Modeling the Information Age Combat Model: An Agent-Based Simulation of Network Centric Operations

Sean Deller, Old Dominion University
Ghaith A. Rabadi, Old Dominion University
Michael I. Bell, Alidade Incorporated
Shannon R. Bowling, Old Dominion University
Andreas Tolk, Old Dominion University

Abstract. The Information Age Combat Model (IACM) was introduced by Cares in 2005 to contribute to the development of an understanding of the influence of connectivity on force effectiveness that can eventually lead to quantitative prediction and guidelines for design and employment. The structure of the IACM makes it clear that the Perron-Frobenius Eigenvalue is a quantifiable metric with which to measure the organization of a networked force. The results of recent experiments presented in Deller, et al., (2009) indicate that the value of the Perron-Frobenius Eigenvalue is a significant measurement of the performance of an Information Age combat force. This was accomplished through the innovative use of an agent-based simulation to model the IACM and represents an initial contribution towards a new generation of combat models that are net-centric instead of using the current platform-centric approach. This paper describes the intent, challenges, design, and initial results of this agent-based simulation model.

Introduction

The Information Age Combat Model (IACM), recently introduced by Cares (2005), attempts to describe combat (or competition) between distributed, networked forces or organizations. The basic objects of this model are not platforms or other entities capable of independent action, but rather nodes that can perform elementary tasks (sense, decide, or influence) and links that connect these nodes. Information flow between the nodes is generally necessary for any useful activity to occur.

Once the IACM has been defined in terms of a network of nodes and links, the language and tools of graph theory (see, for example, Chartrand 1984) can be used for both description and analysis. A concise description of any graph is provided by the adjacency matrix *A*, in which the row and column indices represent the nodes, and the matrix elements are either one or zero according to the rule: $A_{ij} = 1$, if there exists a link from node *i* to node *j* and $A_{ij} = 0$, otherwise. One method used in studying the evolution of complex adaptive systems (chemical, biological, social, and economic) is calculation of the principal (maximum) eigenvalue of the adjacency matrix (Jain and Krishna, 1998). The existence of a real, positive principal eigenvalue of A_{ij} is guaranteed by the Perron-Frobenius theorem. This eigenvalue, λ_{PFE}, represents the ability of a network to produce feedback effects in general and combat power specifically in the case of the IACM.

The structure of the IACM makes it clear that the Perron-Frobenius Eigenvalue (λ_{PFE}) is a quantifiable metric with which to measure the organization of a networked force. The results of recent experiments presented in Deller, et al., (2009) indicate that the value of the λ_{PFE} is a significant measurement of the performance of an Information Age combat force. This paper describes the intent, challenges, design, and initial results of this agent-based simulation model.

The Information Age Combat Model (IACM)

The IACM employs four types of nodes defined by the following properties:

- *Sensors* receive signals about observable phenomena from other nodes and send them to Deciders;
- *Deciders* receive information from Sensors and make decisions about the present and future arrangements of other nodes;
- *Influencers* receive directions from Deciders and interact with other nodes to affect the state of those nodes;
- *Targets* are nodes that have military value but are not Sensors, Deciders, or Influencers.

Each node belongs to a "side" in the competition, of which there are at least two. We will restrict the present discussion to two sides, conventionally termed BLUE (depicted in black in the figures) and RED (depicted in gray). In principle, any pair of nodes can interact, regardless of side, but some restrictions will be found to occur for both theoretical and practical reasons. It is worth noting that Influencers can act on any type of node, and Sensors can detect any type. The Target type was introduced primarily to reflect the fact that not all military assets fall into one of the other three types. In most situations, however, an Influencer will target an adversary Sensor, Decider, or Influencer. The figures in this paper utilize the basic elements of graph theory.

The basic combat network shown in Figure 1 represents the simplest situation in which one side can influence another. The BLUE Sensor (S) detects the RED Target (T) and informs the BLUE Decider (D) of the contact. The Decider then instructs the BLUE influencer (I) to engage the Target. The Influencer initiates effects, such as exerting physical force, psychological or social influence, or other forms of influence on the target. The process may be repeated until the Decider determines that the desired effect has been achieved. It should be noted that the effect assessment requires sensing, which means that this will be conducted in a new circle. This most basic combat network is also referred to as a *combat cycle*.

