

Investigating Organisational Structures with Networks of Planning Agents

Anthony Dekker
Defence Science and Technology Organisation
(DSTO), Canberra, Australia
dekker@ACM.org

Lavindra de Silva
RMIT University,
Melbourne, Australia
ldesilva@cs.rmit.edu.au

Abstract

In this paper, we present a simulation model specifically for investigating formal and informal organisational structures. CAVALIER-NPA (Networks of Planning Agents) is an agent-based simulation written in Java, where agents can communicate and do BDI-based planning. We describe a simple experiment which explores the impact of planning time, communications delays, and workload sharing on a simple mission. Analysis of the experimental results underlines the importance of efficient communication, the balance between planning and reactive behaviour, and the benefits of workload sharing within informal organisational networks.

1. Introduction

Modern communications and information technologies have had a significant impact on both civilian and military organisations. This has led several researchers to ask: what is the best organisational structure for military forces, given these technologies? [1, 2, 3].

In previous work, we have used agent-based simulations to explore this question [4, 5, 6, 7, 8]. These simulations used the CAVALIER tool suite for investigating organisational structures. CAVALIER contains modules for visualising networks [9] and analysing networks [10] as well as simulating networks of agents. However, this work was limited by the fact that the simple agents used were not realistic abstractions of real people or real organisational units. In particular, **planning** is an important aspect of military Command and Control, and realistic simulations of organisational behaviour should incorporate planning.

Also of interest are **informal networks** which complement organisational hierarchies by sharing information and work between peers. Such informal networks helped Toyota to deal with the 1997 fire

which destroyed its sole source of brake fluid proportioning valves [11]. Strong links between dozens of Toyota subsidiaries allowed production of valves to resume within less than a week. Organisational culture, strong trust, information-sharing arrangements, a history of staff exchanges, and effective management contributed to the strength of these links. Similar benefits for informal networking occur in the military environment [12]. Figure 1 illustrates how an informal network complements a formal hierarchy.

There is also theoretical evidence reinforcing the need for informal horizontal linkages, in the form of mathematical analysis of organisational structures [13], but such work is of limited value unless it explicitly models organisational goals and plans, as we do here.

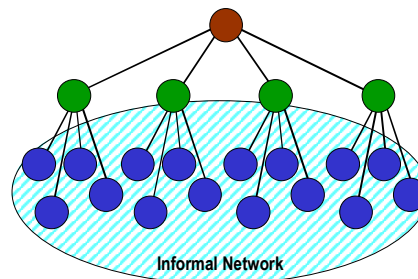


Figure 1: Hierarchy complemented by informal network

Agent-based simulations are useful for exploring military organisational structures such as this, because they are capable of modelling realistic human behaviours such as setting goals and planning, as well as modelling human limits on rationality. Less capable modelling tools such as Petri Nets and Queuing Theory are not able to effectively model these characteristics.

A popular architecture for agent systems is the **Belief-Desire-Intention (BDI)** model [14], which has its roots in Bratman's theory of practical reasoning [15] and Dennett's theory of intentional systems [16]. Beliefs correspond to a (possibly incorrect and partial) representation of the world, desires (or goals) to a future world state that is sought, and intentions to an

action plan for achieving that state. Some BDI-based systems (e.g. JACK [17]) have a fixed “plan library” from which actions are chosen. However, a more powerful approach is for agents to have access to a general-purpose planning facility, so that they can create new action plans in real time [18]. In particular, this allows agents to respond to unusual world states without each response being explicitly programmed.

Human decision-making is characterised by **bounded rationality** [19, 20]. People do not have unlimited planning capability, and so one reason for constructing hierarchical organisations is to decompose complex problems. We are interested in examining such organisations, as well as studying the impact of informal networks.

Bounded rationality in human beings can be modelled by **limited deliberation** [21, 22], where planning is limited by time and/or other resource constraints. For simulation purposes, constraints can be imposed on the number of planning steps that can be performed. This in turn leads to feasibility limits on the depth of planning that is worthwhile considering.

