

# Simulating Agility for Networked Military Systems

Dr. Anthony H. Dekker

*Defence Science and Technology Organisation, Canberra, Australia  
dekker@ACM.org*

**Abstract.** This paper describes some extensions to CAVALIER, a tool for studying the performance of networked military organisations. Specifically, we have improved event handling and added neural-network-based learning techniques. To demonstrate the use of this simulation tool, the paper describes in detail an experiment studying networking and agility. Agility refers to the ability of an organisation, person, or military force to perceive an upcoming threat, and to respond quickly enough to it. In the experiment reported here, a networked force is required to conduct a simple abstraction of a military mission: picking up 100 “items” scattered around a micro-world, while at the same time dealing with hostile forces. The experiment illustrates some key principles of networked operation: First, agents require either early awareness of upcoming threats, or the ability to respond to them very rapidly. These two attributes can be traded off against each other. Second, there is little benefit in networking if agents already have enough information, or if they do not have any information worth sharing. Third, high-quality information creates a situation where motion causes risk; but if this breeds risk averseness, overall mission success may suffer.

## 1. INTRODUCTION

The US Marine Corps’ **Project Albert** [1],[2] introduced improved techniques for Defence analysis, to “collaboratively explore the vast space of possibilities inherent in the questions that our decision makers face in today’s uncertain world.” Project Albert achieves this goal by adopting analytical techniques which [3]:

- Are open rather than closed, and peer-reviewed rather than bureaucratically reviewed;
- Have a data-rich subject-matter orientation, rather than a mechanical model orientation;
- Illuminate, rather than suppress, risk and uncertainty;
- Are future-oriented rather than Cold-War-oriented.

These techniques include **data farming** [4],[18] for exploring the behaviour of parameterised models. In previous work, we have extended data farming to **network farming** [5],[6] for exploring network-based systems. Network farming allows models to be parameterised on particular network topologies as well as on numerical parameters, and analyses the behaviour of the models using various network metrics. We perform network farming using the CAVALIER tool, which integrates agent-based simulation, calculation of network metrics, statistical analysis, and visualisation of results.

An important Project Albert technique is the use of regular international workshops, which explore complex systems over a period of several days, in collaboration with model builders and analysts. The important issue in such workshops is to raise questions, rather than provide definitive answers, and to give participants increased understanding of the systems under study.

Network farming allows questions about network topology and connectivity to be raised and explored.

In this paper, we describe two improvements to the agent-based simulation system embedded within our network farming tool [7]. These improvements are the use of **event queues** for more efficient event management, and **neural-network learning** to reduce the amount of fine-tuning of parameters that is required to obtain realistic behaviour.

We also describe a simulation experiment studying the performance of a networked military organisation, which illustrates some key networking and agility principles (listed in Section 7).

**Agility** refers to the ability of an organisation, person, or military force to perceive an upcoming threat, and to respond quickly enough to it. This paper focuses on agility at the tactical military level, where the threat is of being shot at. However, agility also applies at higher levels, where the threat may require organisational restructuring, cultural adjustment, purchases of technology, or strategic rethinking. Agility requires perceiving the key issue early enough, and then responding rapidly enough to it. To some extent, ability to perceive and ability to respond quickly can be traded off against each other, as our experimental results will show. A more detailed discussion is provided in [17].

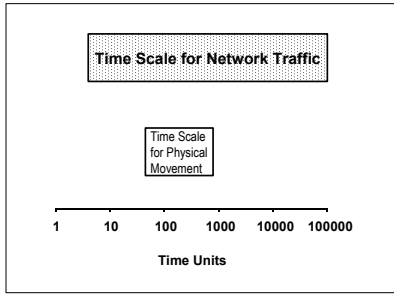
## 2. IMPROVED SIMULATION: EVENT QUEUES

The agent-based simulation systems used by Project Albert (such as MANA, ISAAC, etc.) typically utilise discrete time-steps in a grid-based world. In studying networked systems, we wish to study network transmission speeds which may be either very fast (if electronic), or very slow (if bureaucratic), compared to the physical movement of agents in the world. Figure 1 illustrates the range of timescales in our experiments. In real life, network transmission times range from microseconds to months, and physical movement times from seconds to hours.

