11TH ICCRTS

COALITION COMMAND AND CONTROL IN THE NETWORKED ERA

Agility in Networked Military Systems: A Simulation Experiment

Topics: C2 Analysis, C2 Modeling and Simulation, Network-Centric Metrics

Anthony H. Dekker

Defence Science and Technology Organisation
Department of Defence, Canberra ACT 2600
Australia

Telephone: +61 2 6256 6151
Facsimile: +61 2 6256 6180

dekker@ACM.org

Paper I-031

Agility in Networked Military Systems: A Simulation Experiment

Anthony H. Dekker
Defence Science and Technology Organisation
Department of Defence, Canberra ACT 2600
dekker@ACM.org

Abstract

This paper describes some extensions to our CAVALIER agent-based simulation system, a tool for studying the performance of networked military organisations. Specifically, we have improved event handling and added neural-network-based learning. The CAVALIER tool is intended for collaborative workshops in the style of Project Albert. To demonstrate this use, the paper describes in detail an experiment studying networking and agility. We also include a comparison with the MANA simulation tool from New Zealand. Our experiment illustrates several principles of networked operation, listed at the end of the paper: first, that agents require either early awareness of upcoming threats, or the ability to respond rapidly; second, that there is little benefit in networking if agents already have enough information, or if they do not have any information worth sharing; and third, that high-quality information creates a situation where motion causes risk, yet if this breeds risk averseness, overall mission success may suffer.

Keywords: agent, agility, data farming, NCW, simulation.

1 Introduction

The US Marine Corps' **Project Albert** (Horne *et al* 2000, US Marine Corps 2005) introduced improved techniques for Defence analysis, so as 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 (Brandstein & Horne 1998):

- are open, rather than closed;
- are 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 oriented to the future rather than to the Cold War.

These techniques include **data farming** (Horne 1997, Friman & Horne 2005) for exploring the behaviour of parameterised models. We have extended data farming to **network farming** (Dekker 2004b, 2005a) 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. For example, performance in a simulation model may be analysed in terms of the average number of "hops" between nodes in a network. We perform network farming using the CAVALIER tool, which integrates agent-based simulation, calculation of network metrics, statistical analysis, and visualisation of results.

Another 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.

In this paper, we describe improvements to the agent-based simulation system embedded within the CAVALIER network farming tool (Dekker 2003, 2004a). These improvements include the use of

event queues for more efficient event management, and **neural-network learning** to reduce the amount of fine-tuning of parameters required to obtain realistic behaviour.

We also describe a simulation experiment studying the performance of a networked military organisation, as a way of demonstrating how CAVALIER can be used to illustrate 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. A summary of different agility levels and factors is given in Dekker (2006a), and Figure 1 summarises some of those factors.

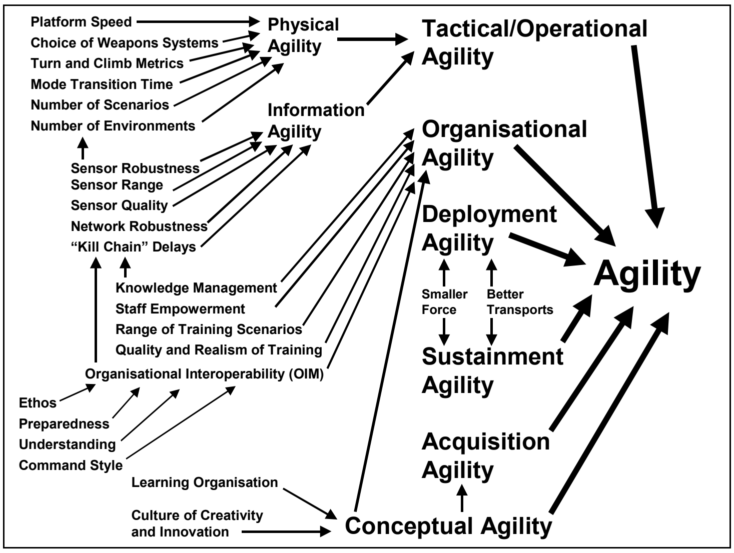


Figure 1: Summary of Factors Determining Agility

The US Army defines agility as "*the ability of friendly forces to act faster than the enemy*" (US Army 1997). Agility requires perceiving the threat from a sufficient distance, 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.

