Dynamic Evaluation of Coordination Mechanisms for Autonomous Agents

Rachel A. Bourne[1], Karen Shoop[1], and Nicholas R. Jennings[2]

[1] Department of Electronic Engineering
Queen Mary, University of London
London E1 4NS, UK
r.a.bourne,karen.shoop@elec.qmul.ac.uk

[2] Dept of Electronics and Computer Science
University of Southampton, Highfield
Southampton SO17 1BJ, UK
nrj@ecs.soton.ac.uk

Abstract. This paper presents a formal framework within which autonomous agents can dynamically select and apply different mechanisms to coordinate their interactions with one another. Agents use the task attributes and environmental conditions to evaluate which mechanism maximises their expected utility. Different agent types can be characterised by their willingness to cooperate and the relative value they place on short- vs long-term rewards. Our results demonstrate the viability of empowering agents in this way and show the quantitative benefits that agents accrue from being given the flexibility to control how they coordinate.

1 Introduction

Autonomous agents are increasingly being deployed in complex applications where they are required to act rationally in response to uncertain and unpredictable events. A key feature of this rationality is the ability of agents to coordinate their interactions *in ways that are suitable to their prevailing circumstances* [5]. Thus, in certain cases it may be appropriate to develop a detailed plan of coordination in which each of the participant's actions are rigidly prescribed and numerous synchronisation points are identified. At other times, however, it may be appropriate to adopt much looser coordination policies in which the agents work under the general assumption that their collaborators are progressing satisfactorily and that no explicit synchronisation is needed. What this illustrates is that there is no universally best method of coordinating. Given this fact, we believe agents should be free to adopt, *at run-time*, the method that they believe is best suited to their current situation. Thus, for example, in relatively stable environments social laws may be adopted as the most appropriate means of coordinating [10], whereas in highly dynamic situations one-off contracting models may be better suited [11], and in-between, mechanisms that involve the high-level interchange of participants' goals may be best [4].

To achieve this degree of flexibility, agents need to be equipped with a suite of *coordination mechanisms* (CMs) (with different properties and characteristics), be provided with a means of assessing the likely benefit of adopting the various mechanisms in the prevailing circumstances, and have the ability to select and then enact the best mechanism. Against this background, this paper develops and empirically evaluates a generic decision making model that agents can employ to coordinate flexibly. Specifically, we identify a number of potentially differentiating features that are common to a wide range of CMs, provide a decision-theoretic model for evaluating and selecting between competing mechanisms, and empirically evaluate the effectiveness of this model, for a number of CMs, in a suitably general agent scenario. This work builds upon the preliminary framework of [2], but makes the following advances to the general state of the art. Firstly, using a range of CMs, we show that agents can effectively evaluate and decide which to use, dependent on their prevailing conditions. Secondly, the evaluation functions associated with these CMs highlight the different types of uncertainty agents need to cope with and the different environmental parameters they need to monitor, in order to coordinate flexibly. Thirdly, we show that individual agent features such as their willingness to cooperate and the degree to which they discount their future rewards affects which CM they adopt.

The remainder of the paper is structured in the following manner. Section 2 outlines the key components of the reasoning model and introduces the exemplar coordination models we evaluate in this work. Section 3 describes the grid world scenario we use for our evaluation. Section 4 formalises the reasoning models. Section 5 describes the experimental results and analysis. Section 6 looks at related work in this area. Finally, in section 7 we draw our conclusions.

2 Coordination Mechanisms

Flexible coordination requires the agents to know both how to *apply* a given CM and how to *reason about* which mechanism to select. In the former case, an agent must have access to the necessary protocols for coordinating with other agents and/or the environment. In the latter case, an agent must be capable of evaluating and comparing the possible alternatives.

2.1 Protocols for Coordination

Coordination involves the interworking of a number of agents, subject to a set of rules. The specification of exactly what is possible in a particular coordination context is given by the coordination protocol [8]. Thus, such protocols indicate the parties (or roles) that are involved in the coordination activity, what communication flows can occur between these parties, and how the participants can legally respond to such communications. Here we refer to the instigator of the coordination as the *manager* and to the other agents that assist the manager as the *subordinates* (or subs).

