UNIVERSITY OF SOUTHAMPTON

THE DYNAMIC SELECTION OF COORDINATION MECHANISMS IN MULTIAGENT SYSTEMS

by

Cora Beatriz EXCELENTE TOLEDO

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Faculty of Engineering and Applied Science

Department of Electronics and Computer Science

February, 2003

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

THE DYNAMIC SELECTION OF COORDINATION MECHANISMS IN MULTIAGENT SYSTEMS

by Cora Beatriz EXCELENTE TOLEDO

This thesis presents and evaluates a decision making framework that enables autonomous agents to dynamically select the mechanism they employ in order to coordinate their inter-related activities. Adopting this framework means coordination mechanisms move from the realm of something that is imposed upon the system at design time, to something that the agents select at run-time in order to fit their prevailing circumstances and their current coordination needs. Using this framework, agents make informed choices about when and how to coordinate, when to respond to requests for coordination and when it is profitable to drop contracts in order to exploit better opportunities. The framework's efficacy is investigated in a range of coordination situations; such as scenarios in which agents have varying dispositions to work cooperatively, in which agents make decisions based on alternative assumptions about their environment, and in which the task of predicting other agents' behaviour varies in difficulty. In all of these cases, a systematic empirical analysis is undertaken to evaluate the effect of such situations on the agent's decision making mechanisms and their effectiveness.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to recognise and express my gratitude to everyone who gave their time, patience, knowledge, friendship and enthusiasm to help me reach this point of my PhD. I have greatly enriched my academic and personal life during this process.

To the Mexico's National Council of Science and Technology, CONACyT (Ref. 56114/134705) and the National Laboratory of Advanced Computer Science, LANIA whose financial support allowed me to carry out my research work.

I am deeply indebted to Prof. Nicholas Jennings for being such a good supervisor through all this experience. His indisputable knowledge in the field, his beyond duty interest in the research and his always valuable suggestions made me realise there is nothing that can't be improved; my gratitude.

I would like also to thank Rachel Bourne for her contribution to the formalisation of the model presented in this work.

My endless gratitude to Dr. Cristina Loyo and Dr. Christian Lemaitre for their encouraging inspiration and undeserved confidence in me. Giving me the opportunity to be part of the LANIA was a decisive point in my life... a very pleasant and formative one.

I am oblige to Professor David Barron whose kindness and patience where never any less to what I could expect from a truly English gentleman.

I owe a big thank you to my friends in Mexico who made me feel their warmth and narrowed the gap imposed by the physical distance. The same infinite thanks go to the friends who opened their hearts to me in Britain in spite of our cultural differences. You are all an amazing bunch!

Last but not least, I would like to tell Alejandro and my family how much I love them; unfortunately there are not enough words in any language to express my love; however I would like them to join me in the joy of completing this little but satisfactory achievement, and like me, put a smile on their faces.

Muchas gracias a todos.

A Alejandro

A mis Padres y hermanos

A mis amigos de toda la vida

Chapter 1

Introduction

An increasing number of computer systems are being designed in terms of autonomous agents (Parunak, 1999; Jennings, 2001; Parunak, 2000; Luck *et al.*, 2003; Jennings, 1998). This is because agent-based approaches have been shown to be well suited to analysing, designing and building complex and distributed systems (Jennings, 2001). More specifically, the features of agents that are most relevant in this context are: *autonomy*: an agent can act without human intervention and it has control over its own actions; *reactivity*: an agent is able to perceive its environment and respond in a timely manner to changes that occur on it; and *proactivity*: an agent takes the initiative in order to satisfy the objectives for which it was designed (Wooldridge & Jennings, 1995; Wooldridge, 2001).

While these features of agenthood concentrate on the properties of individual components that can act flexibly in pursuit of their objectives, much of the power of agent-based computing stems from the fact that agents are also social (Wooldridge & Jennings, 1995; Wooldridge, 2001). This *social ability* means that agents interact with one another to satisfy their own objectives or the objectives of the wider community. Here, systems in which such interactions take place are termed *multiagent systems* (MAS). Examples of such systems include: the design of an artifact in which the final outcome is the result of integrating the efforts of several designers that produce constituent parts of the whole (Durfee *et al.*, 1989; Shen & Barthes, 1995; McGuire *et al.*, 1993); auction houses in which the price of a good is obtained as a result of several agents competing against each other (Wurman *et al.*, 2001; Sandholm, 1999; He *et al.*, 2003), and a mechanism for synchronising access to a common source of information in which several agents negotiate to have the right to access information (Rosenschein & Zlotkin, 1994;

Huhns & Stephens, 1999). In more detail, a MAS system is characterised as a system in which there are a number of autonomous agents that inhabit a common environment and that *interact* with one another for a variety of reasons (see for example Figure 1.1 where various types of interactions are illustrated [1]).

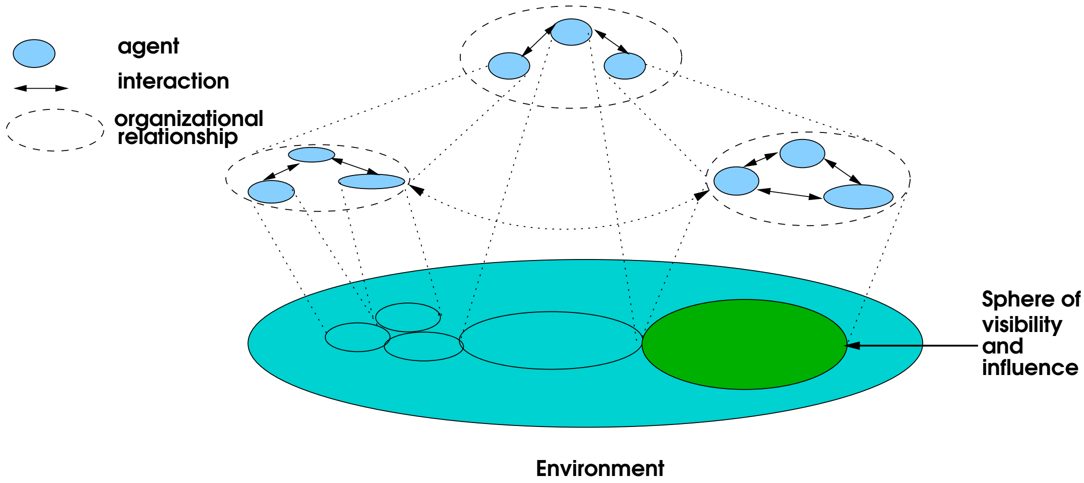


Figure 1.1: Canonical view of a multiagent system (from (Jennings, 2001)).

In this work, the generic term that will be used to cover all types of social interaction is *coordination* (Bond & Gasser, 1988a; Moulin & Chaib-Draa, 1996; Durfee *et al.*, 1989). Thus, in the design of the artifact, for example, it is necessary to coordinate the independent designs into the global one and, moreover, it is desirable that these coordinated activities accomplish the final task with the minimum time and resources. In the auction example, coordination is needed to indicate how the offers and counteroffers between the agents can take place in order to have a final price that is agreed by both parties. Finally, in the synchronisation example, coordination is required to establish the order and time at which each agent should consult and access the central source of information.

In all of these cases, successful coordination results in the overall system performing in a coherent manner. If, however, the coordination is not effective then the overall outcome may be incoherent behaviour (i.e. the system does not work as a unit and the results generated or the communication of those results by the agents are not well synchronised (Huhns & Stephens, 1999)). For example, in a MAS system for dealing with scarce resources, miscoordination could cause resources to be wasted, activities to be needlessly duplicated, or no overall solution to be reached at all. From these and other examples, it can be seen that coor-

[1]Note that agents are grouped in an organisational relationship (Section 2.1.4).

dination is usually needed for one of the following reasons (Durfee *et al.*, 1989; Jennings, 1998; Lesser, 1999; Bond & Gasser, 1988a):

- *Agents pursue global or interrelated individual goals.* Agents might be members of a group with global objectives (cooperative agents) or they might pursue their own local objectives (competitive agents) in a community that is composed of several other agents. In the former case, the agent's interest is to work with others in order to satisfy these global requirements or constraints, to facilitate the work of others or to avoid harmful interactions. For example, in systems in which agents are specialists or experts in particular fields (the tasks are functionally distributed) or they are located in specific regions or areas (the tasks are spatially distributed) they might all contribute with specific parts to the global solution (Durfee *et al.*, 1989; Bond & Gasser, 1988a; Jennings, 1998). In the latter case, the agents' interests are to achieve their own objectives, regardless of the interests of others, but some form of coordination is needed to resolve conflicting situations. The classical example of this type of encounter is when agents play the role of buyer and seller in a market and they compete against each other to obtain higher benefits (Wurman *et al.*, 1998).