2 Simulation Improvements: Event Queues

The agent-based simulation systems used by Project Albert 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, or very slow, compared to the physical movement of agents in the world. When only telecommunications networks are involved, transmission speeds tend to be very fast, but when human processes are involved, transmission speeds tend to be slow, compared to physical movement. Figure 2 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** (Graybeal and Pooch 1980) 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 $Q_1,\ldots,Q_k$ (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.

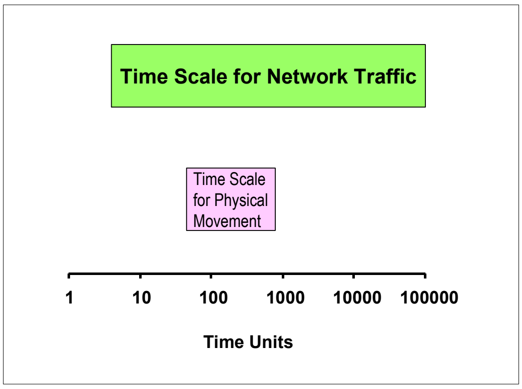


Figure 2: Time Scales for Network Traffic and Physical Movement

3 Simulation Improvements: 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 (1995). As in previous work (Dekker 2005b), 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 (1987).

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 α is a parameter controlling the speed of learning (typically 0.5), and β 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 (Davis 2005).

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 attempts to locate and pick up 100 "items" on a 50×50 discrete grid. While doing this, they 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." Figure 4 shows a snapshot of the simulation in progress, with the "items" shown as green crosses.

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 for the experiment.

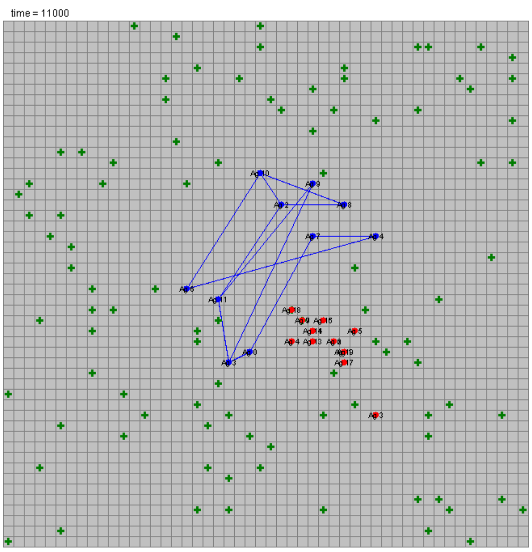


Figure 3: Snapshot of Simulation Run

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 (Dekker 2003, 2004a, 2005b). It avoids division by zero, and has better statistical properties than the more commonly used loss exchange ratio C_h / C_f (Dekker 2004a). For comparison purposes, Table 2 shows the range of loss exchange ratios and combat performance scores obtained in the experiment. 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 100 combinations of parameters shown in Table 1. A limited set of the results of this experiment have been previously reported in Dekker (2006b).

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

Table 1: Simulation Parameter Settings

Score (S)	Loss Exchange Ratio (Red:Blue)
–2.56	0:12
–2	1:12
–1	4:12
0	12:12
1	20:7
2	20:2
3.04	20:0

Table 2: Combat Performance Scores

Network topologies had a significant impact on performance in previous experiments (Dekker 2004a, 2005b), and so the experiment was repeated with five different network topologies, including a "star" topology and the four networks in Figure 4.

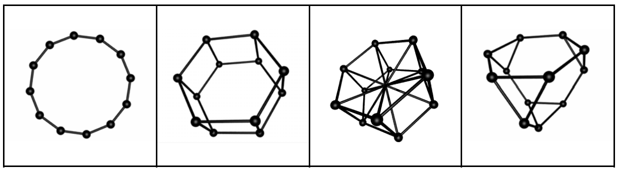


Figure 4: Four Network Topologies

Finally, the entire experiment was repeated a second time, in order to double the number of data points.