For example, in the Contract Net protocol, the manager initiates a two-stage process whereby bids are requested and received, and then selected and subs are appointed. In a simpler mechanism, such as being commanded by a superior officer, a sub simply obeys the commands it receives. In all cases, however, the key point is that for each mechanism an agent supports, it must have the ability and the know-how to enact the protocol.

2.2 Evaluation of Mechanisms

A manager that is faced with a coordination task will have several potential CMs at its disposal. Each such mechanism requires a means of determining the expected value it will provide, which should be comparable with the others available. To this end, an e*valuation function* is needed. The value of a given CM may depend on many features including: the reward structure, the likely time of completion, and the likely availability of subordinates. Generally speaking, the more complex the coordination protocol and reward structure, the more complex the evaluation function. In particular, the more uncertainty that exists in the agent's ability to set up and enact a CM, the harder it is to evaluate its use accurately. Moreover, some of the parameters that are needed for evaluation are likely to vary from mechanism to mechanism. A final consideration is that the value of an agent's current situation may also depend on the state of the mechanism it has adopted. For example, an agent that has successfully recruited other agents may value its state more highly than one still engaged in the recruitment process. For all these reasons, evaluation functions need to be tailored to the specific CM they describe.

When subordinates are invited to assist in coordination, they too must assess the value of accepting. These valuations are typically less complex than those for the managers since the reward on offer and the completion time are generally declared, though in some cases a sub may also need to handle uncertainty. Subs also need to take into account whether accepting an offer would incur any additional cost, such as a penalty for dropping a commitment to another agent.

2.3 Sample Mechanisms

This section outlines the protocols and reward structures for the CMs considered in this work (their evaluation functions are left to section 4). Clearly this list is not exhaustive. Rather our aim is to incorporate specific exemplars that are typical of the broad classes of coordination techniques that have been proposed in the literature. Thus, the precise form of each mechanism is of secondary importance to its broad characteristics and performance profile. Moreover, additional mechanisms can be incorporated simply by providing appropriate characterisations and evaluation functions. Nevertheless we believe that the chosen mechanisms are sufficient for our main objective of demonstrating the efficacy of dynamically selecting CMs.

In this work, tasks are assumed to have several attributes: a minimum personnel requirement (*mpr*), a total requirement of agent effort (*effort*), and a *reward* that is paid to the managing agent when the task is accomplished. Thus a task that has *mpr* of 3 and *effort* of 6 may be realised by 3 agents each contributing 2 units, by 6 agents each contributing 1 unit, but not by 2 agents each contributing 3 units. It follows that tasks with a high *mpr* are likely to incur a delay before the necessary subs are recruited and all agents can start working on the task together. Here agents can only work on one task at a time. However, each agent has a default task ($mpr = 1$, $effort = 1$) that it puts on hold whenever it agrees to participate in a more complex one. The type of task and how many are generated can all be varied experimentally.

Asocial CM: This mechanism is used when a manager elects to perform a task alone; therefore there is no need to coordinate with any subs. The manager adopts the task, works on it and, ultimately, receives all the reward. This CM can only be used on tasks for which $mpr = 1$.

Social Law CM: A manager may elect to have the task performed by invoking a social law (SL) that has been agreed in advance by all the agents in the system[1]. For a task with $mpr = n$, the nearest $n-1$ other agents are commanded to work on the task with the manager. Because the social law extends to all agents, the subordinates cannot refuse to help. They simply assist until they are released from the task. The reward is then divided equally among all the participants. In our experimental setting, the location of subs was performed by a central coordinator allowing minimal set up delay and the prevention of multiple managers attempting to command subs at the same time. A truly distributed version is possible though would require a longer set up time.