- *Agents have limited viewpoints.* Agent systems do not have a centralised control component and there is no a global view of the problem (as shown in Figure 1.1, all the agents have a limited sphere of visibility and influence). Given such conditions, agents need to share and coordinate their partial views in order to obtain a global solution. There are a number of applications in which the knowledge and expertise might be distributed between several agents. For example, when it is necessary to interpret or recognise data in which the sources are distributed in different locations (Lesser & Corkill, 1983); when it is required to control a set of robots which work together (Durfee *et al.*, 1989), or when it is needed to integrate the information gathered by several agents from the Internet (Decker *et al.*, 1997). In all of these cases, agents need to work together to produce a consistent and coherent global solution.

- *Agents share the environment and resources.* In environments with limited or scarce resources, agents need to coordinate in terms of the time and/or the amount of resource they use to either avoid conflicts or to facilitate the achievement of common activities (Bond & Gasser, 1988b). For example,

> in situations in which agents have distributed computational resources (fast computation, memory, specialised software, etc.), they need to agree about the order in which the resources are employed to solve individual or global problems (Durfee *et al.*, 1989).

Against this background, it can be seen that coordination is a key issue in MAS research because it provides the mechanisms to ensure that groups of agents act in a coherent manner. However, coordination, like several other MAS concepts, has been used to describe a variety of phenomena which means that it does not have a generally agreed upon definition and its scope varies considerably. For this reason, the first step is to clarify the concepts involved as they are used in this thesis. Regardless of the dimension, level of abstraction, perspective and discipline, most researchers generally agree that: *coordination is the act of managing interdependencies between activities* (Decker, 1995; Bond & Gasser, 1988a; Moulin & Chaib-Draa, 1996; Wooldridge, 2001).

From the above, it is clear that coordination is needed for a variety of reasons and it encompasses a wide variety of behaviours. Consequently, many methods and techniques have been proposed for coordinating the behaviour of multiple agent systems. These include appealing to a higher authority agent in an organisational structure (Fox, 1981; Malone, 1987; Carley & Gasser, 1999), instituting social laws that avoid harmful interactions (Briggs & Cook, 1995; Shoham & Tennenholtz, 1992), using contracting protocols or bids exchanged in a market place to allocate tasks (Smith & Davis, 1981; Malone, 1987; Wellman, 1993; Huberman & Hogg, 1995), iteratively exchanging tentative plans until all constraints are satisfied (Durfee & Lesser, 1991) and negotiating to find agreements (Rosenschein & Zlotkin, 1994; Sandholm & Lesser, 1995; Huhns & Stephens, 1999).

For the purposes of this work, however, all the methods and techniques for achieving coordination will be grouped under the generic term of *"coordination mechanisms"* (CMs). Thus, for example, contracting, planning, negotiating, exploiting organisational structures are all CMs. In more detail, a *CM determines the course of action that agents can take in order to manage their interactions to achieve their goals.* In other words, a *CM defines how agents should interact, at what time and with which resources to accomplish their goals.*

All of these *coordination mechanisms* have different properties and characteristics and are suited to different types of tasks and environments. They vary in the degree to which coordination is prescribed at design time, the amount of time and effort they require to set up a given coordination episode at run-time, and the degree to which they are likely to be successful and produce coordinated behaviour in a given situation. For example, a CM that involves a significant amount of communication to reach agreements is most appropriate when the agents that inhabit the environment are not only accessible but also few in number. In contrast, a CM that exploits an organisational structure is better suited to situations in which the number of agents is very large and conformance with the structure can be guaranteed in advanced. Generally speaking, however, these dimensions act as forces in opposing directions; coordination mechanisms that are highly likely to succeed typically have high set up and maintenance costs, whereas mechanisms that have lower set up costs are also more likely to fail. Moreover, a coordination mechanism that works well in a reasonably static environment will often perform poorly in a dynamic and fast changing one. In short, there is no universally best coordination mechanism (Galbraith, 1973).

Given this situation, this thesis argues that it is important for the agents to have a variety of coordination mechanisms, with varying properties, at their disposal so that they can then select a mechanism that is appropriate to the coordination episode at hand. Thus, for particularly important tasks, the agents may choose to adopt a coordination mechanism that is highly likely to succeed, but which will invariably have a correspondingly large set up cost. Whereas for less important tasks, a mechanism that is less likely to succeed, but which has lower set up costs, may be more appropriate. Similarly, when it is difficult or expensive to set up a coordination activity (e.g. because there are not many agents available) the agent is likely to pick a mechanism that is highly likely to succeed (even though it will have a high set up cost). In contrast, if coordination can be set up easily or cheaply then it might be more important to choose a mechanism that promises to achieve the task in the quickest possible way.

To date, however, the choice of which coordination mechanism to use in a given situation is something that the designer typically imposes upon the system at design time. Thus, for example, in a given application a particular social law will be used or it will be decided that all coordination activities will be handled by a particular negotiation protocol. This means that in many cases the coordination mechanism that is employed is not well suited to the agents' prevailing circumstances. This inflexibility means that the performance of both individual agents and the overall system may be compromised.

To achieve the desired degree of flexibility in coordination two key decision making components need to be present in the agent (Jennings, 1993):

(i) the ability to select a means of coordinating that is appropriate to the prevailing situation; and

(ii) the ability to assess (and re-assess) an agent's commitment to the on-going coordination activity.

Regarding the first point, this thesis claims that it is inappropriate to impose a particular coordination mechanism upon the system at design time because there is no scope for changing or modifying the mechanism to ensure there is a good fit with the prevailing circumstances (Bourne *et al.*, 2000; Excelente-Toledo & Jennings, 2003a). To circumvent this problem, a decision making framework is presented that enables agents to dynamically select the coordination mechanism that is most appropriate to their circumstances. Thus, this work is *not* concerned with developing actual coordination mechanisms that exhibit the varying properties discussed above, nor with classifying existing coordination mechanisms along such dimensions. Rather, the coordination mechanisms are viewed in the abstract; representing them in a quantitative way. In particular, CMs are modelled in terms of the cost involved in the coordination and the benefits of adopting a particular mechanism (as advocated by (Lesser, 1998, 1999)) (see Section 1.1.2 for a number of illustrative examples). The framework is then concerned with deciding whether coordination should be attempted in a given context (given the agent has a particular set of mechanisms at its disposal) and, if so, which of the available mechanisms is the most appropriate to employ in the prevailing situation. Once a particular mechanism has been chosen for a particular coordination episode, the agents involved are expected to adhere to the rules and procedures specified in the mechanism itself. Thus the rules indicate how the coordination should actually proceed.

Moreover, if the system is open, dynamic and heterogeneous (as is the norm for Web applications (Hendler, 2003), e-commerce systems (He *et al.*, 2003) and grid computing application (deRoure *et al.*, 2003)) agents face even greater difficulty when taking coordinating decisions. This is because in such environments it is impossible to enumerate in advance the wide variety of contexts in which coordination is likely to be needed. In fact, agents face a significant challenge even to know what agents are present at any given moment; because agents can enter and leave the system at any time (openness), because the system and its components are in a continuous state of change (dynamism), or because the agents themselves exhibit different behaviour, have different capabilities and have their own agenda (heterogeneity). In these cases, it is especially important to have agents that are capable of automatically tailoring their coordination decisions to respond to the prevailing context. Thus, a further motivating hypothesis is that it is important to have agents that can learn the appropriate activities and interaction patterns in such challenging environments. Specifically, an agent needs to be able to learn the right situations in which to attempt to coordinate and which CM to use in such situations. Given this observation, a natural extension to the basic coordination framework is to enable the agents to acquire, through run-time adaptation, the knowledge that is important for their decision making (Excelente-Toledo & Jennings, 2002, 2003b). To this end, the basic framework is extended by giving the agents the capability to learn to make the right decisions about their coordination problem.

Regarding the second point, many of the extant coordination mechanism have no explicit procedures or decision making models that enable agents to drop commitments to coordination tasks if better opportunities present themselves. For instance, in the basic version of the Contract-Net protocol (Smith & Davis, 1981), once an agent is awarded a contract, there is simply no mechanism for reneging. This can be detrimental to an agent's performance if it has to turn away high value opportunities in order to persist with its existing ones. The notion of commitment is something that is central to all coordination mechanisms since it provides the basis of trust on which agents perform their part of the social activity. However the imposition of unbreakable commitments can lead to irrational and inefficient behaviour. For this reason, a number of models have been developed that allow commitments to be dropped if specified contingencies arise (e.g., (Cohen & Levesque, 1990; Jennings, 1993)). While this certainly represents an improvement, the drawback is that the specific conditions under which commitments can be broken must be enumerated in advance. In dynamic and unpredictable environments this can be extremely difficult (and sometimes impossible). To provide yet greater flexibility, *levelled commitment contracts* were introduced (Sandholm & Lesser, 1995, 1996). In such contracts, either party can decommit, for whatever reason, as long as they pay the fixed decommitment penalty that is specified in the contract. This type of commitment avoids the problem of having to *a priori* enumerate specific environmental or agent states and allows agents to decommit unilaterally for whatever reason they deem appropriate. However, levelled commitments do not explicitly take the ongoing cost of participating in the coordination activity into account. This is because the decommitment penalty is assumed to be fixed, both for the contractor (manager) and contractee, no matter at what stage of the coordination process the commitment is broken.