Unlike previous experiments (Dekker 2004a, 2005b), the network topologies in this experiment did not impact 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.

For comparison, a version of the experiment using New Zealand's MANA tool is described in Section 7.

5 Experimental 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} by linear regression (all factors discussed in this paper were significant at least at this level). 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 Project Albert workshops. CAVALIER integrates statistical analysis and visualisation tools in order to help with this (Figures 8 and 11 were produced this way). We also export data to Microsoft Excel for visualisation purposes (Figures 5, 6, 9, 10, 13, 14, and 15 were produced this way). These visualisation techniques support the use of CAVALIER in Project-Albert-style workshops.

5.1 Combat Scores: Short Sensor Range

For Blue sensor ranges of 1 or 2, combat scores were very low, as shown in Table 3, and at the front of Figures 5 and 6. 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.

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	$2.17 - 0.14 (\log M)$

Table 3: Combat Score Predictors

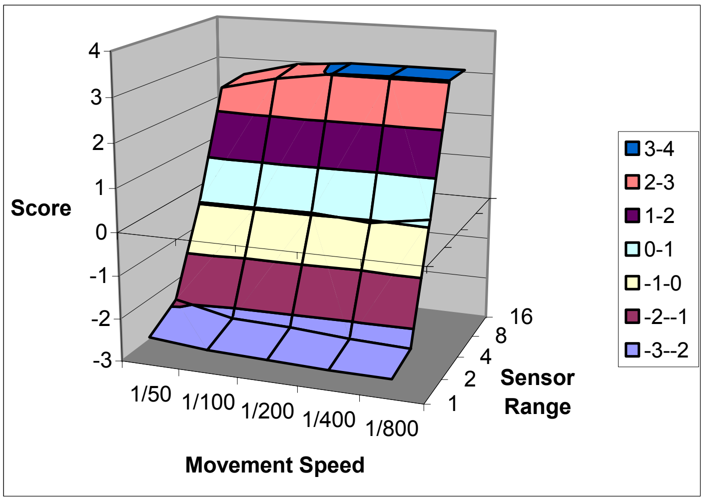


Figure 5: Average Combat Scores as a Function of Movement Speed and Sensor Range

Scores increased very slightly when Blue movement speeds M were fast, as shown on the front left of Figure 5. 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 rapid reaction are alternative means of providing an **agile** force, which can respond rapidly to threats. This applies to all the forms of agility in Figure 1.

In Table 3, the effect is expressed in terms of the natural logarithm of the movement speed M. This logarithm ranged from –7 to –4, and had a significant effect (at the 10^{-27} level by linear regression) for all Blue sensor ranges except 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 (Keegan 1988), 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 3.

Movement speed M had no effect on scores, as shown at the middle of Figure 5, 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 (significant at the 10^{-32} level by ANOVA on the $N = 4$ and $N > 4$ classes).

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 6.

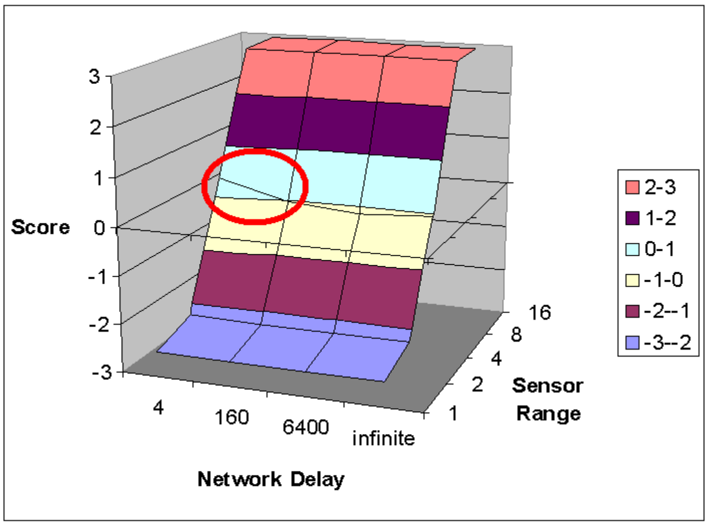


Figure 6: Average Combat Scores as a Function of Network Delay and Sensor Range

Figure 7 illustrates the network advantage, where networking provides sensor information about targets to Blue agents which are in firing range, but can themselves not see the target. Here three networked friendly agents (black dots) can engage the hostile agent (red X), which is in firing range of all three friendly agents (large open circles), even though it is in sensor range (blue circles) for only one agent. The hostile agent, meanwhile, can only retaliate against the one agent it can see.

