesian belief networks.

Task decomposition planning starts with an abstract goal, refines it with successively more concrete (less abstract) subgoals, and terminates when a sequence of atomic actions is found. Subgoals come from the "expansion" specification in action templates that are the raw material of planning. Figure 1 shows a typical action template. Figure 5 gives an example of a template for an atomic action, that is, one with no further decomposition.

```
(define (action transport)
   :parameters (?force_module - force_module
                ?destination - destination
                ?conveyance - conveyance)
   :constraints ((= (get-value ?force_module 'location)
                    (get-value ?conveyance 'location)))
   :expansion (series
                (parallel (contains ?conveyance ?force_module)
                          (adequate_fuel ?conveyance))
                (location ?conveyance ?destination)
                (contains ?conveyance nothing))
   :effect (location ?force_module ?destination)
   :documentation "Load, move, unload")
```

Figure 1: A task decomposition action template.

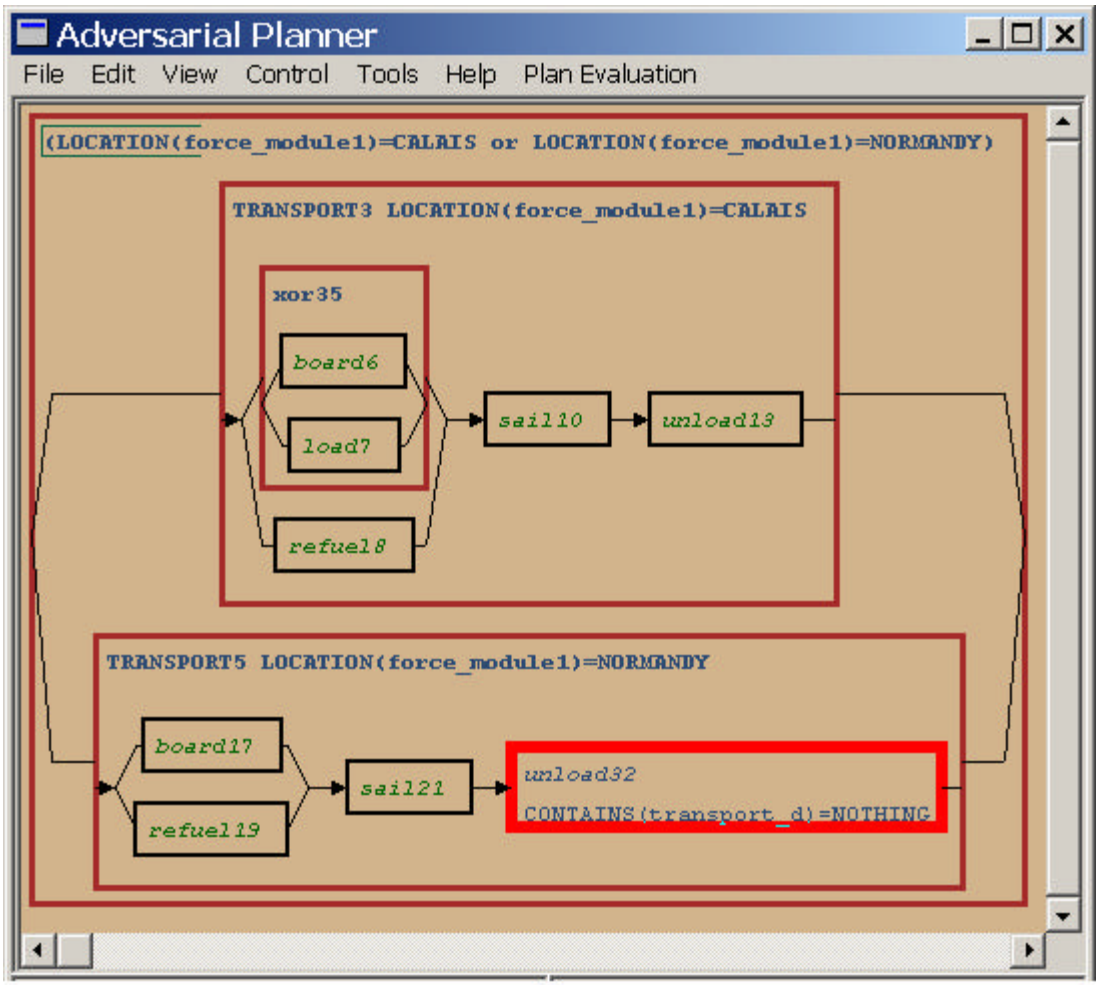

Figure 2: Simple contingency plan. Highlight on action unload32 indicates it depends on an assumption.