To be more realistic, the research position of this thesis is that the penalty should vary depending on the costs the agents have sunk into the coordination. Thus, a coordination that has only recently started, and in which the agents have invested comparatively little time, should attract a lower penalty than one that has been ongoing for a significant period of time and in which the agents have invested significant resources. Here, such commitments are called *variable penalty contracts*, and it is believed that these are essential for a rational treatment of decommitment in flexible coordination scenarios. To this end, this thesis incorporates agents that exhibit varying degrees of commitments and that incorporate different type of penalties for reneging on contracts. Thus, in line with (Sandholm & Lesser, 1995, 1996), this thesis claims that an agent should be able to decommit (for whatever reason) by paying an appropriate penalty to the contract holder (Excelente-Toledo *et al.*, 2001).

In summary, the aim of this research is to develop flexible agents that can reason about the process of coordination and select mechanisms that are well suited to their prevailing circumstances. That is, *the choice of coordination mechanism is made at run-time by the agents that need to coordinate*. More specifically, the objectives of the research in this thesis are four-fold:

- To show that it is appropriate and beneficial for agents to select coordination mechanisms at run-time.
- To define a decision making framework that agents can use in order to select an appropriate coordination mechanism at run-time.
- To show that having flexibility with respect to commitments and penalties can improve performance in coordination activities.
- To show when and where adaptivity can improve an agent's decision making with respect to coordination.

1.1 Assumptions about the agent context

To clearly articulate the underlying point of departure for this research, this subsection clarifies the main assumptions that are made with respect to agency, coordination mechanisms, and the environment in which agents are situated (in line with Figure 1.1).

1.1.1 Agents

This thesis assumes that agents satisfy the definition as advocated in (Wooldridge, 2001) and discussed in earlier section. However, it is additionally assumed that agents are *rational* when they take decisions; meaning they take those decisions that enable them to attain more "successful" goals (Weiss, 1999). However, the problem here is to know whether a particular goal is more likely to be more "successful" than another. Thus, in this context, an agent is rational if, given its perception and knowledge, it makes the right choice between different alternatives. Thus selection typically aims to maximise some measure of performance (Russell & Norvig, 1995b) and here that measure is captured by a utility function [2].

A final comment about agency, as it is used in this thesis, is that the work does not assume any particular type of agent; thus agents can be cooperative or competitive. As will be clarified in subsequent chapters (3 and 4), agents pursue their own objectives (without regard to what others are doing) which clearly shows competitive behaviour. But, on the other hand, the agents may possess different dispositions to work in cooperative situations as a means to obtain a better performance.

1.1.2 Coordination Mechanisms

This thesis claims that all coordination mechanisms need to be represented in a quantitative way in order to enable agents to reason about coordination problems.

[2]The main difference between the proposal as presented in this research (see Chapter 4 for the details) and the standard decision theory (Russell & Norvig, 1995d) is that the former does not incorporate a probability distribution of any kind (nor does it reason under uncertainty), whereas the latter puts together probability and utility theory. This is why in this thesis the proposed framework is referred to as a decision making framework, rather than a decision-theoretic framework.

This assumption is consistent with Lesser's reflection on the field: *"in order to design efficient and effective coordination strategies that will work in a wide variety of environments they must explicitly account for the benefits and the cost of coordination in the current situation in a quantifiable way"* (Lesser, 1998, pp. 137),(Lesser, 1999, pp. 8).

To this end, the first step in characterising coordination mechanisms is to represent them in a common manner so that agents are able to apply reasoning techniques to discriminate among them. In this work, a characterisation that covers the following issues is adopted:

- what are the requirements that need to be fulfilled before the technique can be applied (*requirements*)?
- what are the rules that agents follow in order to complete their interactions (*coordinating algorithm*)? and,
- what degree of coordination is likely after the coordinating algorithm has been applied (*level of achievement*)?

Now, to be able to reason about selecting a particular mechanism in a given situation, an abstraction is built from this basic description. This abstraction needs to identify the meta information that enables agents to distinguish between the alternatives. In this case, the ones considered are the run-time cost to set up the mechanism (*cost to set up*) and the likelihood of it being successful (*probability of success*) [3]. These meta-data attributes are the ones advocated by Lesser and can be regarded, respectively, as being derived from the description of the requirements and the level of achievement. Moreover, a way of quantifying these values is also required and, for illustrative purposes in this section, a simple qualitative scale of high, medium and low is chosen. Combining this altogether, a generic template of Table 1.1 is produced for describing and reasoning about coordination mechanisms.

There is, however, a clarification that need to be made regarding the level of achievement in this table. Generally speaking, it represents an evaluation of the

[3] Other attributes could undoubtedly be added to this list, but here the focus is on those that are believed to be necessary (if not sufficient) for the current purposes. In particular, the reason for not considering the coordinating algorithm as an attribute of the meta-data is because it is believed that as a first step the agents do not need to reason about the details of how the interdependences are managed with each coordinating algorithm. Thus, agents initially decide whether coordination is worthwhile and, in subsequents steps, deal with the specific details of the actual steps involved.

CM: Generic template
COORDINATION TECHNIQUE components.
Requirements. The preconditions that need to be satisfied before the coordinating algorithm can be executed. This covers things such as whether the protocol requires a set up phase (e.g. in multiagent planning a goal decomposition is needed before sub-goals can be assigned to the agents) or whether it needs a particular piece of information to be available (e.g. the number of agents in the system). These requirements might be established at design time and/or during the run-time execution. Examples of the former case are when the agent knows (because it has been hard-wired) how to contact the other agents in the system, what are their corresponding capabilities and so on. Examples of the latter case are when agents acquire the same information, but, as a result of their interaction with others. **Coordinating algorithm**. The detailed plan of actions that have to be followed to achieve coordination once the requirements have been fulfilled. **Level of achievement**. The degree of coordination that is likely from following the steps specified in the coordinating algorithm.
META-DATA attributes.
Cost to set-up: A value measuring the *run-time* costs associated with the requirements noted above. **Probability of success**: The likelihood *in a given environment* that following the steps in the coordinating algorithm will result in a successful coordination.

Table 1.1: Generic template of a CM.

quality of the outcome. Now, measuring the quality of coordination is a difficult task in most circumstances because it is highly influenced by the perspective from which it is evaluated and who actually performs the analysis. For example, for one agent it may be a success to accomplish a common task, no matter what time is spent on it; whereas another might require some minimum amount of time to be spent before it considers the coordination successfully. Thus, this evaluation is normally associated with a particular value judgment of whether coordination is attained or not (Durfee, 1999b; Jennings, 1996). Here, in contrast, it is not the purpose that each agent performs those judgments, neither is it about producing a common agreement about the extent to which coordination was accomplished. Rather, what is meant is that it is possible to measure and obtain a level of achievement that can be associated with features of each coordinating algorithm and the complexity of how it manages and deals with the agents' interactions. Thus, this degree of achievement is *not* related to a certain level of agent satisfaction, but it represents a level of the expected effectiveness given that agents fulfill the requirements and follow the coordinating algorithm [4].

In order to illustrate the use of this generic template, probably the most common coordination mechanism, the Contract-Net protocol (Smith & Davis, 1981) is represented using this nomenclature (see Table 1.2 (characterisations of several other coordination mechanisms are given in Section 2.1)). In this mechanism, a contractor agent (the task manager) is responsible for achieving a given goal. It has to decompose this goal into subtasks and indicate to other agents that it requires assistance with some of these subtasks [5]. The task manager broadcasts a task announcement to all the other agents in the system (along with any special criteria that the bidders must fulfill). Each recipient then decides whether it is interested in taking on the task and, if so, it returns a bid indicating the conditions under which it is willing to undertake the task. Finally, following the expiration of the task announcement, the task manager evaluates the bids (task deliberation) and awards the task to the most appropriate agent. Thus, a contract is established (task allocation) between the manager and the contractors with the winning proposals.

[4] This does not mean that agents have to share their opinions about this level of achievement. On the contrary, each of them might have its own particular representation of these concepts.

[5] In this protocol, the problems of dividing the problem into subtasks (task descomposition) and how the results from the various subtasks are merged into an overall solution (task synthesis) are not considered a part of the coordination activity. However, in coordination mechanisms such as multiagent planning, task descomposition might well be part of the requirements phase.