Pot Luck CM: A manager that elects to use Pot Luck (PL) coordination, sets terms under which it is willing to pay subs on a piecemeal (step-by-step) basis. These terms are then offered to all agents in the direct vicinity (this is called "pot luck" since the manager makes no active effort to recruit subs in the hope that some are already present or wander by shortly). When the task is completed the manager receives the full reward. From the subordinate's point of view, it is occasionally offered "temporary" work for an indefinite period at a fixed rate; it either accepts and works on the task, or declines and ignores the offer. This CM is likely to be more successful when the environment is densely populated. But because the manager issues a blanket offer, it runs the risk of both over- and under-recruitment of subs. A sub can decommit from a PL task at any time at no penalty, keeping any reward it has already earned.

Contract Net CM: A manager that elects to use Contract Net (CN) coordination requests bids from other agents that submit their terms according to their current circumstances. The manager selects from among the bids received and sends out firm offers. An agent receiving an offer either accepts and works on the task, eventually receiving a reward based on its bid, or declines. The manager may thus fail to recruit sufficient subs in which case it repeats the

[1] The process of agreeing the social law is not considered in this work. In particular, this means that the associated costs of acheiving consensus are not factored into the cost of this mechanism.

request-bid-offer cycle. When the task is accomplished, the manager receives the whole reward but must pay off the subs with the agreed amounts. Again, under this CM a manager may under recruit if some agent declines its offer. Once a sub has accepted a task under this CM, it may later decommit though it will have to pay a penalty for doing so. If a manager abandons a task under this CM, it must pay off any subs it has recruited, according to their terms.

3 Grid World Scenario

The scenario involves a number of autonomous agents occupying a grid world (see [2] for more details). Tasks are generated randomly according to the experimental conditions and are found by the agents who must decide whether or not to take them on. Each agent always has a specific (default) task that may be carried out alone. Tasks are located at squares in the grid and each has an associated *mpr*, *effort* and *reward*. One unit of effort is equivalent to one agent working on the task at the square for one time step (provided sufficient agents work on it in total). When accepting a task, an agent must decide how to tackle it; if the *mpr* is greater than one, it must recruit other agents to assist it, according to the various CMs at its disposal. To simplify the evaluation functions, tasks persist until they have been achieved. A more realistic setting with deadlines on tasks will require that the evaluation functions be extended to take into account the possibility of a task not being completed in time.

The agents themselves have various features that can be parameterised to simulate different behavioural types. Each agent has a *willingness to cooperate* (wtc) factor which it uses when bidding for and evaluating tasks; when this factor is low ($wtc < 1$) the agents are *greedy* and when high ($wtc > 1$) they are selfless; in-between ($wtc = 1$) agents are neutral. Agents also use a *discount factor* (see below) which reflects their capacity to value short- vs long-term rewards.

The agents move synchronously around the grid world, being capable of five actions: up, down, left, right and work (remain). To simplify the analysis below, the grid is formed into a torus so that an agent moving up from the top of the grid arrives at the bottom in the corresponding column; similarly for the left and right edges. This enables us to use a relatively simple probabilistic model of square occupancy by agents.

This scenario offers a broad range of environments in which the effectiveness of reasoning about the various CMs can be assessed. While this scenario is obviously idealised (in the tradition of the tileworld scenario for single agent reasoning), we believe it incorporates the key facets that an agent would face when making coordination decisions in realistic environments. In particular, the ability to systematically control variability in the scenario is needed to evaluate our claims about the efficacy of flexible coordination and the dynamic selection of CMs according to prevailing circumstances.

4 Agent Decision-Making

Since the agents always have a specific task to achieve with $mpr = 1$ and $effort = 1$ they always have a non-zero expectation of their future reward. However, they can increase this reward by initiating or participating in collaborative tasks (CTs) with other agents. In deciding on its collaborations, each agent aims to maximise its future rewards by adopting the best mechanism under the circumstances. Thus at all instants in time, an agent will have a current goal being undertaken using a particular CM with the agent taking a particular role.

