Analysis of the Effect of Disinformation on Decision Model for Effective Detection and Usage of Disinformation in Adversarial Environment[1]

Submitted for *the 2005 Command and Control Research and Technology Symposium*

Norman K. Ma[2] Michael S. Miller[3]

Department of Computer Science
Texas A&M University,
College Station, TX 77843-3112
normankma@tamu.edu mmiller@cs.tamu.edu

Summary

Disinformation is a process that spans operational scope and battle domains. We investigate the nature of disinformation by applying it in an abstract and flexible representation called State Space Representation (SSR). SSR is rich enough to represent physical, information, and cognitive military domain objects. This representation is a solid grounding for the diverse knowledge of command and control research. In addition, ample tools are available[4] for both theoretical and engineering tasks to solve state space problems.

This paper demonstrates the utilization of the State Space Representation with an algorithmic implementation of the OODA loop and describes the conditions of disinformation in the algorithm. We conclude that disinformation will not result in sudden unexpected state changes if the situation assessment task is carried out every OODA cycle.

One of the utility of using State Space Representation is the possibility to create a Disinformation Evaluation Assistant (DEA) to help provide command and control capabilities closer to the tactical levels to anticipate the future of battle trend.

Introduction

A young lion quickly learns that making a straight line for the pack of impalas on the Serengeti Plain will not catch a deer to satisfy its hunger. A grown lion uses deceptive actions to catch its prey. Higher-ranking chess players pursue parallel strategies to affect relative advantage, and we have all seen fighters in the boxing ring shifting from right to left as a standard fighting procedure. Players of card games such as poker put on their best Poker Face and execute actions with intent to deceit. Disinformation permeates all human activities, in peace and in war, when playing and when working, and above and

---

[1] This work is being supported in part by the U. S. Air Force Office of Scientific Research under MURI grant #F49620-00-1-0326 and the Department of Computer Science, Texas A&M University

[2] PhD student

[3] PhD candidate

[4] The state space paradigm is used in many fields and mathematicians and computer scientists have developed many tools that operate on state space representation.

below the conscious level. Disinformation and deceit has been investigated by Machiavelli's The Prince, the Chinese Thirty-Six Stratagems, and more scientifically by the U. S. armed forces' joint publication manual on Information Operations. We shall give this intriguing subject a scientific treatment from the computer science point of view using the state space representation [Winston93].

Command and Control is significant in achieving objectives in adversarial situations. Moreover, these conflicts will have different OODA (observe, orient, decide, act) [Shaw85] tempos and different domain focuses (physical, information, and cognitive). In military command and control, the scope of disinformation is categorized as tactical, operational, or strategic. These overlapping levels can be differentiated by the duration of the process, the geographical area of operation, and the richness of problem representation. We investigate the nature of disinformation in the state space problem representation (SSR). This will be done within the context of the OODA process.

Disinformation has gained greater importance with a trend away from large-scale military conflicts and a rise of very complex social and political impacts on all military operations. The days of First World nations being involved in large scale conventional conflicts without concurrent management of the social and political environment are gone. The manifestation of disinformation in operations of war depends on the national culture of that nation. Disinformation and information management have become a crucial central issue for any nation to control as an essential component of the warfighting operations.

The cognitive domain is the innate origin of disinformation. Due to the changing nature of cognitive objects, using state space to describe the cognitive domain is a reasonable choice. Mastering the cognitive domain is essential in permanently winning any battle or competition. As exemplified by the protagonist Winston at the end of the book "1984", that deep down, he does love Big Brother, the ideal ending states of any conflict reside in the cognitive domain, we can represent Winston's final cognitive state as:

$$state = \{loves(Big\_Brother, dog\_fifi, food, \cdots), \cdots\}.$$

This state seems to be satisfactory for both Winston and the Big Brother:

> Axiom 1, the final state, s_{final}, of an adversarial interaction is recognizable by both teams and contains no emittable actions by either of the teams.

A more historical example is the effect of the Marshall Plan at the end of World War II. The plan laid down an economic and political plan for the defeated nations and in doing so provided for an improved treatment of the defeated nations which induced a satisfactory ending state on those nations' psyche, thus arguably given the world an extended period of peace.