In order to allow for such variation in timescales, we require an **event queue** [8], rather than time-steps, to schedule events. However, because events may be scheduled very far ahead, for efficiency of insertion we use an event queue which is an array of priority queues  $Q_1, \dots, Q_k$  (currently implemented as linked lists). Events at time  $t$  are stored in queue  $Q_i$  where:

$$i = 1 + \frac{kt}{1 + t_{\max}}$$

and  $t_{\max}$  is the maximum possible time. For the experiment reported here,  $k = 1000$  and  $t_{\max} = 100,000$ .



**Figure 1:** Time Scales for Network Traffic and Physical Movement

### 3. IMPROVED SIMULATION: LEARNING

Agent-based simulations used in Project Albert tend to have a large number of numerical parameters controlling agent behaviour. Fine-tuning these parameters is time-consuming and a distraction from the main goal of workshops. Accordingly, the simulation reported here incorporates a neural-network learning mechanism for adjusting such parameters, based on that of Dekker and Piggott [9]. As in our previous work [10], a range of different agent behaviours is provided, including:

- Enemy avoidance
- Moving towards enemies
- Random movement
- Moving to squares not previously visited
- Moving towards friends
- Separation from friends

Each behaviour  $B_i$  calculates a velocity vector  $V_i$ , and the velocity of an agent is regularly recomputed to be the vector sum:

$$\sum_i w_i V_i$$

where  $w_i$  is a weight (other kinds of behaviour, not restricted to movement, are also possible). This approach was inspired by the “boids” implementation of Reynolds [11].

However, in the extension described here, when good events occur (such as achieving a goal), the weights  $w_i$  of recently used behaviours are increased, and when bad events occur (such as being shot at), the weights  $w_i$  of recently used behaviours are decreased.

More specifically, each behaviour  $B_i$  also calculates a confidence level  $c_i$ , expressing the appropriateness of the velocity vector  $V_i$ . In addition, an *a priori* weight  $a_i$  is maintained for each behaviour, providing an overall indication of how useful that behaviour is. The actual weight is calculated as:

$$w_i = \frac{a_i c_i}{f}$$

where  $f$  is a scaling factor to ensure that the final vector sum does not exceed the maximum possible velocity.

For each behaviour  $B_i$ , we also maintain an activation history score  $h_i$ , which slowly decays towards zero. When behaviour  $B_i$  is used, the corresponding history score is reset to:

$$h_i = \frac{a_i c_i}{f}$$

Consequently,  $h_i$  is large for behaviours recently given a large weight, and small for behaviours not recently used, or recently given a low weight.

When significant events occur, the *a priori* weight  $a_i$  is adjusted for behaviours with large  $h_i$ . This is done by replacing  $a_i$  by  $(1 - \alpha h_i) a_i + \alpha h_i \beta$ , where  $\alpha$  is a parameter controlling the speed of learning (typically 0.5), and  $\beta$  is a number reflecting the quality of the events (high for good events, and low for bad events). To avoid interactions with later events, the activation histories  $h_i$  are reset to zero after each significant event.

This learning process has the desired effect of rewarding useful behaviours (which lead to good events), and penalising inappropriate behaviours (which lead to bad events). In addition, it makes the simulation more realistic, since humans also learn [19].

### 4. EXPERIMENTAL DESIGN

The purpose of military forces is to carry out various activities such as humanitarian relief, peacekeeping, evacuation, and, of course, war. Such activities are conducted under dangerous situations, which may involve environmental and/or military threats.

The experiment reported here uses a simple abstraction of such activities. A simulated networked friendly “Blue” force of 12 agents has the mission of locating and picking up 100 “items” on a 50×50 discrete grid. A snapshot of one experimental run is given in Figure 2, with the “items” shown as crosses.

While carrying out their mission, the “Blue” agents are opposed by a hostile force of 20 non-networked (but otherwise identical) “Red” agents. The “Red” agents engage in combat with the “Blue” agents, in an effort to prevent the “Blue” agents from picking up the “items.”

Each agent is equipped with a sensor and a weapon. The Blue networked agents broadcast their sensor information across network links to every reachable agent. Table 1 describes the values of various simulation parameters used in the experiment.