CM: Contract-Net protocol.
COORDINATION TECHNIQUE components.
Requirements. To apply the protocol the agents need to have information about how to contact one another (the identification of the possible receivers of the offer) and how to rank incoming bids from potential contractors (the selecting criteria). The information about contractors can be given at design time (in static environments) or determined at run-time (in more dynamic cases). **Coordinating algorithm**. The protocol consists of three phases: identifying potential contractors, making a decision about which contractor to select and actually enacting the agreed task. These phases are based on the message interchanges associated with the sending out of a request by the task manager, the handling of the bids from the potential contractees and the contract assignment respectively. **Level of achievement**. There are several reasons why, once selected, the protocol may fail to result in a successful coordination. For example, the manager may receive no bids from potential contractees (because they are too busy and unavailable or because they are not interested in the task at hand) or the bids received may not satisfy the manager's requirements.
META-DATA attributes.
Cost to set-up: The main set up cost is associated with awarding the contract. This is dependent on how much time is required to determine the set of potential contractors to send the request to (if this is inbuilt the cost is low, if it needs to be determined at run-time it will be more time consuming because it may involve interacting with a broker (Decker *et al.*, 1997)) and the time the agent has to wait for responses before it can make choices (the task announcement expiration-period). **Probability of success**: Because of the many eventualities that might occur, the mechanism has a medium likelihood of succeeding in the coordination tasks in the general case. If further information if available about the specifics of the environment (e.g. many agents can provide various subtasks or the agents are generally cooperative and will offer help whenever possible) then this qualification can be refined.

Table 1.2: Contract-Net Protocol CM.

This generic representation of coordination mechanisms, while not being the main contribution of this thesis, is nevertheless an important advance in the area because:

- It provides a means of standardising, evaluating and, consequently, comparing under the same premises the coordination mechanisms that have been generated in the MAS field [6]. Thus, the existing coordination mechanisms from the literature represent the instances of the possible set of coordination mechanisms that each agent could have at its disposition. From this the agent can discover from the variety of techniques the one it can employ given a coordination problem and the particular circumstances.
- It provides a means of encapsulating the coordination mechanism and the details of how it achieves coordination. It does this by only presenting the abstract consequences of its performance, releasing the agent from details of how coordination is attained.
- It provides a starting point for determining the meta-data that need to be present to enable an agent to reason about coordination decisions. Being more precise, this work asserts that the *cost to set-up* and the *probability of the success* are the minimum constituent factors that should be taken into consideration when reasoning about coordination mechanisms.

1.1.3 Environment

An important element in any multiagent system is the environment that the agents inhabit (see Figure 1.1). Specifically, this research assumes the environment is open, dynamic and heterogeneous (as previously defined). In terms of this research context, Chapter 3 describes how these characteristics are modelled in the specific testbed domain that is used to evaluate this research.

[6]This assumption does not mean that all other attempts to compare or analyse the different coordination techniques are invalid. On the contrary, any detailed analysis performed with the aim of distinguishing between coordination techniques, no matter what the perspective, helps to corroborate the value assignments introduced here (see (Jennings, 1993; Fox, 1981; Malone, 1987; Miles *et al.*, 2002) for examples of such attempts.)

1.2 Contributions of the research

By accomplishing the objectives set out by this research, this thesis advances the state of the art in the following ways.

Firstly, the very idea of letting the agents dynamically select their coordination mechanism has not been explicitly addressed within the field of multiagent systems to date. This assertion does not imply that coordination has not been investigated; on the contrary, a significant amount of research has been performed in this area. However, the study of coordination has been tackled from a different perspective. In particular, the aim of most approaches is not to reason in terms of agents taking decisions about coordinating affairs (as it is in this thesis), but rather to solve particular cases of coordination problems. There are, however, a small number of studies that follow a similar perspective to that of this thesis in dealing with the coordination problem (see Section 2.2 for more details). Nevertheless, in their work, the main drawback is that the real benefits of using these approaches has not been fully demonstrated.

Secondly, this thesis presents a formal framework for capturing the reasoning processes the agents undertake in order to select the coordination mechanism in a flexible manner. Adopting this framework means coordination mechanisms move from the realm of being imposed upon the system at design time, to something that the agents select at run-time in order to fit their prevailing circumstances and their current coordination needs. By abstracting the coordination problem, agents with this framework take coordinating decisions by considering collaboration issues and the context and timing in which those decisions are taken. These components are represented as utility-based functions that agents take into account when taking decisions. No other work in the literature presents decision making procedures that considers these aspects in this way. In the few cases where the problem is attacked from a broadly similar perspective, the problem has been postulated on alternative solutions (Boutilier, 1999), at a different level of abstraction (Decker, 1995) or by considering different components in the agent's rationale (Barber *et al.*, 2000) (see Section 2.2 for more details).

Thirdly, this research extends the basic decision making framework by incorporating the ability to reason about commitment levels and sanctions for decommitment. Being more precise, this thesis considers the decisions involved in determining when to break existing contracts in order to take up more promising opportunities and what level of decommitment penalty to set. While, several authors have regarded commitments to be the foundations of all coordination activities, the approaches followed do not normally integrate commitments into a broader decision making framework (see Chapter 8 for more details). Specifically, this work introduces the notion of *variable penalty contracts*, as an extension to levelled commitment contracts, that incorporate the ongoing cost of participating in the coordination process in the decommitment penalty.

Fourthly, this thesis shows how learning about coordination can be incorporated into the agent's reasoning model and identifies the circumstances in which such adaptation is useful. Thus the model enables agents to learn which CMs to select in specific situations and how such agents can construct simple models of their collaboration context. The existing literature on multiagent learning has not been explored in either of these directions before. In general terms, the efforts of most researchers have been directed towards the use of learning techniques (basically reinforcement learning) to improve coordination as a whole. However, though it can be argued that improving coordination is *the* final outcome to achieve, this work indicates that learning and adaptation have to focus on the agent's decision making problem first and, more precisely, on the decisions regarding coordination. Thus, the aim is here to consider pre-existing coordination mechanisms and evaluate whether to select them or not.

1.3 Thesis structure and overview

This thesis outlined is organised into ten chapters, each of which is summarised below.

Chapter 2. Related Work. This discusses related literature by focusing on existing research in the areas of coordination, flexible reasoning about coordination and multiagent learning.

Chapter 3. The Coordination Scenario. This introduces the scenario that is used to evaluate the effectiveness of the coordination decision making model and its various extensions. The scenario is an abstract one and it is based on a grid world like environment. Specifically, this chapter describes how agents behave in this grid-world scenario, the characteristics of the environment and when and how coordination might arise between the agents in this environment. The main behaviour of the agent is to pursue individual and common goals. By accomplishing common goals agents need to make decisions regarding the abstraction of a set of CMs they have at their disposal. Thus, those common goals are achieved through coordinating agents' activities.

Chapter 4. The Agent's Decision Making Procedures. This presents the basic framework that enables autonomous agents to dynamically select the coordination mechanism they employ in order to coordinate their inter-related activities. This framework formalises the agents' decision making processes; covering, in particular, the decisions that are involved in determining when and how to coordinate and when to respond to requests for coordination.

Chapter 5. Applications of the Coordination Scenario. To help demonstrate the applicability of the agent's decision framework, the main components of the framework and the abstract scenario are used to model coordination in two more concrete application domains: transportation problems and coordinated information retrieval.

Chapter 6. Evaluation Methodology. Whereas Chapter 5 seeks to demonstrate the applicability of the agent's decision making framework, it is even more important to show whether agents do indeed obtain some benefit from using this framework. To this end, this chapter introduces the experimental methodology that is used to perform a systematic empirical evaluation of the framework and the main questions formulated during this research work. Broadly speaking, this methodology consists of defining the hypothetical questions (based on experimental variables), making observations of those variables, testing (using statistical procedures) the significance of the observations and then making conclusions based on whether the hypotheses are accepted or rejected. By following such a methodology this research is able to draw conclusions with statistically significance confidence levels.

Chapter 7. Decision Making Evaluation. This evaluates the effectiveness of the decision making framework introduced in Chapter 4 on both the individual agents and on the overall system. In particular, the focus is on evaluating whether agents do select different CMs given different circumstances and the corresponding effect of those decisions.

Chapter 8. Flexible Commitments and Penalties. This develops and evaluates an extended decision making framework in which agents can drop contracts in order to exploit better opportunities. In particular, the extensions cover the dynamic setting and re-assessment of both an agent's degree of commitment to its partners and the sanctions for decommitment according to their prevailing circumstances.

Chapter 9. Learning Extensions. This introduces learning extensions into the agent's decision making and examines their impact on the process of making run-time choices about the selection of coordination mechanisms. Specifically, two cases are considered. Firstly, when an agent learns to make decisions of when and how to coordinate. Secondly, when agents learn about the key factors that influence their decisions about when to attempt coordination or not.