State Space Model of Information Operation

There are many levels at which disinformation can occur. As a start, we limit our investigation to the disinformation operators in an adversarial environment using a description based on state space. The state space abstraction is sufficient for formulating and analyzing information operations problems. This paradigm represents the problem solving process as a series of transitions from one state to another state in a universe called the state space. Actions enacted by the participants change the present state. For our purposes, we are interested in finding a final goal state that gives us an acceptable solution.

Using the state space paradigm enables the analysis of command and control operators, processes, and models in a common context. State space is conducive to graphical display, and graphical visualization of the state the execution of tasks gives the user a big picture of the present situation by showing the recent and likely tasks (or events) as near neighbor nodes:

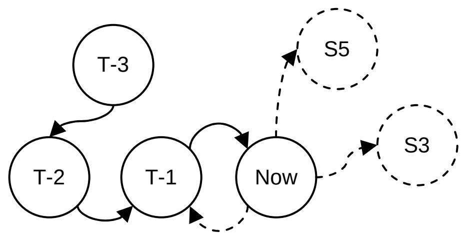

Figure 1 - State space representation gives a big picture understanding of the past, present, and possible futures

The state space model is a set of possible states $S = \{s_1, s_2, \cdots, s_n\}$ in which a state is a sequence of parametric values, $s_i =< p_1, p_2, \cdots, p_m >$. These values describe the objects that have been designated as relevant in a state. The design of a state space representation needs to balance between detail and operational effectiveness; with the goal of visualizing the state space will give a strategic view of the on-going process that include details to assist in making decisions. A hierarchical representation can be used to provide clarity in separating levels of decision-making.

The adversarial game is described by a sequence of changing states induced by actions emitted by either one of the teams, $s_{start} \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} \cdots s_n \xrightarrow{a_{n+1}} s_{end}$. This sequence is from one team's perspective. The opposing team's taxonomy of the environment is likely to be different:

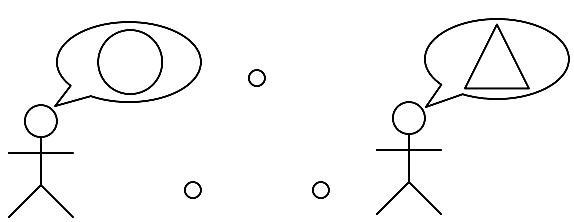

Figure 2 - Different representation of reality by opposing teams

The modern decision-making space is composed of objects from the physical, information, and cognitive domains. Objects are created, modified, expired, and destroyed in the on-going processes. This change of environment is described by state transitions within the problem space. One or more states within the model can also be labeled as goal states. The game ends when one of the mutually recognizable goal states has been reached. Below is a conceptual view of the problem space:

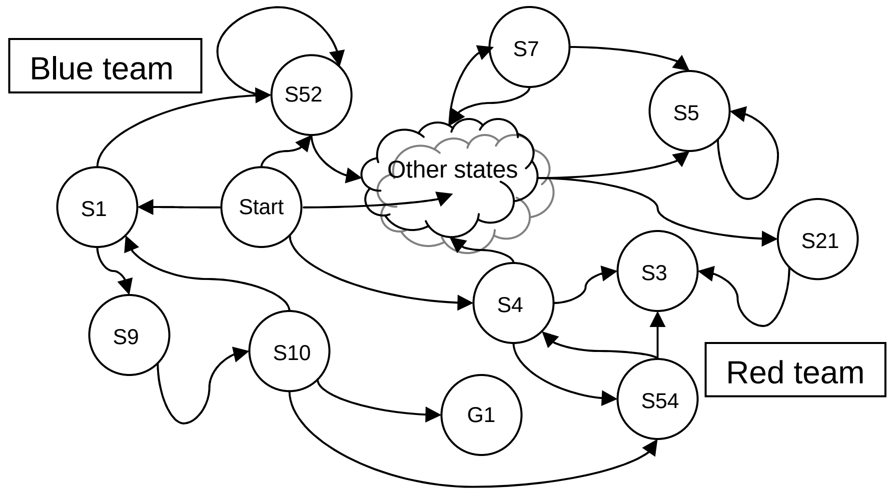

Figure 3 - Adversarial state space diagram

We demonstrate that information operations can be represented using the state space paradigm:

Axiom 2: Information Operations is a sub-process of the Decision-making process.