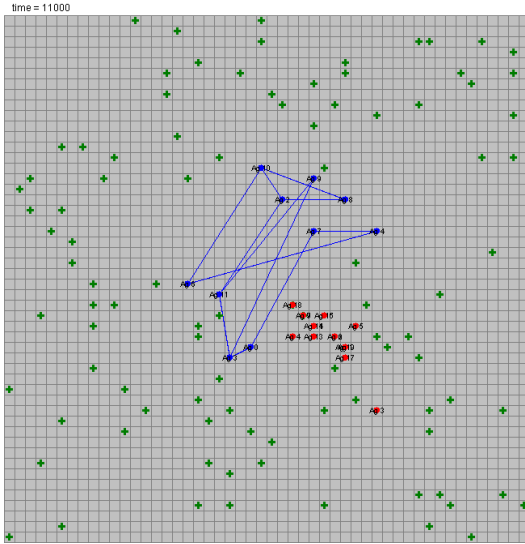


Figure 2: Snapshot of Simulation Run

Table 1: Simulation Parameter Settings

<b>Weapon range:</b>	8 grid squares
<b>Weapon accuracy:</b>	33%
<b>Red Sensor range:</b>	4 grid squares
<b>Blue Sensor range:</b>	1, 2, 4, 8, or 16 grid squares
<b>Sensor accuracy:</b>	100%
<b>Average time between sensor scans:</b>	80 time units
<b>Average time between shooting:</b>	80 time units
<b>Time to transmit a message across a link:</b>	4, 160, 6400, or infinity time units
<b>Maximum Red movement speed:</b>	200 time units per grid square
<b>Maximum Blue movement speed:</b>	50, 100, 200, 400, or 800 time units per grid square

The simulation continues until all the agents on one or the other side are annihilated, or the time limit  $t_{\max}$  is reached, or all the “items” are picked up.

As a measure of combat performance for the friendly force, we use of the natural logarithm of the Adjusted Loss Exchange Ratio (ALER). To be precise, if  $C_h$  are hostile casualties (ranging from 0 to 20), and  $C_f$  are friendly casualties (ranging from 0 to 12), our combat performance score  $S$  is given by:

$$S = \ln \text{ALER} = \ln \left( \frac{1 + C_h}{1 + C_f} \right)$$

This measure of effectiveness has the advantage of being symmetric (inverting the ratio merely changes the sign of the result), and we have used it with success in previous studies [7],[10]. It avoids division by zero, and has better statistical properties than the more commonly used loss exchange ratio  $C_h/C_f$ [7].

Combat scores ranged from  $-2.56$  (corresponding to annihilation of all 12 Blue agents, with no Red casualties), to  $3.04$  (corresponding to annihilation of all

20 Red agents, with no Blue casualties). A score of 0 corresponds to equal Red and Blue casualties.

Of course, the mission of the friendly “Blue” force is to pick up items, rather than to engage in combat for its own sake, and so we also counted the number of items successfully picked up during each simulation run.

To reduce random noise, the score for each randomly generated network was averaged over 10 simulated combat sessions. These averages were calculated for the 100 combinations of parameters shown in Table 1.

In previous experiments [7],[10], network topologies had a significant impact on performance, and so, for each combination of parameters, the experiment was repeated with five different network topologies: ring, star, hexagonal prism, hexagonal prism with diagonals, and truncated tetrahedron. Finally, the entire experiment was repeated a second time, in order to double the number of data points.

Unlike the previous experiments, the network topologies in this experiment did not affect performance, because the effect of networking was fairly subtle (the impact of a forty-fold increase in network delay was noticeable, but not the two-fold increase due to network topology). Consequently, network topologies are not discussed further in this paper.

The total of 10,000 simulated combats (1000 data points) took 50 hours to run on a 2.2 GHz Pentium 4.

## 5. RESULTS: COMBAT SCORES

By far the most important factor determining combat scores was the Blue sensor range. This was statistically extremely significant, with the probability that the result was due to chance being less than  $10^{-27}$  (all factors discussed in this paper were significant at least at this level, using regression analysis or ANOVA as appropriate). We discuss short sensor ranges (1 or 2), medium sensor ranges (4), and long sensor ranges (8 or 16) separately.