This network advantage would become greater with non-identical agents, where some had longer sensor ranges than others.

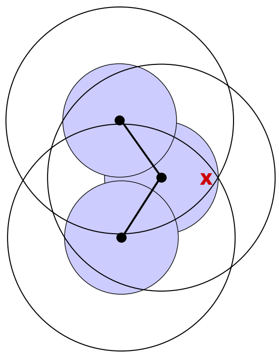


Figure 7: The Network Advantage

Figure 7 demonstrates the general principle that the network is of benefit only when each agent's own information is inadequate, but 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 3. 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 6. 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 (Dekker 2002). 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 (Dekker 2003, 2004a).

For long Blue sensor ranges, physical movement speed M also had an effect on combat scores, as shown at the top of Figure 5. 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 the 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** (Kahneman & Tversky 1979), in which people become risk-averse when faced by choices which have sure positive outcomes.

Van Creveld (1985) 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 (Neillands 1998) 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.

5.4 Combat Scores: Summary

Table 3 shows the best predictors of combat scores for the different sensor range cases, and Figure 8 shows the combined predictor. This predicts combat scores extremely 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.

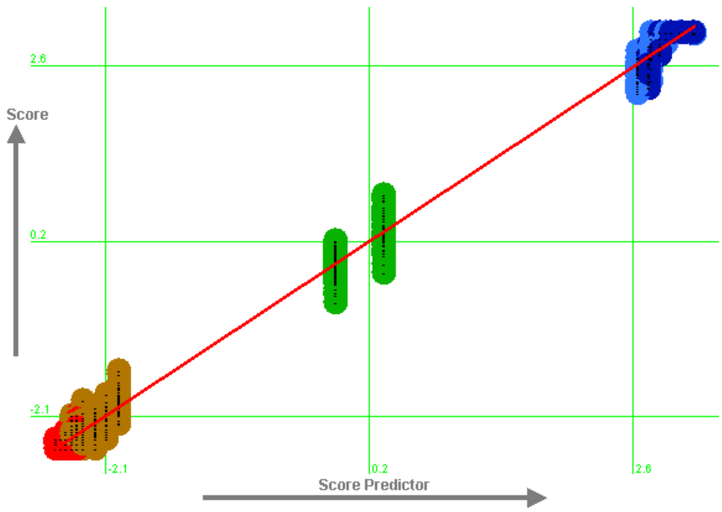


Figure 8: Combined Predictor for Combat Scores

6 Experimental 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 9 illustrates this dependence. For comparison with Figure 5, Figure 10 shows the number of items picked up as a function of movement speed and sensor range. Figure 10 illustrates the enormous drop in mission effectiveness at long sensor range when movement is slow, even though slow movement slightly improves combat scores.