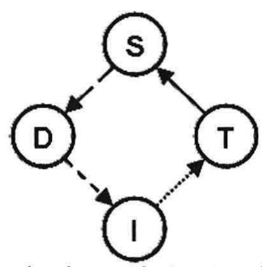

Figure 1. The basic combat network represents the simplest situation in which one side can influence another.

Each of the four links in Figure 1 is shown with a different type of line in order to emphasize the fact that the flows across these links may be very different. In particular, some links may represent purely physical interactions, while others may entail both physical processes and information flows. Two opposing combat cycles comprise the simplest two-sided combat network.

Cares (2005) described the simplest complete (two-sided) combat network as having 36 possible links. While the number of possible links for eight nodes (four each for BLUE and RED) is 64, we were able to exclude 28 and reduce that number to 36 based on the following important assumptions. The results are shown in Figure 2.

- Targets are passive; their only role is to be sensed and influenced. Therefore, 12 links from Targets to any nodes other than a Sensor were excluded.
- Sensors take no action; they provide information to Deciders and Sensors. Therefore, 10 links from Sensors to any nodes other than a Sensor or own Decider were excluded.
- Deciders act only through Influencers but can be sensed. Therefore, 6 links from Deciders to any adversary nodes except a Sensor were excluded.

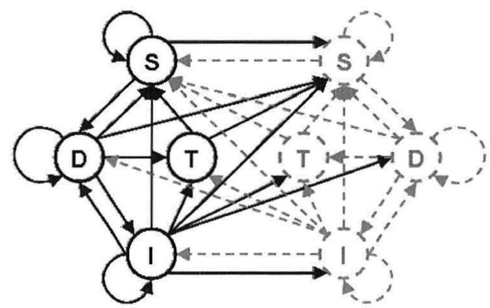

Figure 2. The simplest complete combat network represents all the ways in which Sensors, Deciders, Influencers and Targets interact meaningfully with each other.

When the BLUE/RED symmetry is taken into account, the number of link types is reduced to 18. These are listed in Table 1, where the nodes are identified as in Figure 2. Links between a node and itself in Figure 2 have been interpreted as connecting two different nodes of the same type and side.

Table 1. Types of links available in the IACM.

Link Type	From	To	Interpretation
1	S_{BLUE} S_{RED}	S_{BLUE} S_{RED}	S detecting own S, or S coordinating with own S
2	S_{BLUE} S_{RED}	D_{BLUE} D_{RED}	S reporting to own D
3	S_{BLUE} S_{RED}	S_{RED} S_{BLUE}	S detecting adversary S
4	D_{BLUE} D_{RED}	S_{BLUE} S_{RED}	S detecting own D, or D commanding own S
5	D_{BLUE} D_{RED}	D_{BLUE} D_{RED}	D commanding own D
6	D_{BLUE} D_{RED}	I_{BLUE} I_{RED}	D commanding own I
7	D_{BLUE} D_{RED}	T_{BLUE} T_{RED}	D commanding own T
8	D_{BLUE}	S_{RED}	S detecting adversary

	D_{RED}	S_{BLUE}	D
9	I_{BLUE} I_{RED}	S_{BLUE} S_{RED}	I attacking own S, or S detecting own I
10	I_{BLUE} I_{RED}	D_{BLUE} D_{RED}	I attacking own D, or I reporting to own D
11	I_{BLUE} I_{RED}	I_{BLUE} I_{RED}	I attacking own I, or I coordinating with own I
12	I_{BLUE} I_{RED}	T_{BLUE} T_{RED}	I attacking own T
13	I_{BLUE} I_{RED}	S_{RED} S_{BLUE}	I attacking adversary S, or S detecting adversary I
14	I_{BLUE} I_{RED}	D_{RED} D_{BLUE}	I attacking adversary D
15	I_{BLUE} I_{RED}	I_{RED} I_{BLUE}	I attacking adversary I
16	I_{BLUE} I_{RED}	T_{RED} T_{BLUE}	I attacking adversary T
17	T_{BLUE} T_{RED}	S_{BLUE} S_{RED}	S detecting own T
18	T_{BLUE} T_{RED}	S_{RED} S_{BLUE}	S detecting adversary T

The interpretation of some of the links (types 1, 4, 9, 10, 11, and 13 in Table 1) is ambiguous, and was recognized in the initial development of the IACM (Cares 2005). Overcoming this ambiguity was a necessary step in developing a simulation that would enable an analysis of the value of the λ_{PFE} as a significant measurement of the performance of an Information Age combat force. The simulations presented here are a step in this direction, since they employ only basic combat networks similar to Figure 1, but with the Target replaced by an adversary Sensor or Influencer. These combat cycles (Cares 2005) contain only links of types 2, 3, 6, 13, and 15. Of these, only type 13 is ambiguous.