Axiom 3: Decision-making processes are represented through physical, informational, and cognitive objects.

Axiom 4: Information operations (IO) representation contains physical, informational, and cognitive objects.

Axiom 5: Informational and cognitive objects may represent physical objects.

Axiom 6: Physical objects can be represented by a set of parameters.

Axiom 7: Information Operation objects can be represented by a set of parameters.

Theorem 1: A state of the Information Operations problem space is composed of physical, information, and cognitive objects that can be represented by a set of parameters.

The size of the information operation space, $|S|$ where $S = \{s_1, s_2, \cdots, s_n\}$, depends on the nature of the parameters and should be selected to sufficiently represent reality but not be any more complex. Theorem 1 is the framework which we shall conduct our analysis of disinformation.

Operators of State Space Representation

The command and control process can be described using the OODA loop (observe, orient, decide, and act). While the OODA model was originally used to describe a fighter pilot's activity during an aerial dogfight, its four activities can also be used to describe longer cycle time adversarial activities such as battle campaigns, diplomatic maneuvering, or even commerce trade negotiation. The context of where the fighters fly or where the surface forces operate can be described using the State Space Model (SSM) abstraction. It is a set of possible states $S = \{s_1, s_2, \cdots, s_n\}$ where each state is describe by a sequence of parameter values $s_i =< p_1, p_2, \cdots, p_m >$.

In an adversarial game, each player carries out his own OODA process with the intention of reaching a goal state. A record of an on-going exchange can be described and recorded as a sequence of state transitions caused by the actions of the participants, such as the following sequence $s_{start} \xrightarrow{a} s_1 \xrightarrow{a} s_3 \xrightarrow{a} \cdots s_n \xrightarrow{a} s_{blue\_goal}$ where one of the blue team's goals has been reached after a sequence of state transitions $s_1, s_2, \cdots, s_n$ or an alternate outcome is possible as $s_{start} \xrightarrow{a} s_2 \xrightarrow{a} s_4 \xrightarrow{a} \cdots s_n \xrightarrow{a} s_{red\_goal}$ when a red team's goal has been reached. The size of the state space depends on the parameters that have been chosen in the model and the number of possible values for each of those parameters, $|S| = |P_1| \times |P_2| \times \cdots \times |P_m|$, and $s_i \in S \mid S = \{(P_1 \times P_2 \times \cdots \times P_m)\}$.

For example, if a state space model has been defined to contain four parameters and the possible values of these parameters are:

P_1: (high_traffic, average_traffic, low_traffic)
P_2: (high_influence, average_influence, low_influence)
P_3: (terrorist_attack, no_terrorist_attack)
P_4: (high_economy, average_economy, low_economy)

then there are $|S| = |P_1| \times |P_2| \times |P_3| \times |P_4| = 54$ states, $s_1 \cdots s_{54}$:

State	P_1	P_2	P_3	P_4
s_1	Low_traffic	Low_influence	No_terrorist_attack	Low_economy
s_2	Low_traffic	Low_influence	No_terrorist_attack	Average_economy
s_3	Low_traffic	Low_influence	No_terrorist_attack	High_economy
s_4	Low_traffic	Low_influence	terrorist_attack	Low_economy
s_5	Low_traffic	Low_influence	terrorist_attack	Average_economy
s_6	Low_traffic	Low_influence	terrorist_attack	High_economy
s_7	Low_traffic	Average_influence	No_terrorist_attack	Low_economy
s_8	Low_traffic	Average_influence	No_terrorist_attack	Average_economy

s_9	Low_traffic	Average_influence	No_terrorist_attack	High_economy
⋮	⋮	⋮	⋮	⋮
s_{54}	High_traffic	High_influence	Terrorist_attack	High_economy

Table1, Enumeration of all possible environmental states of 4-parameter state definition

The table above lists all the possible states. These states can also be represented in a graphical format similar to Figure 2.