During plan generation, if an :effect of a template unifies with a subgoal,[1] the action's :expansion tells the planner *what* to do to accomplish the :effect, but not *how* those subgoals are to be accomplished. The planner can consider alternative methods of accomplishing the subgoals. AP attempts to expand each alternative, creating a contingency plan when more than one action can fulfill a subgoal. This is intended to help the user consider all the alternatives (Figure 2).

AP computes a temporal model of the plans it generates. The model consists of numerical time points for each action's earliest and latest start and end times based on dynamic estimates of action duration and the temporal relations among subgoals noted in each abstract action's :expansion. Any of the temporal relations in [Allen 1984] may be used. The :expansion in the template in Figure 1 has two temporal relations. "Series" means the subgoals that follow have to be accomplished in the order listed. "Parallel" means that the enclosed subgoals may be accomplished in any order. In a deception example, a diversionary action might "cover" the beginning of an attack. When AP generates contingency subplans, the temporal information is used to determine if the alternate activities are mutually exclusive.

AP allows variables to designate resources. For example, the action in Figure 1 stipulates that some "conveyance" is available to transport a "force_module." Depending on the resources available, it often is not necessary to settle on a particular conveyance until the plan is prepared for execution. Since our application is concerned with what might happen, rather than planning for a specific outcome, the belief network made from a plan represents all the possible assignments of a designator. To generalize this for ACH, we extended AP so that there need be no identified resources to fulfill all plan parameters. When this happens, the planner simply notes that resources of particular types are required. In the analysis application,

[1] The :effect can be a conjunction of propositions and those that do not unify with the subgoal become side effects of the action.

observing such a resource increases the probability that alternative is viable.

AP allows user-supplied estimates of the probability an action will establish the subgoal it was put in the plan to fulfill, given that all the action's preconditions hold. We impose a condition that the probabilities cannot be 0.0 or 1.0 to preclude premature pruning.

Conventional planners cannot generate a plan if any of the preconditions of the actions required to establish a subgoal do not hold in the initial situation and cannot be accomplished by a planned action. This is problematic when one is uncertain about the disposition and capacities of an adversary. To address ignorance, AP can *assume* preconditions not listed in the input situation (assessment of the current state) and capabilities (actions). This allows AP to develop competing hypotheses with incomplete knowledge and alternatives a user might not consider. Assumptions – if confirmed – tend to be key indicators to the adversary's possible course of action.

ACH suggests that an analyst entertain *all* possible hypotheses. AP can generate many alternate plans, including alternate subplans within a single plan. Contingency planning in concert with the ability to make assumptions can cause combinatorial explosion. This has not been a problem on proof-of-concept domains with which we have experimented, but is an issue for further research.

With the capabilities listed above, AP is able automatically to generate competing hypotheses, fulfilling steps 1, 2, and 3 of ACH. Figure 2 shows a very simple example that we will discuss in the remainder of this paper.

4.2 CONVERTING PLANS TO BELIEF NETWORKS

A preliminary version of our process for converting plans to a belief networks was described by Seligman, et al. [Seligman 2000]. Here we recap the process and describe extensions made for this application.

A plan is a partially ordered sequence of actions. Each action has an input situation and causes (if it succeeds) a subsequent output situation. Situations are sets of propositions, which are relations on objects. Each action is represented as a node with two states: succeed and fail. The predecessors of action nodes are the nodes representing the action's preconditions. The states of the nodes are the possible values of the relation on the arguments. Most domain representations are purposely sparse and only a few of the input situation's propositions are changed by action execution. The domain representation is usually constructed so that propositions have a finite domain and range, although this is not necessary.