Agents may find tasks in the environment or be offered them by other agents. In either case, however, an agent needs a means of evaluating new tasks under each of the CMs. It must also be able to compare these values with its current task so as to determine when to accept a new offer. Thus each CM has an associated evaluation function which it can use to approximate the value of potential new tasks, as well as any current task being undertaken using it. These evaluation functions have some common features, but they also differ between the CMs in that they may use different aspects of the (perceived) environment.

An important consideration that needs to be incorporated into the evaluation of new offers is that of liability. If, for example, an agent has agreed to participate under the Contract Net CM, it may incur a penalty for decommitting – any new offer will need to be greater than the current task *and* the decommitment penalty. Conversely, a manager must manage its liabilities so that, if a task becomes unprofitable, it can "cut its losses" and abandon it (since the situation has already deteriorated from the time it was adopted).

Our agents are myopic in that they can only see as far as the next reward, however since tasks may arrive at different times in the future we discount all rewards back to the present using a discount factor, $0 < \delta < 1$. When $\delta \approx 1$, the difference between long- and short-term rewards is not great; however, when $\delta << 1$, short-term rewards appear more attractive [7].

For the Social Law, Pot Luck and Contract Net CMs both managers and subs mask their valuations according to their own *wtc* factor. This makes coordinating over collaborative tasks a more attractive proposition when *wtc* is high and less attractive when it is low. Evaluation of the Asocial CM is unaffected by *wtc*.

The following subsections give details of the evaluation functions used for each of the aforementioned CMs. These functions are designed to illustrate the sorts of functions that can be used to evaluate CMs. They are necessarily *approximate* valuations, not least because there is a great deal of uncertainty and imperfect information in the scenario. We do not claim that they are the only ones possible, nor that they are optimal, but rather that they are reasonable and demonstrate that CMs can be evaluated in dynamic and unpredictable environments using suitably parameterised functions.

4.1 Asocial CM

This CM simply involves the agent moving towards its task and working there for the necessary number of time steps before receiving the reward. To evaluate a task with $mpr = 1$, $effort = e$ and $reward = R$, an agent discounts the reward it expects by the time until it will receive it. If the distance to the task is l, the expected value of the CM, V_A, is given by:

$$V_A = R\delta^{l+e}$$

If the agent is already performing the task ($l = 0$), the reward is discounted by the amount of effort remaining.

Evaluation of the Asocial CM does not require any additional environmental information.

4.2 Social Law CM

When an agent adopts the Social Law CM, the appropriate number of the nearest agents are commanded to come to the manager's assistance. When the final agent arrives, all agents work on the task until completion and the reward is equally divided among them.

To evaluate a task with $mpr = m$, $effort = e$ and $reward = R$ under this CM, the agent must estimate the time it will take for the furthest agent to arrive at the square. This can be calculated using the average occupancy of each square[2]. The manager adds up the occupancies of the nearest squares to it until it obtains $m - 1$, and uses the furthest square required to estimate the time till all agents will be present, say l. Since this CM can only be invoked on one task at a time, the manager may also take some non-zero set up cost into account, say s. Given these estimates, the expected value of this CM, V_{SL}, is given by:

$$V_{SL} = \frac{R\delta^{(l+s+\frac{e}{m})}}{m}$$

Evaluation of the Social Law CM requires knowledge of the distribution and density of other agents, here represented by average occupancy, as well as any set up costs which using the social law may incur.

4.3 Pot Luck CM

When an agent adopts the Pot Luck CM, it makes no effort to recruit other agents unless they happen to enter the square where the CT is situated. In such cases, the manager offers the potential subordinates employment for an indefinite period at a fixed rate. The terms it offers reflect the agent's perceptions about the *wtc* and discount factor of other agents; it sets a rate that, it believes, is sufficient to attract passers-by until the task has been achieved. Any agents that accept this offer and remain at the square, committed to this task, receive the agreed rate at each step. These offers of piecemeal employment remain until the task is completed or the managing agent abandons it. When the task is complete, the managing agent receives the reward.