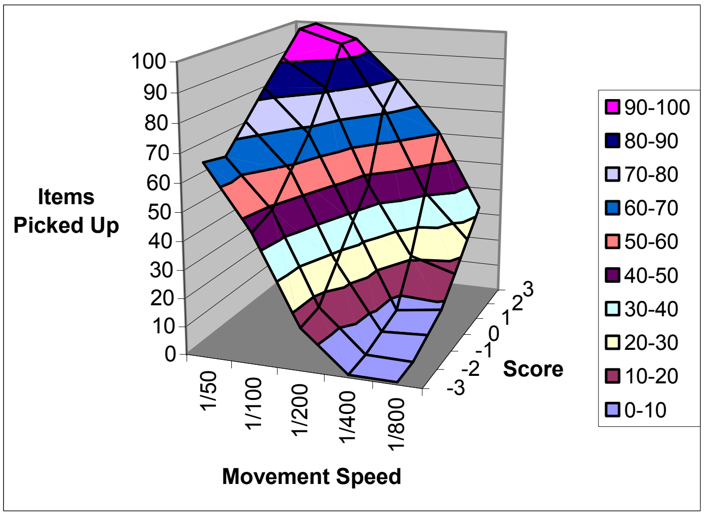


Figure 9: Items Picked Up as a Function of Movement Speed and Combat Score

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

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

Figure 11 shows this regression, which predicts 91% of the variance (a correlation of 0.96), although some nonlinearities and random effects are visible. Figure 12 summarises the various statistical relationships.