A Basic Agent-Based Model Using the IACM

The structure of the IACM makes it clear that the λ_{PFE} is a quantifiable metric with which to measure the organization of a networked force, but is it an indicator of combat effectiveness? To determine this we constructed an agent-based simulation representation of the IACM and conducted a series of force-on-force engagements using opposing forces of equal assets and capabilities, but differing in their connectivity arrangements or configurations. These differences in connectivity often, but not necessarily, lead to unequal λ_{PFE} values.

The agent-based paradigm was utilized for this purpose because the resulting models provide the ability to account for small unit organization, maneuver, and the networked effects that are the focus of our investigation. An additional advantage of utilizing an agent-based simulation was the ability to work around the ambiguities of link interpretation in the IACM. For example, instead of a mutually exclusive choice between defining a directional link from a BLUE Influencer to a RED Sensor (type 13 in Table 1) as either the Influencer "targeting" the Sensor or as the Sensor "sensing" the Influencer, both abilities can be represented in the agent-based simulation.

The first challenge in modeling the IACM concerned the adjacency matrix representation of the network. The IACM as originally described by Cares (2005) uses a single adjacency matrix to reflect the collective organization of both BLUE and RED forces. In this approach, the λ_{PFE} value is dependent on the configurations of both the BLUE and RED forces and might well represent the extent to which feedback effects occur in the engagement. Obviously, BLUE and RED each seek separately to maximize their own networked effects while minimizing those of the opposing force. This cannot be represented by a single λ_{PFE} value, so we calculate separate values (λ_{BLUE} and λ_{RED}) to reflect the potential networked effects of the configurations of each of the opposing forces. These calculations required the adjacency matrices include a single Target node representative of all the enemy forces capable of being targeted. In other words, the values of λ_{BLUE} and λ_{RED} are determined solely by the arrangement of their respective assets, independent of the asset arrangement of the opposing force.

In order to best associate any difference in force effectiveness to the difference in connectivity, the opposing forces consisted of the same number of Sensors, Deciders, and Influencers, differing only in the manner in which they were arranged (i.e., linked). Since the potential value of a Sensor may not equal the potential value of an Influencer, the composition of each configuration considered in this work contained an equal number Sensors and Influencer to preclude any bias towards those configurations that have more of one or the other. Additionally, both types of nodes had identical performance capabilities (i.e., the sensing range was chosen equal to the influencing range, and the speeds of movement of the two types of node were equal).

In order to gain a "first order" understanding of the IACM, we made two key scoping decisions. First, each Sensor and Influencer would only be connected to one Decider (but any given Decider could be connected to multiple Sensors and Influencers). Second, the connectivity within any force was limited to only those "vertical" links necessary to create combat cycles (i.e., link types 2, 3, 6, 13, and 15 in Table 1), which are the essence of the λ_{PFE} (the most basic element of the IACM).

Developing the NetLogo Model

The agent-based simulation environment utilized for this research was NetLogo (Wilenski 1999). The code of the agent-based model closely follows the logic of the IACM, with a few notable exceptions. Agents served as Sensors, Deciders, and Influencers, but Targets were not included as they served no purpose other than to absorb losses. Given that this work represents a "first cut" effort, including Target agents with no detect, direct, or influence capabilities would only serve to clutter the results.

Additionally, Deciders cannot be destroyed in the present model. This was done in recognition of their unique role in connecting multiple Sensors and Influencers. Destruction of a Decider typically renders a number of other nodes useless (effectively destroyed), making it a particularly high value target. Since targets are detected and engaged in random order in our model, we wished to give all targets equal value in order not to generate atypical engagements that might bias the results.