Chapter 10. Conclusions and Future Work. This summaries and discusses the major findings of this research and presents the key open questions that have arisen from this research.

Appendix A. Coordination in Heterogeneous Settings. This presents a range of additional experiments about the effectiveness of reasoning about coordination in heterogeneous contexts.

Chapter 2

Related Work

This thesis is primarily concerned with flexible reasoning about coordination in dynamic environments. Given this, there are a number of subfields that are related to this broad endeavour and these are discussed in the subsections of this chapter. Specifically, the related work is divided into the following topics:

- work on techniques for coordinating multiple agents (Section 2.1);
- work on flexible reasoning about coordination (Section 2.2);
- work on multiagent learning (Section 2.3).

Each of these will now be dealt with in turn.

2.1 Coordinating multiple agents

Coordination has been widely studied by MAS researchers and a number of techniques now exist for coordinating agents' interactions. However, the purpose here is not to analyse the whole range of techniques produced in the field so far, nor is it to provide a deep discussion about their relative advantages and disadvantages since there is already a significant literature addressing such issues (Weiss, 1999; Huhns & Singh, 1997; O'Hare & Jennings, 1996; Bond & Gasser, 1988b). Rather, the objective here is to show how some of the most representative coordination mechanisms can be described using the generic template introduced in Section 1.1.2. In particular, the aim is to demonstrate that each technique can be described using the proposed meta-data of CMs.

2.1.1 Social Laws

Social laws are a means of coordinating systems in which there are a large number of interactions between agents (Shoham & Tennenholtz, 1992, 1995; Briggs & Cook, 1995; Barbuceanu *et al.*, 1998). The general idea is that agents are designed to follow local rules of behaviour that lead to their acting in coordinated ways. For example, robots could follow the rule (social law or convention) of keeping to the right side of a path when navigating in a two-lane road. By doing this (or, more properly, by obeying the law), the robots avoid collisions. Thus, agents in this type of environment pursue their goals by constraining their actions by the use of social rules.

In the particular case of Shoham and Tennenholtz's work (1992, 1995), they present a formal model of a social law system based on logic. In this system, agents have a logical representation of their actions, their states and the social laws. Agents plan their actions but, their decisions about which goals to pursue at which time are ruled by the social law conventions. In particular, the agents' behaviour is constrained in the following way: given a particular state, an action, and the social laws there is a transition function that indicates the next action an agent can execute. So, in any given situation, agents identify their current state and only a subset of the possible actions that the agents could follow are designated as legal according to the social law in force. The authors show that following the convention will guarantee that the corresponding actions of the agents will avoid harmful interactions.

Briggs and Cook (1995) follow the same intuition of having agents with a behaviour prescribed by conventions. However, they claim that social laws might result in having agents that are too restricted. They therefore propose the concept of flexible laws which are a special case of social laws. Flexible laws also constrain the possible actions agents are able to perform, but the change is that they associate degrees of strictness to the laws. The more restrictive the law, the fewer actions that are possible. Thus, agents would first apply the strictest ones when constructing their plans and if the resulting plan is not successful then they select the next strictness level and so on. In this research, coordination is also achieved by following conventions, but the choices about the rules to apply had varying degrees and so offered varying degrees of rigidity in the planning process.

While Briggs and Cook use social laws to reduce the possible actions agents can select during their planning process, Barbuceanu *et al.* (1998) use the same general concept but view social laws in terms of obligations and interdictions. They represent the roles played by agents in an organization through the use of these obligations [1]. For example, the system administrator role in an organization has obligations associated with it (e.g. to update the software installed with new releases) and with its interactions with the other roles in the organization (programmers, developers and so on). Here, again, the social laws dictate the agents' behaviour but, more importantly, they describe how the interactions between roles to coordinate the agents' actions are solved.

In terms of the dimensions of evaluation (Section 1.1.2), the fundamental aspects underlying this approach are the specification of the convention laws. In the examples illustrated, it is assumed that these laws are established by a system designer. Thus, all the possible rules that govern the agents' interactions are pre-defined before the agents plan their actions. This means that the specification process is performed at design time. Once this has been achieved, there are no other requirements on the agents; they simply apply the process of selecting the actions to be executed given particular states. By adhering to the rules, the agents' actions are guaranteed to be free of violation assuming that no unanticipated situations arise. Following the CM generic template of Table 1.1, a coordination mechanism that follows the social laws approach is given in Table 2.1.

2.1.2 Market Protocols

A more general approach for coordinating multiple agents is to view the multiagent system as an economic market (Malone, 1987; Wellman, 1993; Huberman & Hogg, 1995). The basic idea here is that agents play the roles of sellers and buyers in a market fashion to agree about the price of a service or a task [2]. There are basically two elements in this kind of system: the role of the agents that take part in the process and the market structure itself. The former consists of the agents that buy or sell goods by responding to the changes in the price (the actual response is directed by their particular preferences). The latter is the interaction protocol (the coordinating algorithm) that establishes the rules of how the participants agree about the price of the good. In general, the market structure is predefined

[1] This research can also be seen as an example of organisational structures (see Section 2.1.4), however, the authors focus on the establishment of an agent's constraints rather than in the definition of the elements of the organization itself.

[2] In the literature, the terms consumers and producers are also used to refer to the activities of generating products (producers) that the consumers want to acquire (Wellman, 1993).

CM: Social laws.
COORDINATION TECHNIQUE components
Requirements. The prerequisites (i.e. the social laws, the conventions, the obligations) are hand-crafted into the system and this process is performed by a system designer. **Coordinating algorithm**. In the case of Shoham and Tennenholtz it consists of the algorithm that indicates how the agents select their next action to perform. This algorithm establishes how agents adhere to the social rule and, by so doing, it constrains the possible actions the agents perform. **Level of achievement**. This type of convention guarantees the successful coexistence of multiple agents (assuming no unexpected situation arise).
METADATA attributes
Cost to set-up: The pre-requisites are established at design time and this is a very expensive process. The reason for this is because the designer has to fully understand the system to produce a consistent set of laws, so that the complete set of possible interactions that might take place can be addressed. In particular, this needs to determine the harmful ones so they can be avoided (by the use of the conventions) and the beneficial ones that can be carried out. However, at run-time, agents do not require any additional prerequisites to apply the coordinating algorithm (low cost to set-up). **Probability of success**: Agents in this type of system are designed to follow the conventions and, consequently, this dictates that coordinated action should ensue. Thus, the level of achievement of this coordination mechanism is ensured (high probability of success).

Table 2.1: Social Law CM.

and well established (leaving no space for interactions not explicitly defined). To this end, the most well known market structures take the form of auction houses (Wurman *et al.*, 2001; Sandholm, 1999; He *et al.*, 2003) in which the type of auction indicates a prescribed guide of how the bids are treated. For example, the bids could be open and known to all the participants or they could be sealed and none of the bidders know about the others' proposals. Alternative types of auction are when the bidders must follow a defined pattern; i.e. bids should raise the current price until one bid is the winner (when dealing with ascending price auctions) or when bidders are only allowed to decrease because the process starts with a high price (descending price auction) (Sandholm, 1999; Wurman *et al.*, 2001). These protocols specify the rules the agents have to follow in order to propose, to deliver and to take decisions about the proposals [3]. However, although the protocol specifies the rules of how agents can bid, it is clear that agents still have to take their preferences into account to decide how to actually bid. Moreover, if an agent is aware of the offers of the other agents it might use a different strategy than if it does not know about their bids.

In more detail, Huberman and Hogg (1995) model the problem of managing distributed computation using a market-like protocol. The problem here is how to distribute computational resources to different computer programs in a network of computers. For example, a program that basically performs a numerical computation will prefer a computer with numerical hardware. However, a database search task might be assigned to a less specialised computer. The interesting aspect of this proposal is that the use of a particular resource or computer is assigned to the program which offers the best price for its use. In other words, the objective is to distribute the load between machines in the most balanced way to obtain a reasonable allocation of the computer network resources. In this example, the programs are the agents which propose bids given the resources they need and the market structure is defined by the use of prices which model the demand of resources.

In contrast to the previous example that clearly adopts a market-oriented approach to coordination, the Contract-Net protocol (as introduced in Section 1.1.2) is often considered as being market-oriented. However, in its basic assumptions it does not explicitly include an economic component in its bidding process, though Sandholm and Lesser extended the protocol to rectify this (1995). In the basic Contract-Net protocol, coordination is achieved through message interchange as

[3] The full range of these protocols is explored in (Wurman *et al.*, 2001).