Processes of State Space Representation

OODA Loop

In state space representation, a team's goals and objectives are represented as states. The arrows between the states are actions that can be emitted by the competing teams to change the environmental state. In a controlled game such as chess or football the actions can only be generated from one of the participating teams, thus a participant only need to execute its own plan and anticipate the opponent's plan. In real-world domains such as military encounters, political negotiation, and commerce interchange the number of participants is much greater. State transitions will be affected by external actions that are not under the control by either of the two primary participants. This characteristic is another factor supporting quick OODA process cycle time; a quick observe-orient-decide cycle will assure that the action generated is relevant to the current environment.

The state-space model description from the previous paragraph forms a good analytical foundation for disinformation operations. A team has to observe the adversary's actions, evaluate the choices, make a decision, and execute an action to respond to or influence the adversary through the environment. This sequence of activities is executed quickly with the intention of moving the competition context to one of its goal states. Each cycle of the OODA loop takes time $\omega = t_{observe} + t_{orient} + t_{decide} + t_{act}$. In addition to accuracy, adversarial OODA processes can be measured using a relative cycle time $\frac{\omega_{own}}{\omega_{opponent}}$. The goal is to keep this measure under 1. It is desirable to have shorter time cycle than the opponent so that the action emitted by the opponent will be rendered either irrelevant or ineffective due to change of environmental state induced by one's own action. This known as operating within the other side's decision loop:

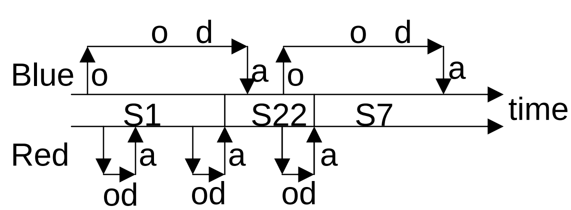

Figure 4 - Faster OODA execution controls state change in environment

Figure 4 demonstrates the evolving environmental state along a time line. The blue team at the top has a relatively longer OODA cycle than the red team. The red team is able to induce an environmental change from s_1 to s_{22} by its action from the second OODA cycle. At which time the action from the first cycle of the blue team is being introduced to an environmental state, s_{22}, which is different than the state, s_1, from which the observe processes and orient/situation assessment [Endsley95] processes have undertaken. The activities in Figure 4 can be recorded as $s_1 \xrightarrow{a_{red}} s_{22} \xrightarrow{a_{red}} s_7 \xrightarrow{a}$. Both of the state changes are caused by the red team's action. This indicates that the red team has the control of the situation and is now in a better position to move the environment into one of its goal states.

Each part of the OODA process receives information from the opponent, make sense of the observation, decide on a course of action, and then execute the selected action. The four OODA activities are not clearly separated in a human because the human brain is an internal process. However, distinguishing the separate OODA activities is beneficial for scientific experiments and also for limiting the project scope when building computational systems that enhances activities that are conducive to OODA modeling. The original application of OODA model is the detailed air-to-air combat described by Shaw [Shaw 85]. In a typical scenario as described by Shaw, both combat pilots execute their own OODA loop of observe, orient, decide, and action within a very short adversarial timeline. Each cycle of the loop is in seconds and the pilot who is able to assess the existing situation and produce an action quickest can assure a favorable result.

Disinformation describes a process where an individual emits incorrect information with the intention to mislead the receiver [Koohang03], $emitter \xrightarrow{action} receiver$. By misleading the receiver, the sender expects the receiver to generate actions that will be advantageous to him. Let $A = \{a_1, a_2, \cdots, a_m\}$ be the set of actions that are understood by both parties. These signals can be anything that is observable in the physical or the information domains: physical activities, communication among entities, or any environmental signals that have been deemed as possible indicators of the opponent's behavior. The emitter transmits these signals purposefully or as a side effect of on-going activities.