The agent rules sets, themselves, function in accordance with the IACM. Sensors detect enemy nodes within the sensing range parameter, and communicate that information to their assigned (connected) Deciders. Deciders communicate the sensing information to their assigned Influencers. Influencers destroy the nearest enemy node that is both "sensed" by a Sensor connected to that Influencer's Decider, and within the influencing range parameter. Deciders direct Sensor movement towards areas of suspected enemy nodes. Deciders direct Influencers to move towards the nearest "sensed" enemy node. All nodes are assumed to perform their functions perfectly and instantaneously. Agent interactions are deterministic, i.e., the probabilities of detect, communicate and kill are all "1". A stochastic dimension to the model can be built once a better understanding of the research questions is gained, and this new dimension can be used to model errors and delays representing technological and human performance factors. Most importantly, the rules sets and parameter values for both BLUE and RED agents were identical.

Each agent in the model is defined as a part of an agentset (i.e., "`breed`") associated with a particular Decider. Since the nodes of the IACM are generic, the most important defining characteristic of any agent is its connectivity. For example, all BLUE Sensors and Influencers connected to the BLUE Decider_1 are established by the following breeds:

```
breed [ BInfluencer1s BInfluencer1 ]
breed [ BSensor1s BSensor1 ]
```

The actual numbers of agents within these breeds will vary according to the configuration being tested. Sliders were utilized for this purpose, thereby enabling the BehaviorSpace feature to vary the configurations automatically and allowing us to execute the large number of engagements necessary to complete this research. The BLUE Decider_1 itself is also defined as a breed, but consists only of just that single agent. Similar agents for all other BLUE and RED Sensors, Influencers and Deciders were established.

The connectivity between these breeds represents the combat cycle links of the IACM (specifically link types 2, 3, 6, 13 and 15 as explained in Table 1). Link types 2 ("detection"), 6 ("order"), 13 ("LOF")[1] and 15 ("LOF") are defined in the simulation by the `directed-link-breed` keyword.

As mentioned earlier in this paper, link type 13 has an ambiguous meaning in the IACM. The `directed-link-breed` keyword defines the Influencer-to-Sensor link as the Influencer attacking an enemy Sensor. Both link type 3 and the other IACM interpretation of link type 13 (i.e., a Sensor detecting an adversary Influencer) will be defined by the sense procedure later in the code. Finally, all agents within each `breed` have certain variables that are tracked during the simulation, such as `side` (i.e., BLUE or RED), `dead` (i.e., agents that are attacked by an opposing Influencer may no longer act), and `sensed` (i.e., at any given `tick` count within the simulation an agent may be within sensing range of one or more opposing Sensors).

Given the large number of engagements within this experiment, it was imperative to utilize the BehaviorSpace feature of NetLogo. To enable this, each of the different force configurations were defined by using the `set` command to establish the appropriate numbers of Sensors and Influencers for each of the BLUE and RED Deciders. For example, BLUE Configuration (i.e., "BID") #0 assigned 5 Sensors and 5 Influencers to

[1] LOF is an acronym for "line of fire," which is a direct horizontal line from a firing weapon to its target.

BLUE Decider$_1$, and one of each to the other 4 Deciders:

```
if BID = 0 [set Bconfig [5 1 1 1 1 5 1 1 1 1]]
if BID = 2 [set Bconfig [5 1 1 1 1 4 2 1 1 1]]
  set number-BSensor1s item 0 Bconfig
  set number-BSensor2s item 1 Bconfig
  set number-BSensor3s item 2 Bconfig
  set number-BSensor4s item 3 Bconfig
  set number-BSensor5s item 4 Bconfig
  set number-BInfluencer1s item 5 Bconfig
  set number-BInfluencer2s item 6 Bconfig
  set number-BInfluencer3s item 7 Bconfig
  set number-BInfluencer4s item 8 Bconfig
  set number-BInfluencer5s item 9 Bconfig
```

BLUE Configuration #2 is nearly identical, differing only in one link. Decider$_1$ now only has 4 assigned Influencers while Decider$_2$ now has 2. The movement of a single link is not trivial as it may have a significant impact on both the λ_{PFE} value and the average probability of Win for that particular configuration. All different force configurations were established in this manner, thereby allowing the BehaviorSpace feature to automatically cycle through all possible engagements between the BLUE and RED configurations instead of running the simulation one engagement at a time.