Visualisation of results is an important part of Defence simulation experiments. CAVALIER integrates statistical analysis and visualisation tools in order to help with this (Figures 5 and 7 were produced this way). It also exports data to Microsoft Excel for visualisation purposes (Figures 3, 4, and 6 were produced this way).

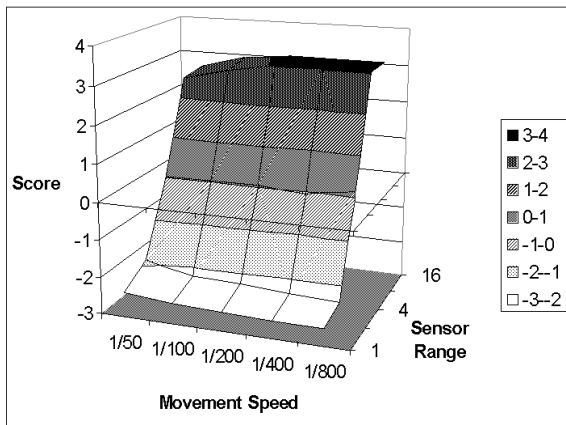
### 5.1 Combat Scores: Short Sensor Range

For Blue sensor ranges of 1 or 2, combat scores were very low, as shown in Table 2, and at the front of Figures 3 and 4. This was because Red units encountered by Blues began firing at a distance of 4 grid squares (the Red sensor range), while they were still invisible to Blue’s sensors.

Scores increased very slightly when Blue movement speeds  $M$  were fast, as shown on the front left of Figure 3. This was because there are two ways of avoiding a threat: being able to see it at a distance, or being able to move rapidly away from it. Long-range sensors and fast movement are alternative means of providing an **agile** force, which can respond rapidly to threats (quickly and accurately destroying the threat is a third form of agility).

**Table 2:** Combat Score Predictors

Sensor Range	Average Score ( $S$ )	Score Predictors
1	-2.48	$-2.11 + 0.07 (\log M)$
2	-2.22	$-1.41 + 0.15 (\log M)$
4	0.04	0.37, if $N = 4$ -0.06, if $N > 4$
8	2.89	$1.92 - 0.18 (\log M)$
16	2.93	$2.17 - 0.14 (\log M)$



**Figure 3:** Average Combat Scores as a Function of Blue Movement Speed and Sensor Range

In Table 2, the effect is expressed in terms of the natural logarithm of the movement speed  $M$ . This logarithm ranged from  $-7$  to  $-4$ .

Scores for short Blue sensor ranges did not depend at all on the network delay  $N$ . This was because there is no benefit in using the network to transfer low quality information of purely local relevance. Indeed, if information overload and network cost issues were taken into account, attempting to transfer such information would actually be of negative benefit.

To put these results in a historical context, consider the age of sailing ships [12], where sensors were restricted to line-of-sight. Here there would have been no advantage to networking distant ships, since they would have had no relevant information to pass to each other (nearby ships did have such information, and were networked optically). In this age, it was increased sailing speed and improved sensors that were necessary.

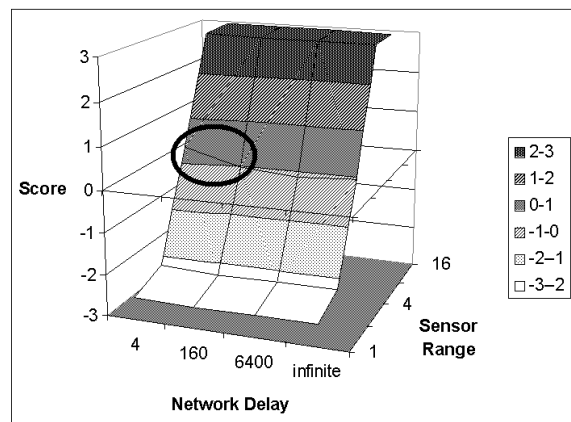
## 5.2 Combat Scores: Medium Sensor Range

For Blue sensor range 4 (equal to the Red Sensor range), combat scores were close to zero, with

approximately equal numbers of Red and Blue casualties, as shown in Table 2.