CM: Market protocol.
COORDINATION TECHNIQUE components
Requirements. The main requirement, which is normally provided by the system designer, is how agents take decisions about the strategy to play in order to maximise their particular utilities. Another run-time requirement which, in general, is not discussed in such protocols, corresponds to the run-time identification of the agents with which communication might be established. **Coordinating algorithm**. This consists of the phases to manage the bids in the particular market structure. It indicates the rules about how the offers and counteroffers should be exchanged between the participants (Wurman *et al.*, 2001). **Level of achievement**. A market protocol cannot always guarantee to achieve success in the allocation because the buyers and the sellers might fail to agree about the price of the good.
METADATA attributes
Cost to set-up: The agents' cost for the run-time identification and the strategy to play is relatively low because this is normally specified by the designer given the particular auction rules in which the agent participates. **Probability of success**: The level of achievement of this coordination mechanisms is less likely to succeed (medium probability of success) than the social laws' coordinating algorithm.

Table 2.2: Market protocol CM.

long as the manager finds a suitable bid that satisfies the allocation criteria. In the extension proposed by Sandholm and Lesser, the idea is that in order to reach deals between the manager and the contratees, agents receive an income by performing contracted tasks and pay for the resources involved in handling these tasks (the income received minus the cost defines the agent's payoff). The system's objective is two-fold; on the one hand, an agent's aim is to maximise its payoffs, while on the other hand, the market-like structure (the coordinating algorithm) allocates tasks with global benefits.

When comparing market protocols and social laws, it can be seen that in the former case the requirements that are needed to coordinate agents' activities are mostly performed at run-time, whereas in the latter case the interactions are defined by a system designer. Considering their corresponding coordinating algorithms, the level of achievement also differs. With social laws, the coordination can almost be guaranteed, whereas in most of the market-like protocols the process of reaching agreements between buyers and sellers is not always possible. Given this observation, market protocols can be characterised by the description given in Table 2.2.

2.1.3 Multiagent Planning

Planning as a means of coordination for multiple agents was first introduced by Corkill (1979). The intuitive concept of multiagent planning involves the agents agreeing about the order in which their actions are executed in order to obtain a coordinated global plan (joint plan). Such planning involves two phases: building the plan (design phase) and executing it (execution phase). The design phase's objective consists of trying to obtain a joint plan where the actions of all the agents are scheduled and the conflicts that might cause harmful interactions have been removed. The challenge of this stage is to reconcile the various choices raised to find the best sequence of actions about which all the participants agree. This agreement not only covers the order and time at which the actions take place, but also the resources assigned to each action. The aim of the latter phase is to execute the actions of the joint plan. However, at execution time, the conditions taken into account when the plan was designed might have changed (for example, unexpected changes might occur in the environment). Thus, agents should be able to resolve any discrepancies that might occur as result of these changes.

In the multiagent planning literature there are two main approaches to constructing and/or executing a plan, the centralised and the distributed solution. The differences between them relate to whether the responsibility for constructing and managing the execution of the plan lies with a single agent or with multiple agents. This responsibility involves things like maintaining the coherency of the plan, solving problems as they arise, and giving priorities to the various actions. In the centralised case, one agent has the global vision or knowledge of the problem and is capable of solving any discrepancies that might appear when constructing or executing the plan. In the distributed case, however, several agents participate in coordinating and deciding upon their actions, about avoiding conflicts when executing the plan, and helping each other to achieve the plan [4].

Following the distributed planning approach, Ephrati and Rosenschein (1994) propose dividing the planning problem into smaller parts (each part has a corresponding sub-plan solution) and then merging the resulting sub-plans [5]. Thus, agents are assigned a sub-goal that they need to satisfy. Having done this, each agent then designs a set of possible solutions (sub-plans) that are combined to obtain an "optimal global plan" [6]. The most challenging aspect in this approach is naturally the merging process because it is here where the conflicting situations emerge. To this end, two key factors underlie the solution of this approach; the use of a heuristic value associated with the cost of constructing the global plan and an iterative synchronisation process in which agents build the global plan by solving their sub-plans in parallel. In this work, the heuristics constrain and direct the search between the possible solutions. At each iteration of the merging process, the heuristic value is re-calculated with a cost of the current state of the formed plan and with the cost of the sub-plans (which are also constantly updated given the new state of the global plan). Agents then communicate, in the form of bids, their associated plan cost. When more than one agent can supply a sub-plan for the same solution, the one with the least cost is chosen to be part of the global plan. The merging process ends when all the subgoals of the global plan have been satisfied. In this solution, the coordination mainly takes place in the merging process; namely by participating in the construction of the global plan and in the way the individual sub-plan solutions are put together.

[4] It is also possible to have all the combinations between those two approaches. Thus a plan can be constructed in a centralised fashion and then executed in a distributed or centralised fashion or the plan may be constructed in a distributed way and then executed it in a centralised or distributed fashion (Durfee, 1999a). However, the interesting cases for the multiagent research community is when distribution plays a role in either phase.

[5] Note that task decomposition and synthesis is assumed valid in this approach as it was in the case of the Contract-Net protocol (see Section 1.1.2).

[6] Optimal joint plan in this context means the one with the least cost.

CM: Multiagent planning (PGP).
COORDINATION TECHNIQUE components
Requirements. The requirements in PGP are that the organisational structure (developed at design time) provides the information about the channels of communication with the other nodes. Thus, the prerequisites needed at run-time to apply PGP are hard-wired into the agents. **Coordinating algorithm**. In PGP each node first plans its actions given its goals and then it continually takes decisions about how to coordinate with the rest. This process is continually performed by the agents and it is here where coordination is attained. **Level of achievement**. PGP endows agents with sophisticated methods to evaluate plans, execute actions and solve conflicts by revising a node's local plans in such a way that the global plan is constantly updated given the independent node interactions. Its coordinating algorithm takes into account the interactions that might emerge and provides mechanisms to deal with them.
METADATA attributes
Cost to set-up: Following the requirements, it can seen that unless the organisational structure is acquired at run-time (in which case this cost is high), the cost to set-up the coordinating algorithm is medium. This is because it is necessary to know the role each agent plays during the design of the plan and its execution and how each of them might contribute to the global solution. **Probability of success**: Because most of the existing interactions between the individual planners are being considered by the coordinating algorithm, the probability of success is high.

Table 2.3: Multiagent Planning CM.

An alternative and more general solution to deal with distributed planning was undertaken in the development of the Partial Global Planning (PGP) framework (Durfee & Lesser, 1991). PGP consists of a set of nodes (agents) that are physically distributed by region and which have to coordinate their activities to produce a coherent global plan of their actions. Each node has a partial view of the global problem (since each node covers only one region) and some regions are shared between nodes (intersected regions). PGP uses an organisational structure (see Section 2.1.4) to provide the node with the information to guide its communication, as well as how the control flows between nodes. To clarify the problem, assume that each node is a fixed videocam pointed to a specific place in a room. Each videocam does not have a total view of the room and some videocams might have intersecting viewpoints. The problem consists of constructing a global image where each videocam provides part of this total view. Now, instead of static and simple nodes, PGP assumes that each node is an agent-planner, and the partial views are local partial plans. PGP is thus a framework where the knowledge (information of each local plan) of each node is shared with the other agents in order to construct a valid and consistent global plan. This global plan refers to the collective activity of the nodes and covers their joint problem solving activity. The information each node shares with the others does not necessarily need to be correct or decisive, but can often represent the tentative local plans each node has. In this model, the nodes are constantly revising their plans to take decisions about when to coordinate with the rest. In short, in PGP coordination takes the form of a negotiation (see Section 2.1.5) in which agents agree about inconsistent partial views and of an organisational structure that guides the agents' planning process to decide when, how and where to form and exchange PGPs. This characterisation of PGP is captured in the CM template of Table 2.3.

2.1.4 Organisational structures

The aim of this research area is to design a multiagent system by making analogies with human organisations (Fox, 1981; Malone, 1987; Carley & Gasser, 1999). Coordination is achieved as a result of modelling agents with a fixed set of responsibilities in a particular organisational structure. The distribution of agents' tasks is carried out through a functional or spatial distribution, through the specialization of tasks, the division of labour and so on. The organisational structure's allure is because it establishes, through agents' roles, lines of control in the structure, authority relationships (useful for resolving discrepancies and redundancies), channels of communication between members and even the amount of knowledge that agents might interchange or communicate [7]. In short, the structure exploits the control and communication aspects between the members of the organization [8]. However, the main problem with the organisational approach is that it requires a high degree of understanding of the problem to detect how the tasks might be decomposed and how the tasks can be coupled together to find a more suitable structure. But, more than this, the decision of which coordination structure to select is taken at design time and it is assumed that the interactions do not change over time.