In this paper, we describe a new simulation module for CAVALIER: Networks of Planning Agents (CAVALIER-NPA). This module uses a BDI-like architecture and a combination of reactive behaviour and depth-limited best-first planning to achieve agent goals. We describe a simple experiment using CAVALIER-NPA to investigate the effect of informal networks in achieving organisational goals.

2. Networks of Planning Agents

CAVALIER-NPA (Networks of Planning Agents) is an event-based simulation written in Java, which extends the previous agent-based simulation models within CAVALIER [4, 5, 6, 7, 8]. It incorporates:

- A **grid-based world** containing obstacles, agents, and various kinds of items which can be picked up, carried, and dropped.
- **Sensors** of varying quality inside agents.
- A **world representation** inside each agent which provides a (possibly incorrect and partial) image of the world. The world representation can be updated with information from sensors and other agents.
- **Communication links** between agents for transmitting sensor information, agent status, and requests for assistance, with adjustable delays between message transmission and receipt.
- **Goal objects** which can be passed between agents. Goal objects include methods for testing whether they have been achieved within a particular world representation, and heuristic methods for rating the desirability of various actions in the context of a

particular world representation. Agents are idle unless they have one or more unsatisfied goals. Figure 2 shows the key operations on goal objects.

- **Actions**, including movement and picking up or dropping items, which can be executed.
- **Plans**, which are sequences of actions.
- A default **reactive behaviour** which takes a random choice of currently available actions, biased according to their desirability in the context of the current goal and world representation.

<p>Class Goal: boolean satisfiedTest (WorldRepresentation rep) float actionDesirability (WorldRepresentation rep, Action act) String pddlRepresentation ()</p>

Figure 2: Key methods for goal objects

CAVALIER-NPA also includes a choice of **planners** inside agents for producing a plan, given the current goal and world representation. One option is a simple depth-limited best-first planner written in Java, which uses the same desirability heuristics as the reactive behaviour. This planner does forward state space search simply by applying possible actions to the current world representation to obtain predicted future world representations. Because of the complexity of testing world representations for equality, there is no attempt to recognise states which have been previously visited. For each agent, the planner can be controlled by setting the maximum number of planning steps which can be considered. This allows some agents (representing headquarters) to spend most of their time planning, while other agents spend most of their time acting. Failure occurs if the planner does not find a plan within the given planning limit, in which case the reactive behaviour is applied. This planner takes approximately 1 millisecond per planning step on a 2.2 GHz Pentium 4. A formal semantics of time-limited planning has been developed as an extension of previous work [18] by the second author.

The other choice of planner within CAVALIER-NPA is an interface to the Metric-FF planner [23, 24]. This uses the Java Native Interface (JNI) to call Metric-FF (which is written in C), and passes descriptions of the current world representation and goal to the planner using PDDL, the Planning Domain Definition Language [25]. This required modifying Metric-FF to allow multiple re-entrant calls, and to take PDDL input as strings rather than as files.

The Metric-FF planner is significantly faster than the simple best-first planner for some problems. However, the use by Metric-FF of the GRAPHPLAN planning heuristic [26] results in considerable pre-processing, which makes it less suitable for limited

deliberation planning, or for dealing with highly complex state spaces (even with a 1GB memory allocation, Metric-FF ran out of memory in attempting to plan in the experiment described below). In addition, PDDL is not sufficiently descriptive to encode all the desirability heuristics for actions. The simple best-first planner is therefore more effective under certain circumstances. Each agent in CAVALIER-NPA can use either of the two planners. However, for the experiment described in the remainder of this paper, the simple best-first planner was most effective, and the Metric-FF interface was not used.

For the specific task of movement to a specified location around obstacles, far more effective special-purpose planners are available, and have been used in previous versions of CAVALIER [5]. However, we wished to specifically examine general-purpose planning in the context of a problem which was sufficiently difficult for the planner to sometimes fail. This is a situation that real people experience in dealing with real military problems. Real people respond to these problems with hierarchical decomposition and informal networks, and these are the phenomena we wished to explore in the following experiment.

3. The experiment

In order to investigate informal networks within hierarchical organisations, we conducted an experiment with 21 agents organised as in Figure 1.