[2] That is, $\frac{numberOfOtherAgents}{numberOfSquares}$.

To evaluate a task with $mpr = m$, $effort = e$ and $reward = R$, the manager assumes that subs will wander by at intervals determined by the remaining average occupancy: if n agents are at large, the interval is given by $numberOfSquares/n$. The manager computes a rate, r, to offer to the subs that it would find acceptable itself in similar circumstances (see below). Based on these assumptions and the task *effort*, the agent computes the expected completion time of the task, *ect* and the future value of the amount it will have to pay out to the subs, p. Then the expected value of applying this CM, V_{PL}, is given by:

$$V_{PL} = (R - p)\delta^{ect}$$

When this CM is in use, the manager uses a similar technique to evaluate the task in progress, taking into account any subs already helping. Clearly, if agents are already present, the value increases considerably. Note that, with the Pot Luck CM, it is possible that the manager recruits more agents than the *mpr*, meaning that the task will be achieved more quickly.

Subordinates evaluating a Pot Luck offer discount the rate offered, r, indefinitely into the future:

$$V_{PL} = \frac{r}{1 - \delta}$$

Thus although the rate may be low compared with the reward for their default task, the fact that they will receive it regularly starting from the next time step makes the offer relatively attractive.

Evaluation of the Pot Luck CM again requires knowledge of the distribution and density of other agents, though this knowledge is used in a different way. Managers need to be able to offer an appropriate rate to subs; in general, the manager will need to consider the other agents' *wtc* and discount factors, though here it assumes them to be the same as its own.

4.4 Contract Net CM

Under this mechanism, a manager broadcasts a request for bids to the other agents. On receiving their replies, it computes the best set of bids and sends out firm offers of employment to the selected agents. These agents may either accept the offer or decline, possibly causing the managing agent to issue more requests. When the task has been completed, the manager receives the reward and pays its recruits the agreed amounts. Since the completion time of the task is unknown, subordinates bid an amount which reflects their current requirements, and when they are paid off this is factored up by their discount rate and their time committed. This means that the manager has an increasing liability to the other agents so long as the task remains unfinished—any extension to the *ect* may therefore greatly affect the value of using this CM.

To evaluate this mechanism, the agent estimates the average distance away of the furthest agent (as described above) and adds to this the communication costs of this CM (3 time steps until subs will be committed) and the duration based on $effort/mpr$. This gives the *ect*. The manager also estimates the likely bid requirements of the subs based on its perception of their *wtc*, discount factors and their specific task rewards. Thus if it anticipates completion in *ect* time steps, with i subs committed 3 time steps hence each bidding r_i, the value of using the Contract Net CM is given by:

$$V_{CN} = \delta^{ect}(R - \sum_i \frac{r_i}{\delta_i^{ect-3}})$$

The reward structure of this CM is such that the subs only receive payment when the manager either completes or abandons the task. At this time they receive their bid factored up by the amount of time they have been committed (agent i receives r_i/δ_i^t after being committed for t time steps). Thus, however long they are committed to the task, the reward they receive, discounted back to when they started, remains the same. In this way the manager can determine its current liability at any stage of the coordination, e.g., when an offer is rejected.

An agent bidding under the Contract Net CM, factors its current reward by its *wtc* and projects it two time steps into the future, since this is when an offer will arrive. It submits this amount, r_i, plus its required discount factor, δ_i, and the time it expects to arrive at the task. The manager selects bids based on how they affect its expected reward. That is, it looks at the surplus reward when each agent has been paid, discounted by how long until it arrives. This simplifies the otherwise combinatorial problem of selecting the best i bids.

Evaluating a Contract Net CM that is underway, involves computing the current liability to agents committed, projecting this forward till the *ect* and discounting this value and the reward back to the present. As time goes by, the value of participating in this CM increases for subs (since the manager will be paying more and more at each time step) and so a sub becomes more committed the longer it has been involved.