Since the focus of this effort is to gain insight into the relationship between the λ_{PFE} value and the effectiveness of a networked force, the agent-based simulation rules of engagement were quite simple. The battlespace (i.e., "world") within the model is deliberately featureless in order to focus on the configurations themselves. The agents are randomly distributed across the battlespace at the beginning of each engagement. Engagements continued until either all of the Sensors and Influencers of one force were annihilated, or both forces were incapable of continued combat (i.e., neither side contained a functioning combat cycle). A single run of the agent-based model will result in a BLUE win, a RED win, or an undecided result.

During each time `tick` of the simulation, the following procedures are executed: `establish-links`, `sense`, `track`, `shoot`, `kill`, `move-Influencer`, `move-Sensor`, and `reset`. The `establish-links` procedure establishes the links defined by the `directed-link-breed` keyword earlier in the code. It does so by `breed`, thereby ensuring each Sensor and Influencer is connected to only one Decider.

```
to establish-links
  ask BDecider1s [
      ask    BSensor1s    [create-detection-to
myself [set color blue] ]
      ask   BInfluencer1s    [create-order-from
myself [set color blue] ] ]
```

At this time, two of the four necessary links (types 2 and 6) of the IACM combat cycle have been established in the simulation. Link type 3 and one of the two interpretations of link type 13 are established in the `sense` procedure. In this procedure, every Decider asks its assigned Sensors (i.e., "`in-link-neighbors`") to identify all adversary Sensors and Influencers within its sensing range (i.e., "`s-range`"). Upon identification, the specific sensed variable of the targeted agent for that particular opposing Decider is set to a value of "1." The `s-range` parameter remains constant for all Sensors, either BLUE or RED, over time. The `sense` procedure depicted below includes all opposing `breeds` (only `RInfluencer1s` is shown in this example) and is repeated for every BLUE and RED Decider `breed`:

```
to sense
  ask BDecider1s [
    ask in-link-neighbors [
      ask RInfluencer1s in-radius s-range [set
   sensedBD1 1]
```

The remaining links necessary to complete the IACM combat cycles (link type 15 and the alternate interpretation of link type 13) are established by the `track`, `shoot`, and `kill` procedures. During the `track` procedure, every Decider asks its assigned Influencers (i.e., "`out-link-neighbors`") to identify all adversary Sensors and Influencers within its influencing range (i.e., "`i-range`"). Upon identification, the targeted agent is linked to that particular Influencer using the `create-LOF-from-myself` keyword. The `i-range` parameter remains constant for all Influencers, either BLUE or RED, over time. The `track` procedure depicted below includes all opposing `breeds` (only `RInfluencer1s` is shown in this example) and is repeated for every BLUE and RED Decider `breed`:

```
to track
  ask BDecider1s [
    ask out-link-neighbors [
      ask    RInfluencer1s    in-radius    i-range
[create-LOF-from myself]
```

Now that the complete IACM combat cycle has been established, the `shoot` and `kill` procedures represent its execution. During this procedure, each Decider directs its assigned Influencers to identify the single closest opposing Sensor or Influencer with which it shares a LOF link. This limits all Influencers to the same rate of fire of one targeted node per time `tick`. Identification is portrayed by setting the `dead` variable equal to "1."

```
to shoot
  ask BDecider1s [
    ask out-link-neighbors [
      ask out-link-neighbors [
        let   $targets-sensed   turtles   with
[(sensedBD1 = 1) and (side = 2)]
          if any? $targets-sensed [
```

```
        ask    min-one-of    $targets-sensed
[distance myself] [set dead 1] ] ] ] ]
```

Following this identification, the `kill` procedure deletes all agents that have been "sensed," "tracked" and "shot." The purpose of separating the `kill` procedure from the `shoot` procedure is to allow simultaneous shots, thereby precluding any advantage that would be gained by the order of execution of the `shoot` procedure code.

```
to kill
  ask turtles with [(dead = 1)] [die]
end
```

The collective effect of the `sense`, `track`, `shoot`, and `kill` procedures is to require that a Sensor and an Influencer must be assigned to the same Decider and within their respective `s-range` and `i-range` in order to successfully complete a combat cycle (i.e., eliminate the targeted node).