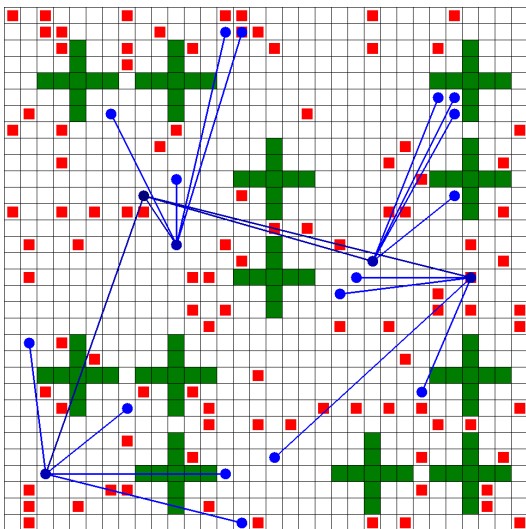


Figure 3: Example world

However, all messages between peers were still sent via a network which matched the organisational hierarchy. The mission of these agents was to pick up 100 items in a randomly generated 32x32 grid-based world containing 12 cross-shaped obstacles. Figure 3 shows one of the randomly generated worlds used. The small squares are the items to be picked up, and the networked circles are the agents.

The commanding agent subdivides the problem of picking up all 100 items into four problems of picking up all items within a quadrant of the world. These four goals are then passed on to the intermediate commanding agents, who further subdivide the problem into picking up all items within 8x8 subgrids. These goals are then assigned to the best-placed “worker” agents, using the Hungarian Algorithm for optimal assignment [27]. The 16 “worker” agents carry out these goals using a combination of the simple best-first planner and the reactive behaviour, as described above. To reduce the planning tree branching factor, the planner only considers horizontal and vertical movement, although agents can also move diagonally.

The 8x8 subgrids of the world can differ both in the number of items and the number of obstacles they contain. Agents which are overworked can therefore share work with each other via the informal network. They do this by sending goals to peer agents which are (believed to be) idle. The goal of picking up items in a specific subregion is passed to the idle peer which is closest to that subregion. In addition, a **favour size** is computed to be the area of the subregion plus ten times the distance of the peer agent from the subregion. Each experimental run had a maximum allowable favour size, ranging from 0 (no work sharing) to 1,000. We also compared work sharing with all 15 peers to work sharing only with the 3 peers under the same intermediate commander.

Table 1 shows the values of various experimental parameters considered. Ten experimental runs were executed for each combination of parameters, giving a total of 1,280 experimental runs, which took 53 hours to execute on a 2.2 GHz Pentium 4.

4. Results and analysis

All 1,280 experimental runs resulted in the agents successfully completing their mission, and picking up all 100 items. However, the time taken to do so varied enormously, from 4,500 to 140,500 simulated time units. The main factors influencing the time taken were the limit on planning steps and the communications delay (the latter mainly due to the two messages required in the hierarchy). Figure 4 shows the average time taken for each combination of these two factors.

Table 1: Experimental parameters

Time for agent to move one cell:	100 time units horizontally and vertically, 141 time units diagonally.
Sensors:	100% perfect sensors covering entire world.
Communications delay:	10; 100; 1,000; or 10,000 time units.
Planning limit:	50; 500; 5,000; or 50,000 steps.
Planning depth:	4; 6; 7; or 9 (computed from planning limit).
Thinking speed:	1 planning step per time unit.
Maximum favour sizes:	0; 10; 100; or 1,000.

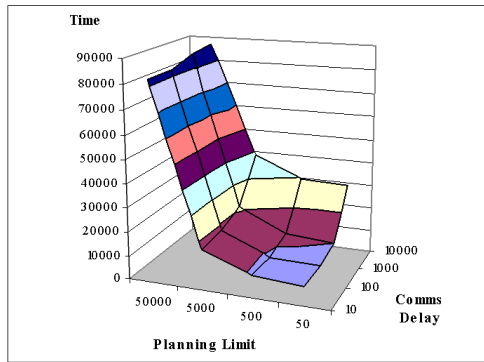


Figure 4: Time taken to complete mission by planning limit and communications delay

Some of the combinations of communications delay and planning limit did not have any statistically significant differences in the time taken, and these combinations were therefore grouped together.