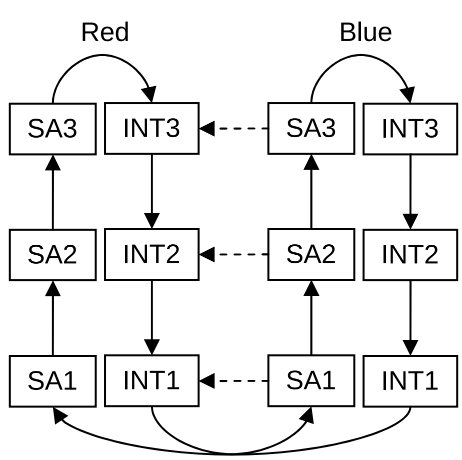

Diagram 1: Red and Blue OODA processes in the cognitive domain plan

Disinformation is the dissemination of information with intent to deceit. We explore a representation of disinformation that can be processed by a computing system. Using the state space representation described in the previous section, described below is a walk through of an algorithm using the OODA process within disinformation information operations. The red team has the option to emit misleading actions to degrade the blue team's information position [Alberts01]. In this case degrading the information position can be specifically defined as preventing the blue team from accurately modeling its plan. The Red team has goal states $S_{goal} = \{s_{g1}, s_{g2}, \cdots, s_{gn}\}$, is able to affect changes in the problem state by possible actions $A = \{a_1, a_2, \cdots, a_m\} = \bigcup_i a_i$, and has plan ψ,

Definition 1. A plan, ψ, is a sequence of states $< s_1, s_2, \cdots, s_r >$ where $s_r \in S_{goal}$.

The plan defined in this particular context is a sequence of states where the last one is a goal state. When interpreted in the war-fighting context, a plan contains a sequence of states. The immediate goal of the team is to reach the next state in the plan. The following listing describes a command and control assistant algorithm that is based on a simplified OODA process:

```
// Red team algorithm, disinformation distributor
// Input: Environmental observation
// Output: Recommended action
// Use state space representation
Loop
    If goal state reached
          Halt
    Observe
          Identify the present state using situation assessment
          If present state equals to a goal state then
                  Housekeeping
                Repeat Loop
    Orient
          If opponent changed plan
                new_plan ← Identify opponent's plan
                opp_plan ← old_plan + new_plan
          If own plan not viable
                own_plan ← re-plan(own_plan, opp_plan)
    Decide
          action ← Select next action(own_plan, opp_plan)
    Action
          Recommend action(action)
          wait(quiescence period)
    Housekeeping
Repeat Loop
```

Figure 5 - Command and control assistance algorithm

This algorithm operates on a state space problem description of command and control. It functions as a Disinformation Evaluation Assistant (DEA) that processes environmental input and prints out recommended actions. Its only output is recommendations to the decision maker and the program will halt when a goal state is recognized or user stops the program at line 6 and 7[5].

OODA: Observe

Line 8 through line 12 is the Observation task of the OODA loop. This means identifying the present state so that the next Orientation phase of OODA can assess the progress of the existing plan. Identifying a state is situation assessment with the specific goal of selecting a state in the state space. The cognitive objects are derived from physical objects; the physical environmental signals are collected, filtered for collection errors, a representation built, and transferred through the information network and finally transformed into cognitive objects:

$$o_{physical} \xrightarrow{collection} o_{information} \xrightarrow{abstraction} o_{cognitive}$$

Situation assessment is significant because, in addition to the own teams' actions, the environmental state is being changed by both adversary actions and natural events. The result of the Observation phase will identify the present state, $\{s_{now} \mid s_{now} \in S \wedge s_{now} = \bigcup p_i\}$. Identifying the present state might not be easy because the environment is affected by both teams' actions and uncontrollable environmental factors so the new state can be quite distant from the previous state. The size of the state is another factor that can make situation assessment difficult. There are $|S| = |P_1| \times |P_2| \times \cdots \times |P_m|$ states in a problem space and the number of states might be infinite due to the parameters that were chosen to represent the state, thus an exhaustive sequential matching algorithm is inefficient. A viable situation assessment algorithm can use the knowledge of the previous state to aid in identifying the next state. A simple algorithm for situation assessment is listed below:

```
// Situation Assessment algorithm in state space
state_previous <- state_now
candidate_states <- neighbor(1, state_previous)
     and state_now
for all candidate_states
     if state parameters equal new parameters
          state_now <- state
          return state_now
// The new state is drastically different than
  // previous state.
candidate_states <- states_of(own plan)
for all candidate_states
     Match state to parameters
          return state_now
     else
          add neighbor(1, state) to candidate_states
```