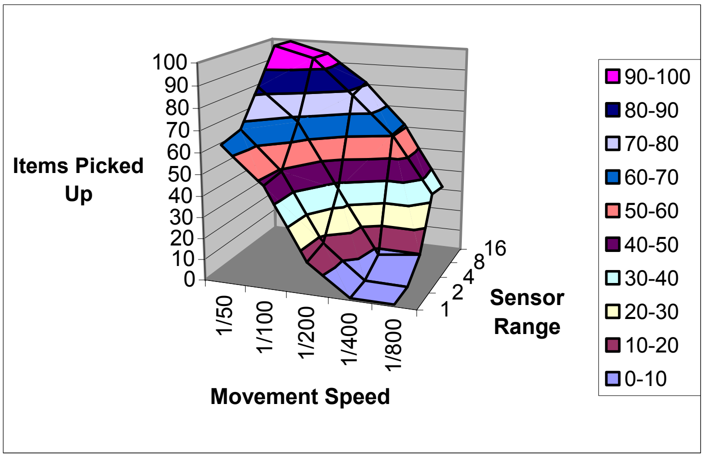


Figure 10: Items Picked Up as a Function of Movement Speed and Sensor Range

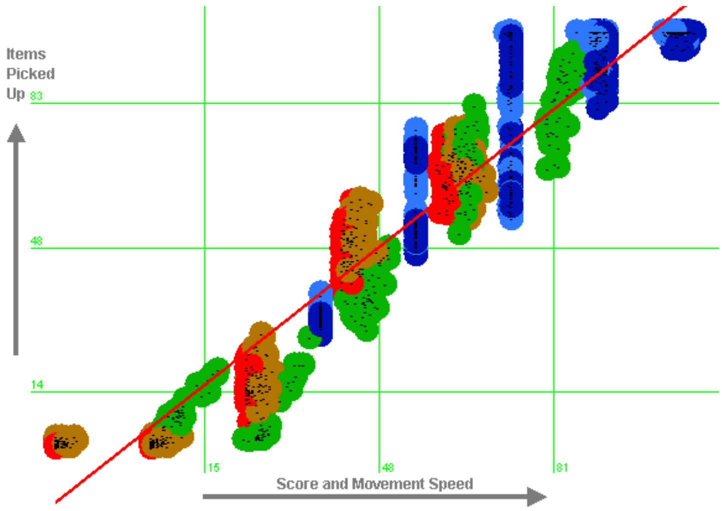


Figure 11: Items Picked Up as a Function of Regression Equation

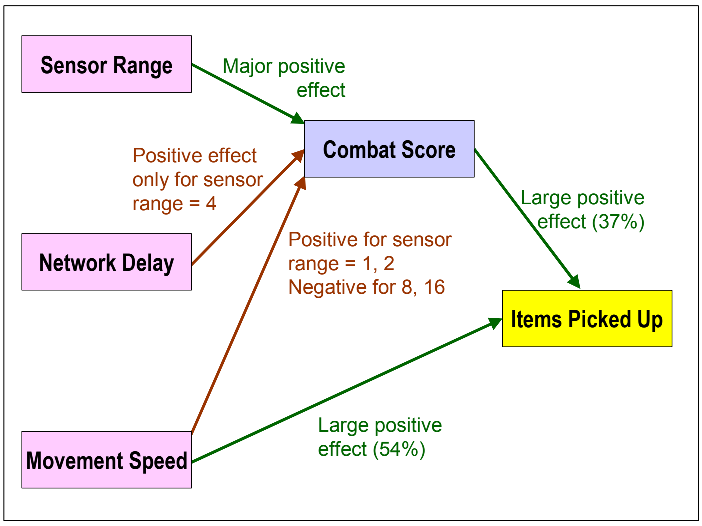


Figure 12: Summary of Statistical Relationships

7 Using MANA

The experiment was repeated using New Zealand's MANA tool (Lauren & Stephen 2002, Friman & Horne 2005), on a 200×200 grid and with a 500 time-step limit. Values for speed and sensor ranges were scaled up accordingly. It was not possible to exactly capture the experimental scenario within MANA, since MANA does not allow for exploration of the grid or for collection of "items" as possible goals. As an approximation, the "items" were treated as a special class of Red agents, which did not move or shoot, and which were "collected" by shooting them with a specialised weapon.

MANA has a less random movement rule than CAVALIER, and hence only 20 runs per combination of parameters were required to get meaningful results. In order to support experiments of this kind, CAVALIER includes a module for "patching" MANA's XML scenario files to include arbitrary network topologies constructed within CAVALIER. However, given the observation about network topologies noted in Section 4, only MANA's standard within-squad networking was used for this experiment.

Because of MANA's limited ability to perform multiple parameterised runs, the experiments were performed by hand using MANA's GUI interface. Consequently, the range of parameters used for the MANA experiment was more limited than with CAVALIER, and was concentrated on values previously identified as interesting.

Figure 13 shows the MANA equivalent of Figure 5 (for comparison, Red movement speed was 100, and Red sensor range was 15). Figure 13 has the same strong dependence of combat scores on sensor range as in Figure 5. However, the relationship between movement speed and combat scores is quite different. MANA appears to be less able than CAVALIER to model agility phenomena, presumably because CAVALIER's event-based time control is better able to capture phenomena such as "dodging out of the way" and "moving outside one's circle of perfect information."

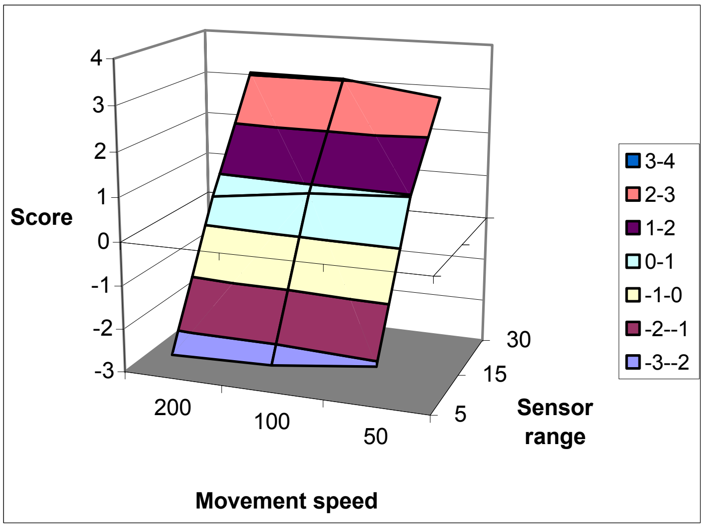


Figure 13: Average MANA Combat Scores as a Function of Blue Speed and Sensor Range

Figure 14 shows the MANA equivalent of Figure 6. In this case, the two figures tell the same story: the primary impact on combat scores is from sensor range, but there is a noticeable "network advantage" when Blue and Red sensor ranges are the same (both 15). This is moderately statistically significant ($p < 0.03$). Strangely, Blue agents seem to have an advantage over Red agents even when they have the same movement speed and Sensor range. This continues to hold even if the network delay is increased to be effectively infinite (i.e. longer than the simulation time). The reasons for this are obscure, but appear to result from an interaction with the agents representing "items."