Movement speed  $M$  had no effect on scores, as shown at the middle of Figure 3, because the opposite effects of movement speed at short and long sensor ranges cancelled each other out. However, the network delay  $N$  did have an effect.

For network delays  $N$  of 160 time units or more, which were too slow to provide a combat benefit, the average score was  $-0.06$  (corresponding to annihilation of all 12 Blue agents, with 11 Red casualties). For a fast network, with a delay of only 4 time units in transmitting a message across a link, the average score increased to 0.37 (corresponding to Red losing 18 out of 20 agents in annihilating the 12 Blue agents). This network advantage is highlighted on Figure 4.



**Figure 4:** Average Combat Scores as a Function of Blue Network Delay and Sensor Range

The network advantage arises when the network provides sensor information to Blue agents which are in firing range of a target which they cannot themselves see. This network advantage would become greater in the presence of dedicated surveillance agents. As a general principle, the network is of benefit only when each agent's own information is inadequate, and at the same time, other agents have relevant information.

## 5.3 Combat Scores: Long Sensor Range

For Blue sensor ranges of 8 or 16, combat scores were very high, as shown in Table 2. Typically, the Red agents were destroyed at the cost of few, if any, Blue casualties.

Scores were not affected by network delay, as shown at the top of Figure 4. This was because Blue agents could shoot Red agents at maximum range, before the Red agents could see them. Since the local information of Blue agents was already adequate, there was no benefit in obtaining additional information via the network. Indeed, if data fusion and network costs were taken into account, attempting to transfer information under such circumstances would actually be of negative benefit.

This observation was confirmed by previous studies involving an air strike scenario [13]. In the case of a

distributed architecture and moderate operational tempo, that study showed (as would be expected) a positive benefit in sharing information when sensors were poor, but not when sensors were good.

In configurations where agents are not identical, so that “shooter” agents have poor on-board sensors, and information must be collected from specialised “sensor” agents, the network will be much more important, as demonstrated in previous simulation studies [7].

For long Blue sensor ranges, physical movement speed  $M$  also had an effect on combat scores, as shown at the top of Figure 3. However, in this case, fast movement **decreased** combat scores. This was because sensor data was collected only at intervals (on average, every 80 time units), and fast movement speed allowed agents to move outside their circle of perfect information.

However, although fast movement had a small negative effect on combat scores, Section 6 shows that it had a very large positive effect on overall mission effectiveness (items picked up).

This is an important issue, because organisations may become addicted to high-quality information. The drive to maintain this high-quality picture may actually reduce agility by slowing down responses. Such addiction can result from the **certainty effect** [14], in which people become risk-averse when faced by choices which have sure positive outcomes.

Van Creveld [15] discusses how the drive to create certainty in the military sphere has resulted in larger and larger headquarters with more and more rigid processes. This rigidity leads to a reduced ability to respond to new kinds of threats. It is partly in response to this decrease in flexibility that military Special Forces [16] have tended to develop their own independent and more flexible command structures.

In a business environment, organisations with sophisticated business intelligence systems are often pressured to delay new operations until the business intelligence systems can be modified to deal with them. A common solution is to create subsidiary companies, which are smaller and more agile, and less constrained by parent company processes.

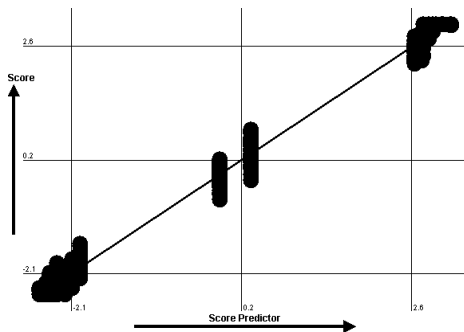


Figure 5: Combined Predictor for Combat Scores

### 5.4 Combat Scores: Summary

Table 2 shows the best predictors of combat scores for the different sensor range cases, and Figure 5 shows the combined predictor. This predicts combat scores quite accurately, with a correlation of 0.998. The two cases for a network delay of 4 are clearly visible in the centre of the graph.