[5] A user button-press event or hardware interrupt will stop the application, or as in many existing systems, the user can always choose to ignore the recommended actions.

```
17   candidate_states <- states_of(opponent's plan)
18   for all candidate_states
19        Match state to parameters
20             return state_now
21        else
22             add neighbor(1, state) to candidate_states
23   Loop
24        if goal state reached
25             Halt
26        Use heuristics to guess a state
27        candidate_states <- neighbor(1, state) and state
28        for all candidate_states
29             match state to parameters
30                  retrun state_now
31   repeat loop
```

Figure 6 - Situation Assessment Algorithm in State Space

The above situation assessment algorithm is necessary when the number of states is large and cannot be enumerated. Ideally all the states of a problem would be finite and stored in a database with its corresponding reaction plan. Each state functions as a key to the pre-formulated game plan for that particular situation, otherwise building a plan in real-time will slow down the OODA process. Moreover, matching a state in a small problem space is easier if using simple linear search. The algorithm would have complexity $O(\frac{n}{2})$ where n is the number of states $|S| = \times_i |P_i|$. In practical command and control situations with sufficiently detailed state description, the number of states is likely to be too large for effective linear search. In that case, then it is reasonable to use a more sophisticated algorithm to identify the state. Figure 6 first attempts to match the neighboring state of the previous state, lines 1 to 8. This is reasonable if the OODA cycle time is short. If no state matches the new parameters, then the algorithm searches through its own plan based on the assumption that the actions that have been emitted could move the environment along the plan. Neighbors of states in the plan are also matched. The opponent's plan, as estimated by the own team, and its near neighbors are searched next. This is based on the assumption that the opposing team has moved the environment to fulfill its plan since the last OODA cycle. Heuristic methods are tried next after searching through the plans. This can include Case Based Reasoning (CBR), Expert System (ES), Bayesian Estimation (BE). Both Case Based Reasoning and Expert System contain trigger facts that are not generalizable, while Bayesian Estimation gives a decimal number in $[0 \cdots 1]$ to a state that indicates the likelihood that the state matches the new environmental parameters. Since the likelihood of occurrence of an event in Bayesian Estimate is based on prior probabilities of other events, this modeling technique would be difficult to implement if the problem space is large.

OODA: Orientation

The Orientation task follows the identification of the present state by the Observation task. The result of Orientation is to update your own plan and also that of the opponent's plan, a theoretical treatment of plan recognition is give by Kautz [Kautz87]. The orientation task is where disinformation processing takes place in the OODA model. A Disinformation Evaluation Assistant (DEA) will keep a plan, ψ_{own}, as its intended course and also keep track of the opponent's plan ψ_{opp}. In the OODA process and the State Space Representation, we give the following definitions as a background for disinformation analysis:

> Definition 2. Disinformation in an adversarial environment includes emitting actions with intent to deceit.
>
> Definition 3. Deception in a planning problem using state space representation means emitting actions reaching states that are circuitous toward the goal state.