One of the first attempts to introduce organisational concepts in a multiagent system was undertaken by Lesser and Corkill (1983) with their work on the Distributed Vehicle Monitoring Testbed (DVMT). DVMT was built to test and evaluate the benefits of several types of solving networks (being precise, alternative configurations of networks of nodes). DVMT simulates a network of nodes that perform distributed interpretation of tracking data corresponding to vehicles moving among them. The objective is to produce a coherent path for the vehicles from the independent views of the nodes. In other words, a network is a structure in which several cooperative processing nodes work together to accomplish the global goal by exchanging their partial solutions. Here, the organisational structure represents the relationships between the nodes based on spatial (location of the nodes) and functional (tasks they pursue) requirements. The network configuration also establishes which channels of communication are allowed between the nodes and the control relationships that exist between them. For example, some configurations explore hierarchical organizations in which authority relationships are exploited, while others test a lateral organization in which such relationships are not possible. Another type of evaluation involves the functional distribution in which some nodes perform specific tasks (e.g. some nodes only integrate the results). In summary, this testbed explores a set of possible network configurations and evaluates the benefits for the system as a whole of the various options. The corresponding CM representation is given in Table 2.4.

[7] An important advantage of an organisational structure is that it has a broad applicability and it can be used as an additional source of information when dealing with complex coordinating scenarios. For example, in PGP, information exploiting this structure is used as part of the meta-knowledge that agents use in their problem solving (see Section 2.1.3).

[8] In this sense, Malone (1987) sees a market as a special case of this type of structure in which it is not necessary to establish a hierarchy.

CM: Organisational structure (DVMT).
COORDINATION TECHNIQUE components
Requirements. The more relevant aspects needed to coordinate activities are provided in the organisational structure. Whether this information is provided, acquired or learnt by agents, constitutes the essential conditions to model coordination under this perspective. In DVMT the structure is specified at design time. **Coordinating algorithm**. Agents reason and take decisions in terms of the relationship specified in the structure. **Level of achievement**. In the DVMT several measures at both the node and system levels were implemented under several points of view including the type of communication allowed, error in the data, communication channels characteristics and so on. Their aim was to explore the behaviour of organisational structures given several factors. Because of this, the level of achievement in DVMT is based on the particular organisational structures exploited and the factors taken into consideration. Thus, in more general terms, the central problem with this technique is that in general everything is sustained in the organisational structure selected. Thus, the success or failure in solving interactions is based on how this is sorted out in the selected organization.
METADATA attributes
Cost to set-up: When the organisational structure is defined at design time and agents know the role they play and the relationships between other members, the run-time cost to set-up this type of coordination mechanism is relatively low. If, however, this information is acquired during execution, this cost can become very high. **Probability of success**: The degree of coordination achieved with this technique is effective as far as it models correctly the interactions that might emerge during the execution. Once again, if the structure makes a good conceptualisation of the interactions in the system their effectiveness is guaranteed. (medium cost of probability of success).

Table 2.4: Organisational structure CM.

In a somewhat different vein, Durfee *et al.* (1989) exploit the Contract-Net protocol as a means of automatically generating network organisations at run-time. In their view, nodes in a contracting relationship can organise themselves into a supply chain every time they allocate a sub-task to another agent. For example, once a task has been awarded to an agent and it becomes a contractee, it might decide to become a manager of a sub-task of this task may therefore initiates a new announcement phase. One reason for doing this could be to find a sub-contractee that can perform the task (or a component of it) in less time or with a lower cost than itself. The flexibility achieved by sub-contracting nodes therefore generates a dynamic organisational of manager-contratee relationships.

2.1.5 Negotiation

Negotiation is a well studied approach for coordination in MAS and consequently a significant amount of research work has been undertaken in this area (see (Jennings *et al.*, 2001) for an overview). In its broadest sense, negotiation is the process that agents follow in order to look for agreements. These agreements are often pursued by agents by making offers and counteroffers in a search for a consensus. In this context, coordination is accomplished by the process of reaching agreements through conflict resolution. Given the amount of research in this area and the difficulties in drawing a firm boundary around it, Müller (1996) proposes a categorization that seeks to clearly identify its range of operation as a coordination mechanism. This classification covers the following aspects:

- Negotiation Language. This category involves the communication issues related to negotiation. In particular, this research deals with the primitives used in the message interchange when agents make offers and counteroffers. It also covers the syntax, semantics and structure of the objects the agents communicate about.

- Negotiation Decision. This is concerned with modelling the reasoning of the agents involved in the negotiation. Thus, it covers the decision making aspects, the definition of the agent's preferences, their utility functions and the different strategies they can play (conciliatory, competitive, cooperative, etc.).

- Negotiation Process. This category deals with the negotiation process from a global point of view. In particular, two aspects are analysed: the procedural

> negotiation model and the system behaviour. The former establishes the general procedure (the steps) agents follow to reach agreements, it defines the behaviour of an agent in the negotiation process. The latter covers more abstract concepts such as the quality, the fairness and the stability of the negotiation process from the participants point of view and the system as a whole.

In general terms, from Müller's characterisation, it can be seen that there are a large range of possibilities involved in negotiation protocols and, consequently, a wide variety of likelihoods for reaching agreements. For example, the more complex the terms allowed in the communication language, the more difficult it is for agents to determine the exact meaning of the communicated utterances. Moreover, any negotiation protocol indicates how the agents interact with one another and takes into consideration most of the possible situations which could be present during its execution. However, the less restrained the agents' interactions, the higher the likelihood of a conflict situation appearing during negotiation, and hence the more difficult it is to find a deal between agents.

While negotiation is clearly related to several of the aforementioned coordination mechanisms (especially the market based ones) there are a number of characteristics that distinguish it from them. First, in a negotiation protocol agents can typically produce more complex communicative utterances than those that are possible in market protocols. In general, in a market protocol, the structure of the messages used to establish communication is very simple and the number of primitives is limited. In a negotiation protocol, however, the negotiation language provides more complex primitives to generate more sophisticated communicative utterances. For example, in a market protocol, an agent can propose a bid indicating the price it expects to pay for a good and the manager's reply (following the Contract-Net terminology) can be an acceptance or a refusal. In a negotiation protocol, however, the bid might establish not only the price the agent expects to pay, but the reasons and justification why the agent assumes the price of the bid is reasonable (see (Kraus *et al.*, 1993; Jennings *et al.*, 2001) for more details). Moreover, the negotiation might be over multi-attributes (such as price, quality and delivery date), whereas auctions tend to focus on price. Second, despite the fact that both methods have common elements in the negotiation decision aspects (i.e. both protocols, in general use utility functions and preferences to guide their decisions when searching for agreements) the roles the agents play in reaching agreements in the different mechanisms may vary. In a market protocol, agents generally play competing strategies because each actor is interested in maximising its own profits. In negotiation, in contrast, agents may not only compete, they could also follow different strategies based on how the situation is developing (e.g. an agent might react and alter its behaviour if it receives a threat rather than a counterproposal). Also, price is inherent to almost all market protocols but it is not fundamental in negotiation. Thus, agents might negotiate for many other reasons. In PGP, for example, agents negotiate to solve inconsistencies and to agree to have a common or consistent point of view (this is obviously not a competitive attitude and price is not an element of the negotiation). Third, in the negotiation process category, agents tend to use a more sophisticated language that allows them to generate complex interactions protocols between agents (Labrou *et al.*, 1999). Thus, it can be seen that market protocols and, more specifically, auction protocols, look for deals under a precisely-defined protocol, whereas in negotiation protocols, agents have more freedom to argue, to convince or to give arguments about their decisions which inherently generate a more complex interaction mechanism [9].

One of the seminal pieces of works on negotiating protocols is that of Rosenschein and Zlotkin (1994). In their work, they illustrate a well studied set of negotiation protocols and, at the same time, they identify the particular situations in which those protocols can be successfully applied [10]. In this set of protocols, agents are mainly competitive and participate in a negotiation by defining the elements of the negotiation decisions, such as preferences, utility functions and the strategy to play. The negotiation protocol indicates the rules of the game that the agents follow in order to establish agreements. To this end, agents know which move to take during the agreement process.

One particular example of their negotiation protocols is the Monotonic Concession protocol. In this mechanism, the agents' objectives are to reach agreements through the interchange of offers and counteroffers. A deal is reached if one of the agents proposes something that corresponds to, or is higher than, what the other agent expects. The protocol is split into rounds and, in each one, an agent might propose a higher offer than its previous one or maintain it at the same

[9] Naturally, there is the possibility of taking a less rigorous position in this comparison and a market protocol could be seen as the simplest version of a negotiation protocol in which agents negotiate for the price of a good. However, this is not the position taken here.

[10] It is important to notice that the main assumptions of this work are consistent with those in this thesis in the sense that they identify the importance of identifying and distinguishing the protocols with which agents deal with their interactions and with the strategy to solve the problem.