Action node beliefs are computed based on the states of the precondition nodes and the user-specified estimate of the probability of success of the action. Action nodes are predecessors of nodes representing their effects. Under typical persistence assumptions [Shoham 1988], a proposition's value persists until an action changes it. Thus, failure of an action means that the value of the proposition after the action was scheduled to execute would be the same as its value at the latest corresponding node before the action. Hence, a typical propositional node has two parents.[2] A representative segment of a belief network made from a plan is depicted in Figure 3.

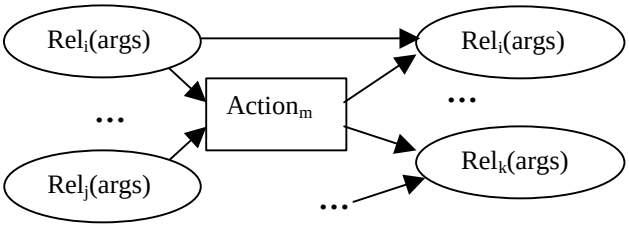

Figure 3: Segment of a belief network from a plan

AP's extensions are uncomplicated to represent in a belief network. Contingency nodes are treated as disjunctions, and will be exclusive disjunctions if temporal and resource constraints indicate that children subnodes are mutually exclusive. Designators that represent possible assignments become parent nodes of the actions where they originate. Designator nodes are treated the same way as action precondition nodes in the sense that they must take on legal combinations of values for the action to execute. The domain and range of these nodes is determined by constraints imposed by the actions that use the resources. For example, if you want to transport troops, then the conveyance should not be a tanker. Finally, assumptions are treated like propositions without predecessors and have a default probability. Setting the default to a low probability, say 0.10, ensures that analysis will indicate all but the most trivial assumption as crucial to the hypothesized outcome.

4.3 IDENTIFYING KEY INDICATORS

Steps 4 and 5 of ACH require identifying indicators of an adversary's intention. This is where ACH is susceptible to bias when people, as they often do, fail to weigh the impact of evidence by its false positive rate [Dawes 2001] and misestimate prior probabilities. To avoid these errors, we treat each state in the network as a potential two-category dicotomizer [Duda 2001]. The minimum error rate discriminant for a two-category dicotomizer is:

[2] We allow the possibility of decay into a state of ignorance for propositions with a range of unknown cardinality.

$$g(e_i) \; ? \; \ln \frac{P(e_i \mid ? \;)}{P(e_i \mid ? \;)} \; ? \; \ln \frac{P(? \;)}{P(? \;)}$$

Pattern classification doctrine would have us compute g(e_i) and apply the decision rule that when g(e_i) > 0 and e_i is observed, then ? is more likely, its complement otherwise. We do *not* compute the second term of the discriminant, the prior log likelihood of the outcome ? . It is not necessary for focusing attention on the most important evidence of the alternatives, and by ignoring it we avoid relying on potentially biased priors.

Summarizing, the ACH as we have implemented it, consists of the following steps:

0. Generate a contingency plan representing one or more hypotheses about the possible course of action of an adversary (Figure 2)
1. Create a belief network from the plan (Figure 3)
2. Enter a finding ? ?-- typically the success state of the plan or one of the branches of a contingency plan.
3. Store the conditional probability of all states for only those nodes that precede the hypothesized outcome state ? , P(e_i |??). This is done since we are trying to identify evidence that will indicate the adversary's intent before it can be accomplished.
4. Remove the finding ? ?and enter a finding of its complement? .
5. For each state e_i compute the log likelihood ratio, ln[P(e_i |??)/P(e_i |?)],For human factors purposes, we scale the likelihood ratios to (-1,1).
6. Apply a threshold to eliminate states that provide little evidence to distinguish between ? ?and ? . On a (-1,1) scale, a threshold of +/- 0.05 is typical.
7. Display the results, as in Figure 4.

The closer a state's (scaled) log likelihood is to 1.0 or -1.0, the more diagnostic that state is of ? ?or? . These are states the deceiver must hide, as revealing them should lead the deceived to recognize the true course of action. Conversely, the counter-deceiver must look for these states.