The main Orientation process data structure is the three-tuple $<\psi_{own}, \psi_{extern}, <\psi_{opp}^{1}, \psi_{opp}^{2}, \cdots \psi_{opp}^{n} >>$. The first member of the tuple is the own team's actual plan, the second member of the list is the team's projected external plan, and the last item is a history of the opponent's plans, including the latest assessment. This object would have been updated at the end of the Orientation process. The first two items, own plan and external plan, would be the same if disinformation is not needed. On the other hand, a false plan can be generated to disinform the opponent. Below is the Orientation algorithm:

1   `// Completed the observe process and knows the present state`
2   `if state_now is not equal next_state(`ψ_{opp}^{n}`)`
3   ψ_{opp}^{n+1} `<- assess_opponent_plan(`$<\psi_{opp}^{1}, \psi_{opp}^{2}, \cdots \psi_{opp}^{n}>$`)`
4   s_{opp_goal} `<- find_opponent_goal(`$<\psi_{opp}^{1}, \psi_{opp}^{2}, \cdots \psi_{opp}^{n+1}>$`)`
5   `if state_now is not equal next_state(`ψ_{own}`)`
6   ψ_{own} `<- re-plan()`
7   `if want to disinform`
8   ψ_{extern} `<- generate_external_plan(`ψ_{own}`)`
9   `else`
10   ψ_{extern} `<-` ψ_{own}

Figure 7 - Orientation task including disinformation detection and disinformation creation

Figure 7 is the top-level algorithm for the Orientation task, it reassesses both its own and the opponent's plan with an option to disinform. Lines 2 through 4 checks if the existing model of the opponent's plan is correct, and the algorithm reassesses opponent's plan and goal if the model does not correspond to reality. Assessment of the opponent's plan is done first so that the most up-to-date information is available to for its own plan. Lines 5 and 6 examine the progress of the present plan and re-plan when the situation does not match what is expected. That is, the present state is different than the expected state. Line

7 is the choice to disinform. Line 8 will generate a plan for external presentation if disinformation is deemed appropriate.

The interesting parts of the algorithm are in the details of `assess_opponent_plan(), find_opponent_goal(), generate_external_plan().` These procedures operate on the SSR (state space representation) of the command and control domain and will use established graph algorithms such as the definition of a neighborhood [Cormen02]. These functions can also be fulfilled with the choice of appropriate heuristic methods. The `assess_opponent_plan()` procedure assumes that the opponent is operating under a plan to reach its objective. This procedure called upon during each OODA cycle to produce or re-affirm the opponent's plan. If the opponent is sending disinformative actions then it is likely that its plan will be inconsistent.

> Axiom 8: In adversarial interaction with a disinformative opponent, the perceived plan will change.

To identify the opponent's plan, it is easier if there is a good assessment of its goal state, then all paths from the present state to that goal state are possible candidates for their plan, $\{\psi = < s_1, \cdots, s_n > | \, s_1 \equiv s_{now} \wedge s_n \in S_{goal}\}$. The `assess_opponent_plan()` will select the most likely plan, save that plan, and inform the user of this new development. In the case the opponent's goal state is not known, then `find_opponent_goal()` will find that goal. As the course of actions unroll and historic plans are being accumulated, a likely goal can be identified from the set of last states of the opponent's historical plans; this is based on the assumption that the opponent's seemingly capricious actions are derived from its hidden real and stable plan. On the other hand, in the case disinformation is indicated, `generate_external_plan()` can generate a false plan. A starting candidate set of possible false plans are: $\{\psi = < s_1, \cdots, s_n > | \, s_1 \equiv s_{now} \wedge s_n \in S_{goal}\}$. To deter easy identification of its own goal, the ending state should be selected from the neighbors of the goal state. Modeling in this algorithm is limited to the opponent's plan. The model also includes detection for disinformation as in Figure 6.

The final two tasks of OODA in SSR are Decide and Act. At this point, the algorithm has a viable plan and it will search through actions that are available at the present state to reach the next state on the plan and then emit that action.

Modeling and Bayesian Estimation

Modeling in this algorithm is limited to the estimation of the opponent's plan, and that model is used to detect disinformation, as described in Figure 4. Many methodologies have been used to increase the effectiveness of the cognitive computing effort. Notably, Bayesian Estimation, Expert System, Case-Based Reasoning, and Neural Network. Disinformation has been operationally described as projecting an external plan that is different than the internal plan. When the actions induced by the sham external plan have been received by the opponent, the opponent's situation assessment process will perceive a plan that plan and recognize that plan at its face value. In is only through the evaluation of historic record of past plans that possibility of disinformation is discovered. If there is a deviation from expectation (i.e. present state is not the same as expected state as according to the opponent's plan) then `assess_opponent_plan()`, and `find_opponent_goal()`procedures are activated to assess the opponent's latest plan, and apparently only consume the time overhead needed to run the two procedures, thus

> Axiom 9: Disinformation can be detected and will cause opponent model to be rebuilt in unremarkable time.