### 6. MISSION RESULTS: ITEMS PICKED UP

The purpose of the Blue agents in this experiment was to pick up as many items as possible, not to engage in combat for its own sake. The number of items picked up was determined by two factors, operating independently. These were the movement speed  $M$  (predicting 54% of the variance), and the combat score  $S$  (predicting 37% of the variance). Essentially the combat score  $S$  determined the average number of Blue agents available, and the movement speed  $M$  determined how many grid squares they could visit, and hence how many items they could pick up. Figure 6 illustrates this dependence.

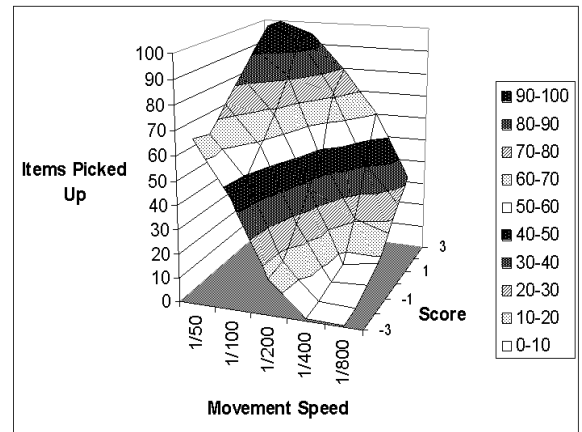


Figure 6: Items Picked Up as a Function of Blue Movement Speed and Combat Score

The average number of items picked up ( $P$ ) fitted the linear regression equation:

$$P \approx 26.1 (\log M) + 8.98 S + 184.4$$

Figure 7 shows this regression, which predicts 91% of the variance (a correlation of 0.96), although some nonlinearities and random effects are visible. Figure 8 summarises the various statistical relationships.

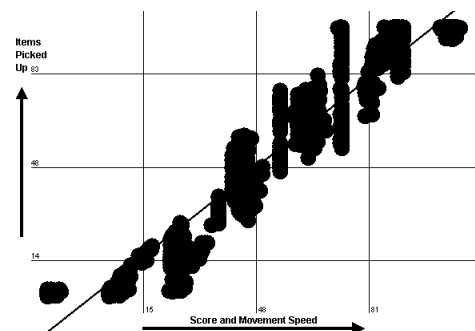


Figure 7: Items Picked Up as a Function of Regression Equation

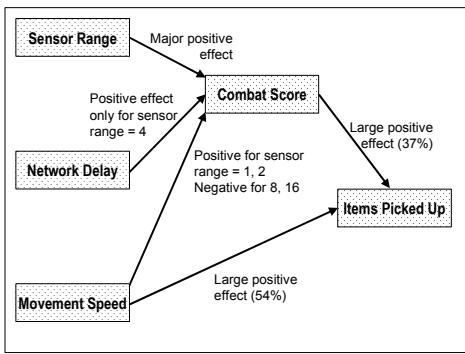


Figure 8: Summary of Statistical Relationships

## 7. CONCLUSIONS

In this paper, we have outlined two extensions to the CAVALIER network farming tool: improved event management to handle a wider range of timescales, and neural-network-based learning to allow reinforcement of the most effective behaviours. Through a simple experiment, we have illustrated the tool's simulation, statistical analysis, and visualisation capabilities.

The experiment reported here explored the relationship between sensor range, networking, and speed of movement of a networked military force carrying out a search activity under enemy fire. In particular, the experiment explored agility: the ability to perceive and respond quickly enough to a threat. Statistical analysis of the experimental results illustrated five networking and agility principles, which were as follows:

- Agents with limited information of purely local relevance gain no benefit from networking.
- Agents with sufficient information of their own gain no benefit from networking.
- Agents with moderate amounts of information gain a competitive advantage by sharing information.
- Agents require either early awareness of upcoming threats, or the ability to respond to them very rapidly.
- High-quality information creates a situation where motion causes risk; but if this breeds risk averseness, overall mission success may suffer.

Three of these principles relate to the fact that networking is only sometimes beneficial, and two relate to agility trade-offs.

## 8. ACKNOWLEDGEMENTS

The author is grateful to Sally Long for coding parts of the simulation software; and to Gina Kingston, Robert Mun, and two anonymous referees for comments on an earlier draft of this paper.