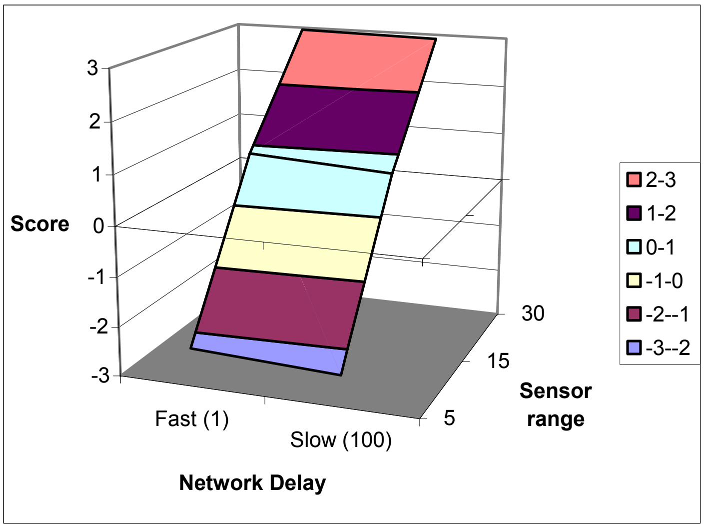


Figure 14: Average MANA Combat Scores as a Function of Blue Network Delay and Sensor Range

Figure 15 shows the MANA equivalent of Figure 9. Although both figures show that the number of items picked up increases with the combat score and the Blue movement speed, Figure 15 shows several irregularities not visible in the CAVALIER results. The regression equation for the number of items picked up is similar to that for CAVALIER, but predicts less of the variance (84% instead of 91%):

$$P_{\mathrm{MANA}} \approx 21.3\,(\log M) + 9.56\,S - 47.0$$

Because of the complexity of the various parameters controlling MANA, it is difficult to determine the cause of the irregularities in Figure 15. They may reflect inappropriate settings of movement weight parameters.

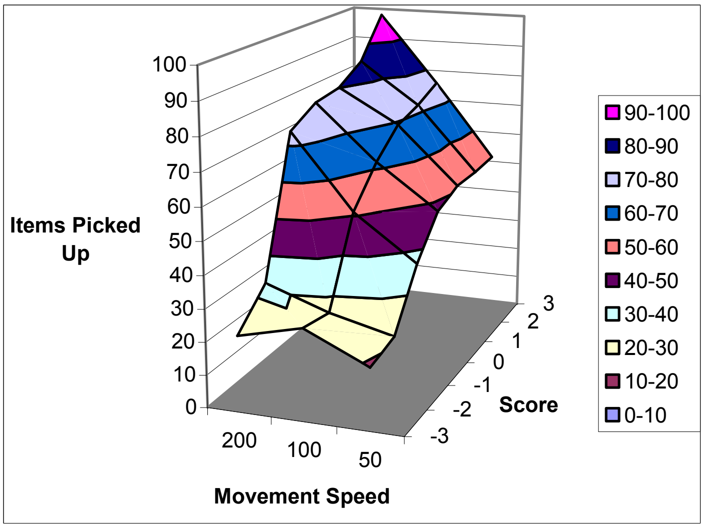


Figure 15: Items Picked Up as a Function of Blue Movement Speed and Combat Score

In general, although some of our experimental results with CAVALIER have been replicated in MANA, some of the more subtle aspects of agility proved impossible to replicate. However, this is not surprising: the simulation improvements to CAVALIER were intended precisely to be able to capture these aspects. The use of event queues in CAVALIER allows more accurate simulation of phenomena at different timescales, and the use of neural-network learning makes the result less dependent on precise settings of weight parameters.

8 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. We have showed how the CAVALIER tool can be used for workshops by "walking through" an experiment. CAVALIER is well suited to these purposes, because it integrates simulation, statistical analysis, and visualisation capabilities within the one tool.

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, shown in Figure 16. Three of these principles relate to the fact that networking is only sometimes beneficial, and two relate to agility trade-offs.

This is a rich list of observations, and demonstrates that by using CAVALIER within a workshop setting (in the style of Project Albert) useful insights on networked operation would be obtained.