Table 2: Mean and standard deviation of time taken to complete mission

		Planning limit			
		50,000	5,000	500	50
Comms delay	10,000	84,300 sd 18,300	38,500 sd 3,900	29,300 (sd 3,200)	
	1,000		20,900 sd 3,400	11,500 sd 2,400	10,000 sd 2,300
	100		17,500 sd 2,700	9,000 sd 2,500	7,800 sd 2,200
	10				

Table 2 shows the mean and standard deviation of the time taken for the resulting groups. The remaining differences between the cells in Table 2 were all statistically significant (at the $p < 0.0002$ level or better, by analysis of variance).

The remaining factor of interest was the “favour size” for workload sharing. This interacted with the planning limit, since at very high planning limits workload sharing was not effective: it led to excessive planning by peers, which took longer than simply letting agents carry out their goals themselves. Figure 5 illustrates the relationship.

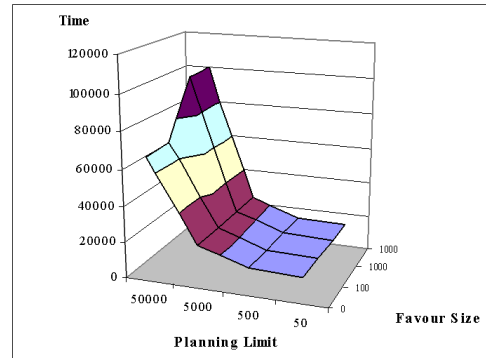


Figure 5: Time taken to complete mission by planning limit and “favour size”

To examine the impact of “favour size” in more detail, the extreme cases of 50,000-step planning limit and 10,000-step communications delay were ignored, and we looked at the difference between the time taken and the predicted times based on planning limit and communications delay (i.e. the values in Table 2). Table 3 shows the impact made by “favour size.” There was a statistically significant difference between the first two and last two columns ($p < 0.00002$). In other words, workload sharing decreased the time taken for mission completion by about 800 time units on average (10% of the best times). This was accomplished partly by reducing the result spread, as shown by the decrease in standard deviation. Figure 6 shows this more clearly.

Table 3: Impact of “favour size” on time taken to complete mission

Favour size	0	10	100	1,000
Mean time difference	+520	+310	-380	-450
Standard deviation	3,100	3,100	1,800	1,800
Maximum	+15,200	+11,800	+5,500	+7,600

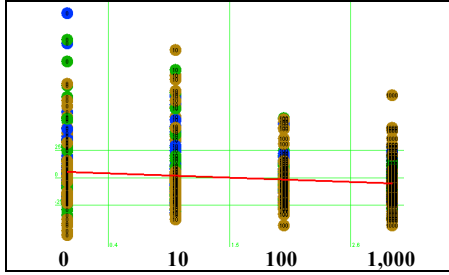


Figure 6: Distribution of time adjustments due to “favour size”

The 10% impact of workload sharing on the time for mission completion would probably be strengthened with a more sophisticated workload balancing algorithm. In particular, techniques from workload balancing in multiprocessor computers could be applied [28]. One consequence of the data in Table 3 is that as long as the “favour size” is large enough (i.e. at least 100), the exact value is not important. Indeed, the limit on “favour size” does not appear to be necessary.

When we compared work sharing with all 15 peers to work sharing only with the 3 peers under the same intermediate commander, there was no statistically significant effect. This is presumably because the experimental setup ensured that the best-placed agents for workload sharing were generally the ones under the same intermediate commander, anyway.

5. The impact of thinking speed

A disappointing feature of the data in Figure 4 and Table 2 was that planning did not seem to be particularly beneficial, and optimal performance occurred when the planning limit was only 50 steps. This was because the reactive behaviour was quite effective in achieving the mission of picking up items.

CAVALIER-NPA allows the “thinking speed” of agents to be adjusted. That is, the number of simulated time units taken by each planning step can be varied. In a second experiment, the “thinking speed” was increased to 100. The communications delay was fixed at 10 time units, and the “favour size” at 100. Fifty simulated runs each were executed for planning limits of 50 and 500, giving the results in Table 4.