If a sub decommits, even through being coopted under Social Law, it must pay the manager a decommitment penalty, which is intended to compensate the manager for the time wasted. Although not implemented here, a more sophisticated evaluation function would require an estimate of the likelihood of subs decommitting. Additionally, more flexible decommitment penalties could be set dynamically to reflect the prevailing circumstances. This aspect of our work is being investigated concurrently and is reported in [6].

Evaluation of the Contract Net CM requires similar knowledge to the Pot Luck CM.

5 Experiments, Results and Analysis

Our first set of experiments assessed the accuracy of each CM's evaluation function; the circumstances under which each is selected; and to what extent this additional flexibility benefits the managing agent. To this end, we conducted a series of simulations for a 10 x 10 grid, containing just one CT with $reward = 15$ (default tasks have $reward = 1$). Furthermore, all agents have $wtc = 1$ and $discount = 0.9$. We varied the number of agents (from 5 to 25), the *mpr* (from

1 to 8) and the *effort* (from 10 to 30). Since this represents a very large sample space, we report a selection of representative results.

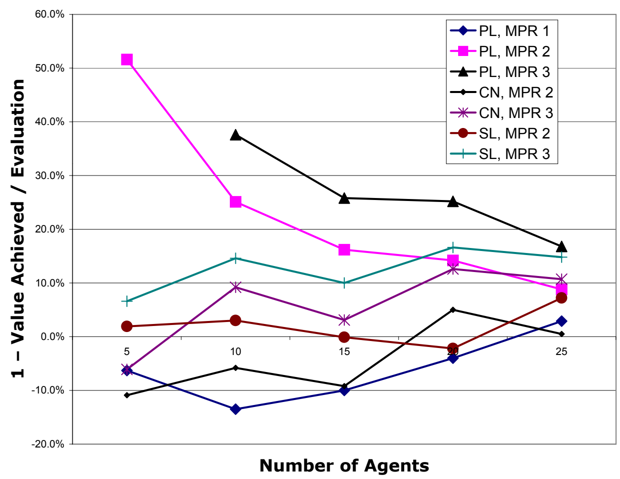

Fig. 1. Performance of CM Evaluation Functions

Figure 1 plots the difference between the value managers actually achieved and their evaluations, for each CM, $mpr = 1, 2, 3$ and $effort = 10$. (SL and CN do not apply when $mpr = 1$; the value of PL when $mpr = 3$ and there are only 5 agents is negative and so the manager sticks with its default task.) PL is the least accurate, overestimating the true value by 50% in the worst case. SL and CN are more accurate reflecting their lower levels of uncertainty. The graph indicates that the accuracy of all CMs tends to improve as the agent population increases. Bearing in mind that, for a discount factor of 0.9, a one step error in *ect* leads to about a 10% error in evaluation, these results show that the evaluation functions work acceptably under a variety of conditions. This demonstrates that it is feasible to evaluate CMs based on estimates of environmental parameters.

Figure 2 shows the type of CM selected by the managers under different agent densities and task profiles. It is clear that, at least from the manager's point of view, for high agent densities and low *mpr*, PL is the most preferred. This is due to PL's low communication costs (no active recruitment), low payout rate to subordinates and the ability to over-recruit. However, under more general conditions, CN is the most preferred: the communication cost (set up delay) and the relatively high rates paid to subs are compensated for by the reduced time till all agents arrive. For tasks with extremely stringent requirements, i.e., high *mpr* or *effort* in low density populations, the speed and certainty of SL leads to it being selected over CN. In fact, the choice between CN and SL depends mainly on the reward assigned to the task—for lower task rewards, SL is preferable to

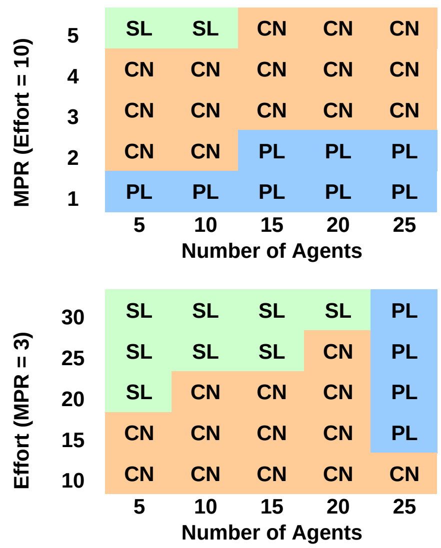

Fig. 2. Preferred CM by Task/Environment

CN. In the extreme, even SL is unprofitable and managers opt to perform their default tasks instead.