## REFERENCES

1. Horne, G., Johnson, S. & Meyer, T. (2000) "Project Albert: Overview," *Military Operations Research Society Workshop on Advancing C4ISR Assessment*, Army War College, Carlisle, Pennsylvania, October.

2. US Marine Corps (2005), Project Albert Website, accessed May 2005: [www.projectalbert.org](http://www.projectalbert.org)
3. Brandstein, A.G. & Horne, G.E. (1998), "Data Farming: A Meta-technique for Research in the 21st Century," *Maneuver Warfare Science 1998*, US Marine Corps Combat Dev't Command Publication: [www.mcwl.quantico.usmc.mil/Albert/research/data\\_farming\\_rtf](http://www.mcwl.quantico.usmc.mil/Albert/research/data_farming_rtf)
4. Horne, G. E. (1997), "Data Farming: A Meta-technique for Research on 21st Century Questions," briefing presented at the US Naval War College, Newport, Rhode Island, November.
5. Dekker, A.H. (2004), "Network Farming for the Analysis of Complex Network Systems," Presentation to the Complex Adaptive Systems in Defence Workshop, University of Adelaide, July.
6. Dekker, A.H. (2005), "Simulating Network Robustness for Critical Infrastructure Networks," *Proc. 28th Australasian Computer Science Conference*, V. Estivill-Castro (Ed.), Newcastle, Australia. Conferences in Research and Practice in Information Technology, vol 38, p59–68. Available at [crpit.com/confpapers/CRPITV38Dekker.pdf](http://crpit.com/confpapers/CRPITV38Dekker.pdf)
7. Dekker, A.H. (2004), "Simulating Network Robustness: Two Perspectives on Reality," *Proceedings of SimTecT 2004 Simulation Conference*, Canberra, pp 126–131.
8. Graybeal, W.J. & Pooch, U.W. (1980) *Simulation: Principles and Methods*, Winthrop Publishers.
9. Dekker, A.H. & Piggott P.K. (1995), "Robot Learning with Neural Self-Organization," *Proceedings of Robots for Australian Industries*, National Conference of the Australian Robot Association, Melbourne, 5–7 July, pp 369–381.
10. Dekker, A.H. (2005), "Network Topology and Military Performance," in Zerger, A. & Argent, R.M. (eds), *MODSIM 2005 International Congress on Modelling and Simulation*, Modelling and Simulation Society of Australia and NZ, December, pp 2174–2180, ISBN 0-9758400-2-9: [www.mssanz.org.au/modsim05/proceedings/papers/dekker.pdf](http://www.mssanz.org.au/modsim05/proceedings/papers/dekker.pdf)
11. Reynolds, C.W. (1987) "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Computer Graphics*, vol. 21, no. 4, pp 25–34.
12. Keegan, J. (1988) *Battle at Sea*, Pimlico.
13. Dekker, A.H. (2002), "C4ISR Architectures, Social Network Analysis and the FINC Methodology: an Experiment in Military Organisational Structure," DSTO Report DSTO-GD-0313, January.
14. Kahneman, D. & Tversky, A. (1979) "Prospect Theory: An Analysis of Decision under Risk," *Econometrica*, vol. 47, no. 2, pp 263–292, March.
15. van Creveld, M. (1985), *Command in War*, Harvard University Press.
16. Neillands, R. (1998) *In the Combat Zone: Special Forces Since 1945*, Orion.
17. Dekker, A.H. (2006), "Measuring the Agility of Networked Military Forces," *Journal of Battlefield Technology*, vol. 9, no. 1, March, pp 19–24.
18. Friman, H. & Horne, G.E. (2005) "Using Agent Models and Data Farming to Explore Network Centric Operations," *Proc. Winter Simulation Conference*, Kuhl, M.E., Steiger, N.M., Armstrong, F.B. & Joines, J.A., eds. [www.informs-cs.org/wsc05papers/132.pdf](http://www.informs-cs.org/wsc05papers/132.pdf)
19. Davis, P. (2005) "New Paradigms and New Challenges," *Proc. Winter Simulation Conference*, Kuhl, M.E., Steiger, N.M., Armstrong, F.B. & Joines, J.A., eds. [www.informs-cs.org/wsc05papers/128.pdf](http://www.informs-cs.org/wsc05papers/128.pdf)