Table 4: Time taken to complete mission with 100 times faster planning

Planning limit	50	500
Mean time taken	5,850	5,430
Standard deviation	750	930

For this second experiment, the increased planning speed made planning more effective than the reactive behaviour, and hence planning to an increased depth (with a limit of 500 planning steps) was worthwhile. The planning limit of 500 produced a statistically significant average decrease in time taken of 400 time units (compared to a planning limit of 50, $p < 0.02$).

In addition, the average time taken for both cases was about 2,500 time units faster than in the first experiment (statistically significant at $p < 10^{-25}$). The agents were therefore benefiting significantly from planning, even with relatively small planning limits.

6. Discussion

We have presented an agent-based simulation system for studying formal and informal organisational structures. In order to realistically describe organisational goals, it uses the Belief-Desire-Intention (BDI) architecture together with a combination of reactive behaviour and planning.

Our simulation experiment involved a 21-agent hierarchical team picking up 100 items in a randomly generated 32×32 grid-based world containing 12 cross-shaped obstacles (as illustrated in Figure 3). The performance of the team was given by the time taken to pick up all the items.

Performance was increased by not wasting too much time in planning. Consequently, in the first experiment, the best performance was obtained by planning for the minimum number of steps. However, when the thinking speed (the number of planning steps per time unit) was increased, planning became more effective than the default reactive behaviour, and planning for 500 steps became worthwhile. We plan to conduct laboratory studies of real military planning teams to determine the planning speed which best models teams of real people.

Lengthy communications delays (such as might result from excessively bureaucratic processes) decreased performance. Distributing work from the commanding agent to the 16 “worker” agents required two messages, and this explains the main impact of communications delay.

Workload sharing across the informal network improved performance when planning limits and communications delays were not extremely high. Since agents became idle when they had completed their assigned goals, workload sharing reduced the chance that one agent was still busy long after the others had finished.

The simulation system presented here will form the basis for more detailed and more accurate simulations of real military organisations which we plan to conduct

in the future. These simulations will require a greater range of agent goals and allowable actions, and may use the Metric-FF planner to simulate more complex planning by larger headquarters. In particular, we wish to explore the spectrum of organisational structures ranging from fully centralised through hierarchical to fully decentralised. We have previously explored this spectrum with more simplistic simulations [4, 8]. The experiment presented here has demonstrated the capability of CAVALIER-NPA for studying organisational factors using a more realistic model of military headquarters, by incorporating a planning capability within agents.

7. Acknowledgements

Valuable discussions with Richard Taylor (DSTO), James Harland (RMIT University), and James Chase (University of Tasmania) are gratefully acknowledged.