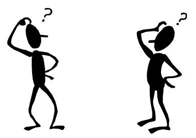
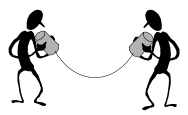
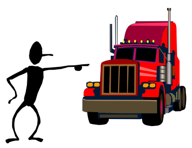
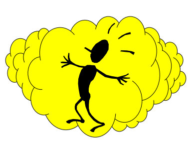
	Agents with limited information of purely local relevance gain no benefit from networking.
	Agents with sufficient information of their own gain no benefit from networking.
	Agents with **moderate** amounts of information gain a competitive advantage by sharing information.
	Agents require either early 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.

Figure 16: Principles Illustrated by Experimental Results

9 Acknowledgements

The author is grateful to Sally Long and Sarah Smith for coding parts of the simulation software; and to Gina Kingston and Robert Mun for comments on an earlier draft of this paper. Figure 4 was produced by interfacing CAVALIER to the Persistence of Vision™ ray-tracing package.

10 References

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 Development Command Publication. www.mcwl.quantico.usmc.mil/Albert/research/data_farming.rtf

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

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. www.dsto.defence.gov.au/corporate/reports/DSTO-GD-0313.pdf

Dekker, A.H. (2003), "Using Agent-Based Modelling to Study Organisational Performance and Cultural Differences," *Proc. MODSIM 2003 International Congress on Modelling and Simulation*, Townsville, Queensland, 1793–1798. Available electronically at mssanz.org.au/modsim03/Media/Articles/Vol 4 Articles/1793-1798.pdf

Dekker, A.H. (2004a), "Simulating Network Robustness: Two Perspectives on Reality," Proceedings of SimTecT Simulation Conference, Canberra, 126–131. www.acm.org/~dekker/Dekker.SimTecT.pdf

Dekker, A.H. (2004b), "Network Farming for the Analysis of Complex Network Systems," Presentation to the Complex Adaptive Systems in Defence Workshop, University of Adelaide, July.

Dekker, A.H. (2005a), "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*, **38**, 59–68. Available electronically at crpit.com/confpapers/CRPITV38Dekker.pdf

Dekker, A.H. (2005b), "Network Topology and Military Performance," in Zerger, A. and Argent, R.M. (eds), *MODSIM 2005 International Congress on Modelling and Simulation*, Modelling and Simulation Society of Australia and New Zealand, December, pp 2174–2180, ISBN: 0-9758400-2-9: www.mssanz.org.au/modsim05/proceedings/papers/dekker.pdf

Dekker, A.H. (2006a), "Measuring the Agility of Networked Military Forces," *Journal of Battlefield Technology*, **9** (1), March, pp 19–24.

Dekker, A.H. (2006b), "Simulating Agility for Networked Military Systems," Proceedings of SimTecT Simulation Conference, Melbourne, pp 101–106.

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. www.acm.org/~dekker/robot.pdf

Friman, H. & Horne, G.E. (2005), "Using Agent Models and Data Farming to Explore Network Centric Operations," *Proceedings of the 2005 Winter Simulation Conference*, Kuhl, M.E., Steiger, N.M., Armstrong, F.B. & Joines, J.A. (eds). www.informs-cs.org/wsc05papers/132.pdf

Graybeal, W.J. & Pooch, U.W. (1980), *Simulation: Principles and Methods*, Winthrop Publishers.

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.

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.

Kahneman, D. & Tversky, A. (1979), "Prospect Theory: An Analysis of Decision under Risk," *Econometrica*, **47** (2), 263–292, March.

Keegan, J. (1988), *Battle at Sea*, Pimlico.

Lauren, M.K. & Stephen, R.T. (2002), "Map-Aware Non-Uniform Automata (Mana)—A New Zealand Approach to Scenario Modelling," *Journal of Battlefield Technology*, **5** (1), March.

Neillands, R. (1998), *In the Combat Zone: Special Forces Since 1945*, Orion.

Reynolds, C.W. (1987), "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Computer Graphics*, **21**(4), 25–34.

US Army (1997), *Field Manual FM1-111: Aviation Brigades*. www.cavalrypilot.com/fm1-111/toc.htm

US Marine Corps (2005), Project Albert Website, accessed May 2005. www.projectalbert.org

van Creveld, M. (1985), *Command in War*, Harvard University Press.