Upon completion of all combat cycle execution, all remaining Sensors and Influencers are moved. The `move-Influencer` procedure directs all Influencers to move towards the nearest opposing Sensor or Influencer that has been sensed by a friendly Sensor assigned to the same Decider. If there are no qualifying opposing Sensors or influencers, then the Influencer will not move. Each time `tick` includes five iterative moves of a distance of "1" that are sequential between Deciders and sides in order to preclude any advantage of moving first or last. An example iteration for one Decider follows below:

```
to move-Influencer
  ask BDeciderls [
    ask out-link-neighbors [
      let    $targets-sensed    turtles    with
[(sensedBD1 = 1) and (side = 2)]
        if any? $targets-sensed [
          set    heading    towards    min-one-of
$targets-sensed [distance myself] forward 1 ]
] ]
```

The `move-Sensor` procedure directs all Sensors to move towards the nearest Sensor or Influencer that is not currently sensed by a friendly Sensor assigned to the same Decider. This procedure is necessary to enable both sides to eventually target those opposing Sensors and Influencers that did not start the simulation within any friendly Sensor's `s-range`.

```
to move-Sensor
  ask BDeciderls [
      ask in-link-neighbors [
        let    $targets-sensed    turtles    with
[(sensedBD1 =
   1) and (side = 2)]
          if not any? $targets-sensed [
            let $targets-unsensed turtles with
                [(sensedBD1 = 0) and (side =
2)]
              if any? $targets-unsensed [
                let $nearest-unsensed min-one-
of $targets-unsensed [distance myself]
                  set heading towards $nearest-
unsensed forward 1 ] ] ] ]
```

The final procedure during each time `tick` is `reset`. During this procedure, all sensed variables are reset to "0" and all links, to include the LOF "tracking" links, are deleted in preparation for the `establish-links`, `sense`, `track`, `shoot`, `kill`, `move-Influencer`, `move-Sensor`, and `reset` procedures for the next time `tick`.

Initial Results

The initial experiment consisted of all possible engagements of the 42 different configurations of two networked forces (BLUE and RED), each containing 7 Sensors, 3 Deciders, 7 Influencers, and 1 Target. The sole Target node is representative of all the opposing nodes vulnerable to destruction. Additionally, the capabilities for each of these node types were identical between the forces. A comprehensive test of each of these 42 configurations against each other required 1,764 different engagements. Each engagement was represented by 30 replications, each with a random distribution of the BLUE and RED nodes across the battlespace.

The results clearly show that the probability of a BLUE win increases for those BLUE configurations with a greater λ_{PFE} value. A simple linear regression confirmed this with a coefficient of determination (R^2) equal to 0.896 for the following equation:

$$y = 1.0162(x) - 1.5780$$

where, y = the average probability of a BLUE win for that configuration

x = the λ_{PFE} value of a configuration

Summary & Conclusion

The agent-based simulation described in this paper was employed to analyze the value of the λ_{PFE} as a quantifiable metric with which to measure the organization of a networked force. This simulation was specifically designed to overcome the challenges of the IACM link ambiguity and the large number of engagements necessary to complete the research. The results of recent experiments presented in Deller, et al., (2009) indicate that the value of the Perron-Frobenius Eigenvalue is a significant measurement of the performance of an Information Age combat force.

References

Deller, Sean, Bell, Michael I., Rabadi, Ghaith A., Bowling, Shannon R., and Tolk, Andreas (2009). Applying the Information Age Combat Model:

Quantitative Analysis of Network Centric Operations. Accepted for publication in *The International C2 Journal*.

Cares, Jeff. 2005. *Distributed Networked Operations*. iUniverse, New York.

Chartrand, Gary. *Introductory Graph Theory*. Dover Publications, 1984.

Jain, Sanjay, and Krishna, Sandeep. 1998. Autocatalytic Sets and the Growth of Complexity in an Evolutionary Model. *Physical Review Letters* 81: 5684-5687.

Jain, Sanjay, and Krishna, Sandeep. 1999. Emergence and growth of complex networks in adaptive systems. *Computer Physics Communications* 121-122: 116-121.

Wilenski, Uri. 1999. *NetLogo*. Center for Connected Learning and Computer-based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/NetLogo.