Figure 3 shows the total reward achieved by managers using each CM in isolation, and when the managers have the capability to dynamically choose which CM to use. The number of agents is fixed at 15, and the task profiles have fixed $reward = 15$, $effort = 10$ and mpr ranging from 1 to 8. As the mpr increases, the surplus reward available to managers naturally decreases, but the graph shows that agents who can coordinate flexibly maintain high levels of reward (in fact the line for ALL roughly traces the maximum over all CMs).

When taken together, these results clearly show that providing alternative mechanisms for coordination is beneficial to agents that are required to interoperate under changeable environmental or task conditions.

Our second set of experiments examined the overall effect on the system when the agents displayed different characteristics in terms of their wtc and $discount$ factors. The general hypothesis here is that when agents are less greedy, more collaborative tasks will be achieved and that an agent's perspective on future rewards may well affect the type of CM it chooses. To test this, we conducted a series of experiments varying wtc from 0.25 to 3 and $discount$ from 0.5 to 0.95 (the number of agents was fixed at 15, and the tasks have fixed $reward = 15$, $effort = 10$ and $mpr = 2$).

Figure 4 shows the total number of CTs achieved for each discount factor as wtc increases. When the discount factor was 0.8 or less, the managers always selected SL, because this minimises the ect. However, when the agent's wtc is low, it devalues its own reward for CTs to the extent that the default task is

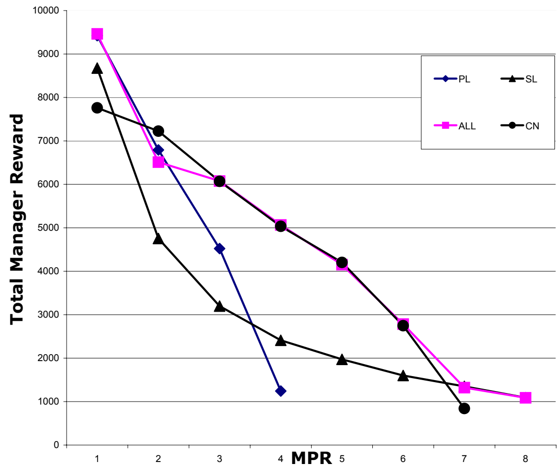

Fig. 3. Total Reward Achieved by Managers

often chosen instead. The choices for discounts of 0.9 and 0.95 were identical, with PL being chosen for $wtc \leq 1$ (agents neutral or greedy) and CN being chosen otherwise (agents selfless). This indicates that, if potential subordinates are likely to require relatively large rewards and the manager can afford to wait, it will choose to do so, paying out less to subordinates. These results confirm that the relative value of using different CMs is indeed affected by the both the individual characteristics of an agent and its beliefs about the environment in which it operates.

6 Related Work

The majority of previous work on multi-agent system coordination assumes it is a design time problem (e.g., [10, 11, 4]). However [5] has argued that agents need the flexibility to coordinate at different levels of abstraction, depending upon their particular needs at a given moment in time. To date, however, this work has not developed mechanisms for explicitly reasoning about which level to coordinate at in a given situation. Such flexibility was also built into cooperative problem solving agents by [9]. Here, agents could choose to cooperate according to various conventions which dictated how they should behave in a particular team context. These conventions varied in terms of the time they took to establish and the communication overhead they imposed. However, again, there was no reasoning mechanism for determining which convention was appropriate for a given situation. Boutilier [3] presents a decision making framework, based on multi-agent Markov decision processes, that does reason about the state of a coordination mechanism. However, his work is concerned with optimal reasoning within the context of a given coordination mechanism, rather than actually reasoning about which mechanism to employ in a particular situation.