This process outlined in this section addresses the main sources of bias interferes with people's ability to recognize deception. The next section gives an example to illustrate how it might be used for counter-deception.

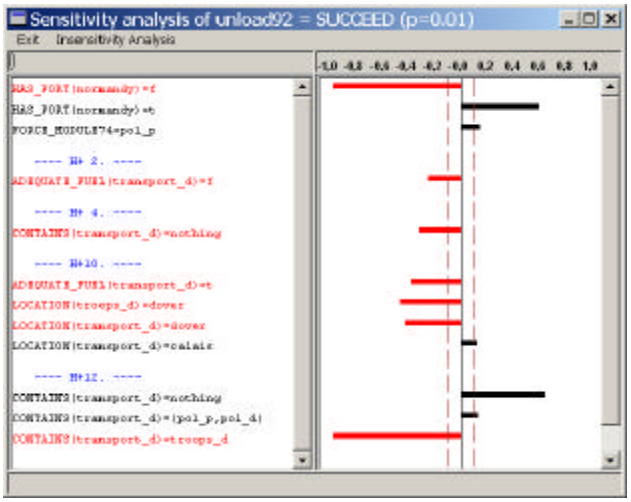

Figure 4: Analysis of Normandy as objective of D-Day invasion is most sensitive to having port facilities.

5 EXAMPLE: D-DAY

The D-Day invasion of France via Normandy was the turning point in World War II. A key factor in its success was the Allied campaign of deception that played on Germany's predisposing that the invasion would come via the Straits of Dover in the vicinity of Pas de Calais. Pas de Calais was considered a favorable landing site for the invasion, as it was on the most direct route to Germany, minimized flying time for air cover, and would help the Allies avoid the threat of V-1 flying bombs. An important consideration influencing the German assessment was that an invasion requires port facilities to offload troops and supplies. The Pas de Calais region had four major ports; Normandy had none.

Allied planners knew they could not break through Germany's defensive forces if the German Army concentrated where the Allies choose to land. Therefore, the Allies had to convince the Germans to defend some point other than the true Allied objective. Pas de Calais was the obvious invasion objective, so they had either to convince the Germans of another, or find an alternative landing site and deceive the Germans so that they would not consider the true destination a serious objective. As we know from history, the latter is what happened.

Allied planners embarked on a deception campaign called BODYGUARD whose purpose was to reinforce the German preconception of an Allied landing at Pas de Calais. They carefully hid the key fact that the Allies would not need to capture a major port because they had built transportable port facilities, called MULBERRY, to use on the Normandy beaches and had devised the first undersea oil cables, PLUTO (pipelines under the ocean).

MULBERRY and PLUTO headed the Allied list of "items which it is undesirable" for the enemy to see. We implemented a simple version of BODYGUARD to illustrate our ACH process.

We started with an existing planning domain description for the transportation of supplies. The key action represents unloading a ship at a destination, depicted in Figure 5.

```
(define (action unload)
   :domain transportation
   :parameters (?conveyance - ship
                ?destination - destination
                ?force_module – force_module)
   :precondition (and (location ?conveyance ?destination)
                      (contains ?conveyance ?force_module)
                      (has_port ?destination))
   :effect (contains ?conveyance nothing)
   :probability-of-success 0.95
   :duration 2.0)
```

Figure 5: Unload action from transportation domain

We created an initial situation with the relevant facts of BODYGUARD. From these parts we planned the transportation of invasion supplies from the south of England to alternate destinations in France.

The first plans we generated were from the Allied point of view and indicated nothing remarkable; with port facilities you can supply an invasion at either Normandy or Pas de Calais, sensitive only to the usual mundane items such as availability of sufficient transport. But the Germans did not know about MULBERRY or PLUTO, so we deleted the proposition **has_port(Normandy)** from the initial situation,[3] to represent the German preconception. The result was a single feasible plan with Pas de Calais as the destination – the upper branch of the contingency plan in Figure 2 – just as the Germans concluded.