8. References

- [1] D. Alberts, J. Garstka, and F. Stein, *Network Centric Warfare: Developing and Leveraging Information Superiority*, (US) Department of Defense C4ISR Cooperative Research Program Publications Series, 1999.
- [2] D. Alberts and R. Hayes, *Power to the Edge*, (US) Department of Defense C4ISR Cooperative Research Program Publications Series, 2003.
- [3] S. Atkinson and J. Moffat, *The Agile Organization*, (US) Department of Defense C4ISR Cooperative Research Program Publications Series, 2005.
- [4] A.H. Dekker, C4ISR Architectures, Social Network Analysis and the FINC Methodology: An Experiment in Military Organisational Structure. DSTO report DSTO-GD-0313, 2002.
- [5] A.H. Dekker, Applying the FINC Methodology to the Land Environment. DSTO report DSTO-GD-0341, 2002.
- [6] A.H. Dekker, "Using Agent-Based Modelling to Study Organisational Performance and Cultural Differences," *Proceedings of the MODSIM 2003 International Congress on Modelling and Simulation*, Vol. 4, Townsville, Queensland, July 2003, pp. 1793–1798.
- [7] A.H. Dekker, "Centralisation and Decentralisation in Network Centric Warfare," *Journal of Battlefield Technology*, Vol. 6, No. 2, July 2003, pp. 23–28.
- [8] A.H. Dekker, "Centralisation vs Self-Synchronisation: An Agent-Based Investigation", to appear at the 11th *International Command and Control Research and Technology Symposium (ICCRTS)*, Cambridge, UK, 2006.
- [9] A.H. Dekker, "Visualisation of Social Networks using CAVALIER," *Proc. InVis.AU, the Australian Symposium on Information Visualisation (Conferences in Research and Practice in Information Technology, Vol. 9)*, Sydney, Australia, 3-4 December 2001.
- [10] A.H. Dekker and B. Colbert, "Network Robustness and Graph Topology," *Proc. ACSC04, the 27th Australasian Computer Science Conference (Conferences in Research and Practice in Information Technology, Vol. 26)*, Dunedin, New Zealand, 18-22 January 2004.
- [11] D.J. Watts, *Six Degrees: The Science of a Connected Age*, William Heinemann, 2003 (pp. 254–260).
- [12] L. Warne, D. Bopping, I. Ali, D. Hart, and C. Pascoe, "The Challenge of the Seamless Force: The Role of Informal Networks in Battlespace," *Proc. 10th International Command and Control Research and Technology Symposium (ICCRTS)*, McLean, Virginia, 13–16 June, 2005.
- [13] P.S. Dodds, D.J. Watts, and C.F. Sabel, "Information exchange and the robustness of organizational networks," *Proc. Nat. Acad. Sci.* Vol. 100, No. 21, 14 October 2003, p. 12516–12521.
- [14] A.S. Rao and M.P. Georgeff, "Modeling rational agents within a BDI-architecture." *Proc. 2nd International Conference on Principles of Knowledge Representation and Reasoning*, 1991, pp. 473–484.
- [15] M. Bratman, *Intentions, Plans, and Practical Reason*, Harvard University Press, 1987.
- [16] D. Dennett, *The Intentional Stance*, MIT Press, 1987.
- [17] P. Busetta, R. Rönquist, A. Hodgson, and A. Lucas. JACK Intelligent Agents – Components for Intelligent Agents in Java, AgentLink News Letter, Agent Oriented Software Pty. Ltd., Melbourne, January 1999.
- [18] S. Sardina, L. de Silva, and L. Padgham, "Hierarchical Planning in BDI Agent Programming Languages," *Proc. AAMAS-06*, 2006.
- [19] A. Newell and H.A. Simon, *Human Problem Solving*, Prentice-Hall, 1972.
- [20] G. Gigerenzer and R. Selten, *Bounded Rationality*, MIT Press, 2002.
- [21] R.E. Korf, "Real-Time Heuristic Search," *Artificial Intelligence*, Vol. 42, No. 2–3, 1990, pp. 189–211.
- [22] T. Dean, L.P. Kaelbling, J. Kirman, and A. Nicholson, "Planning with Deadlines in Stochastic Domains," *Proc. 11th National Conference on Artificial Intelligence*, Menlo Park, California, 1993, pp. 574–579.
- [23] J. Hoffmann and B. Nebel, "The FF Planning System: Fast Plan Generation through Heuristic Search," *Journal of Artificial Intelligence Research*, Vol. 14, 2001, pp 253–302.
- [24] J. Hoffmann, "The Metric-FF Planning System: Translating 'Ignoring Delete Lists' to Numeric State Variables," *Journal of Artificial Intelligence Research*, Vol. 20, 2003, pp 291–341.
- [25] A. Gerevini and D. Long, BNF Description of PDDL 3.0: www.cs.yale.edu/homes/dvm/papers/pddl-bnf.pdf, 7 Oct 2005.
- [26] A.V. Blum and M.L. Furst, "Fast Planning Through Planning Graph Analysis," *Artificial Intelligence*, Vol. 90, 1997, pp. 281–300.
- [27] N. Christofides, *Graph Theory: An Algorithmic Approach*, Academic Press, 1975.
- [28] A.S. Partridge and A.H. Dekker, "Speculative Parallelism in a Distributed Graph Reduction Machine," *Proc. 22nd Hawaii International Conference on Systems Sciences*, Vol. II (Software Track), Kailua-Kona, Hawaii, 3–6 January 1989, pp. 771–779.