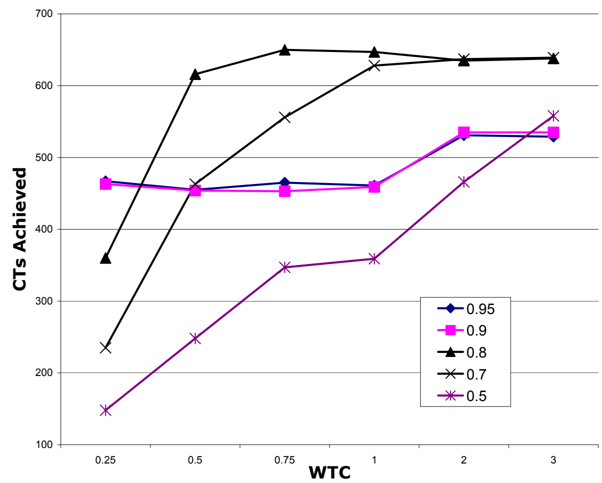

Fig. 4. CTs Achieved by WTC and Discount Factor

[1] present a software engineering framework that enables agents to vary their CMs according to their prevailing circumstances. They also identify criteria for determining when particular mechanisms are appropriate. However, the decision procedures for actually trading-off these criteria are not well developed. Finally, [2] provide a framework in which CMs are characterised by set up costs and probability of success and can be evaluated accordingly; however, their agents use a contract-net style protocol for all their interactions.

7 Conclusions

This paper presented a framework in which agents can evaluate and apply differing CMs and has demonstrated that agents can benefit from such flexibility. It has shown that CMs can be practically evaluated using appropriate environmental parameters despite the uncertainty agents face. However, these experiments use static environmental conditions and the agents involved use assumptions about the environment. To overcome these restrictions, in our future work we will allow agents to monitor and learn the relevant environmental parameters so that they can react to dynamic environments. Agents will then be in a position to adapt their own attributes (*wtc* and *discount*) to better suit their circumstances. Given agents that learn and adapt to their environment, it will also be important to assess whether alternative evaluation functions or heuristics impact on agent performance.

References

1. K. Barber, D. Han, and T. Liu. Coordinating distributed decision making using reusable interaction specifications. In *Proc 3rd Pacific Rim Wsp. on Multi-Agents*,

pages 1–15, Australia, 2000.

2. R. A. Bourne, C. B. Excelente-Toledo, and N. R. Jennings. Run-time selection of coordination mechanisms in multi-agent systems. In *Proc. 14th European Conf. on AI*, pages 348–352, Berlin, Germany, 2000.
3. C. Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 478–485, Menlo Park, CA, 1999.
4. E. Durfee and V. Lesser. Partial global planning. *IEEE Trans. on Systems, Man, and Cybernetics*, 21(5):1167–1183, 1991.
5. E. H. Durfee. Practically coordinating. *AI Magazine*, 20(1):99–116, 1999.
6. C. B. Excelente-Toledo, R. A. Bourne, and N. R. Jennings. Reasoning about commitments and penalties for coordination between autonomous agents. In *Proceedings of 5th International Conference on Autonomous Agents (Agents 2001)*, pages 131–138, 2001.
7. J. Filar. *Competitive Markov decision processes.* Springer, New York, 1997.
8. A. Haddadi. *Communication and cooperation in agent systems.* Springer Verlag, 1996.
9. N. R. Jennings. Commitments and conventions. *The Knowledge Engineering Review*, 8(3):223–250, 1993.
10. Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proc. of the 10th Nat. Conf. on AI*, San Diego, CA, 1992.
11. R. G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Trans. on Systems, Man, and Cybernetics*, 11(1):61–70, 1980.