The German High Command did not have our tool. If they had it, the next step would be to allow AP to make assumptions for preconditions that can't be accomplished with actions. The relevant part of the analysis on the plan in Figure 2 is shown in Figure 4. The top two lines indicate that the decisive factor that would make an invasion at Normandy feasible is port facilities. This is what the Allies knew to hide. The Germans should entertained this deception hypothesis and tried to determine if the Allies could establish its key precondition.

[3] alternately, we could have set its probability to 0.0

6 CONCLUSION AND FURTHER RESEARCH

We have described a system that automates Heuer's ACH as the basis of a counter-deception decision support system. Our effort now is on extending the core ACH process. On the front we will build an interface to help users create domain descriptions. The planning system will fill in these plans and create contingencies as alternatives.

On the back end of our ACH we are creating a system to suggest deception tactics to keep an adversary from recognizing the true plan (dissimulation) and ways to give the adversary a false apprehension of reality (simulation). The temporal model generated with the alternate courses of action will be an important input to this process.

After we complete a deception planning system we will extend it to counter-deception planning using AP's counter-planning process. Along the way, we will conduct experiments to see if our system can (a) reliably plan deceptions, and (b) reliably detect deceptions.

Acknowledgements

The research reported in this paper is sponsored by The MITRE Corporation.

References

Allen, James. F. (1984) "A General Model of Action and Time" *Artificial Intelligence* 23, 2, July 1984.

Applegate, Carol, C. Elsaesser, and J. Sanborn (1990) "An Architecture for Adversarial Planning," *IEEE Transactions on Systems, Man, and Cybernetics*, Volume 20, Number 2, January, 1990.

CIA (1998), *Press Release: Indian Nuclear Testing.* www.cia.gov/cia/public_affairs/press_release/archives/1998/pr051298.html

Dawes, Robyn M. (2001) Everyday irrationality: how pseudo scientists, lunatics, and the rest of us systematically fail to think rationally. Westview Press.

Dragoni, A. F. (1996) "Maximal Consistency, Theory of Evidence and Bayesian Conditioning in the Investigative Domain;" *Proceedings of the Fifth Iberoamerican Confernece on Computer Science and Law*, Havana.

Duda, Richard O., P.E. Hart, and D. G. Stork (2001) *Pattern Classification*. Second edition, John Wiley & Sons, Inc.

Elsaesser, Christopher (1989) "Explanation of Probabilistic Inference," *Uncertainty in Artificial Intelligence 3*, L.N. Kanal, T.S. Levitt, and J.F. Lemmer

(Editors), Elsevier Science Publishers B.V. (North-Holland), pp. 387-400.

Fischhoff, Baruch (1982) Debiasing. *Judgement underuncertainty: Heuristics and biases*, Daniel Kahneman, Paul Slovic, and Amos Tversky (Editors), Cambridge University Press, Cambridge, United Kingdom, pp. 422-444.

Gilovich, Thomas, D. Griffin, and D. Kahneman (2002) *Heuristics and Biases* Cambridge University Press, Cambridge, United Kingdom.

Heuer, Richards J. (1999) *Psychology of Intelligence Analysis.* Washington: Central Intelligence Agency Center for the Study of Intelligence.

Johnson, Paul E., S. Grazioli, K. Jamal, and R. G. Berryman (2001) "Detecting deception: Adversarial problem solving in a low base-rate world," *Cognitive Science* 25(3), May-June.

Jones, R. V. (1995) "Enduring principles: Some Lessons in Intelligence," *CIA Studies in Intelligence*, Volume 38, Number 5.
http://www.cia.gov/csi/studies/95unclass/Jones.html

Seligman, Leonard J., P. Lehner, K. Smith, C. Elsaesser and D. Mattox (2000) "Decision-Centric Information Monitoring," *Journal of Intelligent Information Systems*, Kluwer Scientific Publishers, 14(1).

Shoham, Yoav (1988) *Reasoning about change*, The MIT Press, Cambridge, MA, USA.

Whaley, Barton and J. Busby (2002), "Detecting Deception: Practice, Practitioners, and Theory" in Roy Godson and James J. Wirtz (Editors.), *Strategic Denial and Deception: The Twenty-First Century Challenge*. New Brunswick: Transaction Publishers.