This conclusion does not indicate a sudden large change similar to chemistry's Titration Effect or business' Tilting Point Effect. Instead, the OODA loop process will just take expected longer time to complete. However, drastic change would be likely if the original OODA cycle time is slow and assessment of disinformation is not done at every OODA cycle to save time.

The utility of Bayesian estimation is to give a real number in $[0 \cdots 1]$ that indicate the probably of a particular state [Gottinger75]. The estimate takes a state as input and returns a real number in the OODA algorithm, Bayesian estimate can be used in:

> Lines 9 of Figure 5 to identify the present state
> Line 15 of Figure 5 to identify the opponent's plan [Albrecht98]
> Line 26 of Figure 6 to identify the present state
> Line 3 of Figure 7 to identify the opponent's plan

If the number of possible states is reasonable then theoretically it is possible to calculate the probability of all the states and return the state with the highest probably, but it is more likely that state space is large and only a selected significant states will be monitored and probability assigned to the states, and that would take a bounded time and not be a source of uncertainty.

Conclusion

Disinformation is a process that spans operational scope and battle domains. We investigate the nature of disinformation by applying it in an abstract and flexible representation called State Space Representation (SSR). SSR is rich enough to represent physical, information, and cognitive military domain objects. This representation is a solid grounding for the diverse knowledge of command and control research. In addition, ample tools are available[6] for both theoretical and engineering tasks to solve state space problems.

This paper demonstrates the utilization of the State Space Representation (SSR) with an algorithmic implementation of the OODA loop and describes the conditions of disinformation in the algorithm. We conclude that disinformation will not result in

[6] The state space paradigm is used in many fields and mathematicians and computer scientists have developed many tools that operate on state space representation.

sudden unexpected state changes if the situation assessment task is carried out every OODA cycle.

One of the utility of using State Space Representation is the possibility to create a Disinformation Evaluation Assistant (DEA) to help provide command and control capabilities closer to the tactical levels to anticipate the future of battle trend.

Reference

[Alberts01] Alberts, D. S.; Garstka, J. J.; Hayes, R. E.; Signori, D. A. Understanding Information Age Warfare, CCRP Publication Series, www.dodccrp.org

[Albrecht98] D. W. Albrecht, I. Zukerman, and A. E. Nicholson, “Bayesian Models for Keyhole Plan Recognition in an Adventure Game,” User Modeling and User-Adapted Interaction, vol. 8, no. 1-2, 1998, pp. 5-47.

[Cohen87] P. R. Cohen, H. J. Levesque, Intention = Choice + Commitment. In Proceedings of the Sixth National Conference on Artificial Intelligence, Seattle, Washington, 410-415, 1987.

[Cormen02] Cormen, Thomas H., Leiserson, Charles E. et. al., “Introduction to Algorithms Second Edition”, the MIT Press, 2002, page 669.

[Endsley95] Endsley, Mica R., “Toward a Theory of Situation Awareness in Dynamic Systems”, Human Factor 1995, 37(1), 32-64.

[Gottinger75] Gottinger, Hans Werner, “Bayesian Analysis Probability and Decision”, Vandenhoeck & Ruprecht, Göttingen.

[Kautz87] Kautz, H. A. A formal theory of plan recognition, Technical Report, University of Rochester, Rochester, New York.

[Koohang03] Koohang, A.; Weiss, E. Misinformation: toward Creating a Prevention Framework, *Information Science*, InformationScience.org, June 2003.

[Shaw85] Shaw, Robert L., “Fighter Combat, Tactics and Maneuvering”, 1985, Naval Institute Press, Annapolis, Maryland.

[Winston93] Winston, Patrick Henry, “Artificial Intelligence Third Edition”, Addison-Wesley, 1993.