CM: Negotiation protocol (Monotonic Concession).
COORDINATION TECHNIQUE components.
Requirements. The main agent requirements are the strategy to play (the purpose of the negotiation), their preferences and to know how to contact the other agents. In other words, the negotiation decision in terms of Müller's characterisation. **Coordinating algorithm**. This refers to the steps agents follow to reach agreements. This algorithm establishes how the interaction takes place and, in the case of the monotonic concession protocol, this process is done by rounds and some actions are not permitted (e.g. decreasing the offer. **Level of achievement**. The fundamental aspect to measure the probability of success is to look at how the agents' behaviour is constrained. In other words, if the negotiation protocol constrains agent's actions, and in particular their responses, based on the other's preferences the range of possibilities are reduced. However, if these phases are not indicated, more conflicting situations are likely to emerge during the negotiation process. In the monotonic concession protocol the harmful and the beneficial interactions are clearly identified. Thus, the possible actions of the agents are constrained by phases, and, more importantly, the possible actions about how much to bid are considered in the protocol itself. If agents negotiate using this protocol, a deal cannot be guaranteed, but at least it has higher expectation of occurring than using more unconstrained protocols. Thus, it can be said that the more constrained the agent's behaviour is, the higher the probability of success of the negotiation.
META-DATA attributes.
Cost to set-up: The requirements broadly correspond to the ones of the Contract-Net protocol. Thus, the same cost can be associated. **Probability of success**: Given the above description, it can be said that the monotonic concession is a highly constrained protocol which for the purposes of this analysis is associated with a high probability of success.

Table 2.5: Negotiation CM.

level. To ensure the protocol terminates and that agents do not deadlock, agents in subsequents rounds are not allowed to offer less than they proposed previously. The outcome of the protocol can then be one of two options, if none of the agents concedes in a particular round then a conflict is reached and the protocol ends. Otherwise a successful agreement between agents is guaranteed. Table 2.5 presents this negotiation protocol using the generic abstraction of coordination mechanisms.

2.1.6 Discussion

This analysis of existing coordination techniques has added detail to some of the claims made in Chapter 1. Specifically, none of the techniques are concerned with how they may be reasoned about at run-time in order to select the method that is best suited to the prevailing situation, none of the methods clearly distinguish between the agent's reasoning about coordinating aspects and the strategy that is actually used to manage the agents' interactions and all these coordination techniques can be modeled using the generic representation developed in Section 1.1.2.

Regarding the first point, the analysis performed here showed that most of the extant coordination mechanisms are concerned with how interactions between the agents should be dealt with at design-time. Thus, their basic assumption is that the designer analyses the system's coordination needs, selects an approach to satisfy these needs, and then imposes this choice upon the individual agents and the overall system. Therefore, agents, do not undertake run-time reasoning about the selection of particular coordination protocols.

Regarding the second point (and related to the first point), agents generally incorporate in their reasoning the characteristics of the application domain and those of the coordinating strategy. Moreover, the line of separation between these aspects depends on the particular coordination mechanism. For example, in negotiation, one element clearly identified is that agents need to possess a way to reason about which agreements are more beneficial than others (the negotiation decisions following Müller's characterisation) and, the other aspect, relates to the rules which govern the agents' interactions (Müller's negotiation process). However, this is not the case with the planning mechanism in which the agent has to determine both what actions should be performed, as well as how it should interact with others.

Turning to the last point, this section has shown that some of the key existing coordination mechanisms can be represented in the common template of Table 1.1. While this is not conclusive proof that all coordination mechanisms could be represented in this way, it is at least indicative that a highly variable set of methods can be.

2.2 Flexible reasoning about coordination

As stated previously, this thesis is concerned with ensuring there is flexibility in reasoning about coordinating agents, rather than developing new mechanisms to coordinate agents or enhancing existing ones. Flexibility, in this context, means agents make more decisions about their coordination activities at run-time so that their choices can better reflect their prevailing circumstances. Thus, this work deals mainly with flexible deliberation. However, flexibility in multiagent systems with respect to coordination can have a broader scope than this (Lesser, 1999). Specifically, it can cover:

- Flexibility in particular protocols such as multiagent planning or market-like protocols (Section 2.2.1).
- Flexibility in reasoning between alternative coordination mechanisms (Section 2.2.2).

Each of these will now be dealt with in turn.

2.2.1 Flexibility in particular protocols

Durfee and colleagues have been concerned with dynamically deciding on a variety of coordination parameters in the context of multiagent planning (Durfee & Lesser, 1991; Durfee & Montgomery, 1991). In particular, they have studied two different problems:

i) deciding the level of detail to reason about in multiagent planning (Clement & Durfee, 1999a, 1999b) and

ii) selecting at run-time plans that have the highest expected quality (Pappachan & Durfee, 2000).

In the former case, the key issue is determining the right level of abstraction that agents should exchange about their plans. Being precise, the problem is the following: if agents coordinate their plans at the highest level of abstraction (the higher the abstraction level, the shorter the plan) possible conflicts could emerge in the lower levels, and, on the contrary, if the coordination is done at very low levels then the cost involved in solving the conflict of longer plans may be very high. Thus, in this work, the agents use different levels of detail of plan representation and each agent can work at different levels in different coordination contexts. This means that agents dynamically choose what to represent (what to coordinate over) using planning as the mechanism for achieving this. The authors facilitate this flexibility by deriving summary information about the representation of the plans. This covers preconditions (which must hold at the beginning of execution) and postconditions (which are the effects that must hold at the end of execution). The idea is to use this information to analyse all possible interactions (this process is done off-line) between the plans (and their corresponding sub-plans) and then obtain a summary of those interactions. The kind of information this process aims to detect is the conditions related to the plan execution (for example, which conditions *must* always hold when a plan is executed or *may* be executed depending on the path a plan follows and so on). Thus, during the run-time execution, this information can be used to coordinate the plans by exchanging this information [11]. From this description, it can be seen that their contribution is more concerned with dynamic reasoning over one mechanism (planning), rather than reasoning over a set of coordination mechanisms. Nevertheless, it is believed that the importance of the information used in their summaries can also be used when reasoning about any coordination mechanism. Specifically, in this context, it can be used to populate the coordination template for the planning coordination mechanism (see Section 2.1.3).

In the latter case, Pappachan and Durfee (2000) associate a variety of measures to plans in order to take decisions about the coordination of joint activities. To be more specific, they propose a heuristic to select the plan to execute based on performance metrics of the plan's quality. This heuristic takes three components into account: the plan reward (the reward gained after the plan execution), the plan cost (the total cost of all the actions performed) and the plan reliability (the likelihood of successfully completing the plan). The coordination process is ruled

[11] The coordination is perform using a merging algorithm as per Ephrati and Rosenschein in Section 2.1.3

by a central coordinator that takes decisions about the plan to execute based on the heuristic. The interesting aspect in this solution is that the authors see the multiagent planning problem as a multi-criteria problem in which to generate the heuristic the three components are weighted (based on which attributes are more important than others) and taken into consideration each time a conflict in the plan coordination occurs. Because in some situations one component might be more important or desirable than others, several strategies are evaluated by assigning different weights to each criteria. Thus, this work has a number of similarities to the research of this thesis; including the necessity of taking decisions at run-time, the existence of a metric to measure the quality of a coordination outcome and criteria to trade-off the alternatives. However, once again, the reasoning is performed over only one coordination strategy (planning).

2.2.2 Flexibility in reasoning between alternative CMs

This section discusses research whose aim is to introduce flexibility into an agent's deliberation process with respect to choosing between coordinating techniques.

One of the first attempts to discriminate coordinating techniques in the area of distributed scheduling problems was the work of Ramamritham and co-workers (1989). They proposed a mechanism that would deliberate about the selection of various task assignment algorithms. In particular, they assign tasks to nodes by taking into account timing and resource requirements (for example, they would prefer to assign a task to a node with enough resources to deal with it rather than to one with no resources at all). The objective of the scheduling is to maximise the number of completed tasks before their deadline (compared to the number of invoked tasks). During the scheduling process, nodes receive tasks and decide whether they can complete the task with their own resources; if not, then all nodes cooperate to locate the node which can guarantee the completion of the task.

The authors propose and evaluate four algorithms for selecting the nodes: (i) random, (ii) focused addressing, (iii) bidding and (iv) flexible. The random algorithm involves randomly selecting the node (note that in this algorithm there is no cooperation between nodes). The difference between the last three algorithms is how the nodes cooperate with one another to select the one that carries out the task. Focused addressing uses an estimation of the surplus of the nodes and assigns the task to the node with most surplus (this "node surplus" is periodically calculated by each node and it is exchanged between the nodes that previously assigned one another tasks in such a way that the most updated information is communicated to the nodes that had most recently assigned tasks to one another). The bidding algorithm is a version of the Contract-Net protocol where a subset of nodes are asked to propose bids specifying their resources and timings. Then, once the proposals have been received the node with the best bid is the one that is selected. Finally, the flexible algorithm consists of deciding whether to use the bidding, the focused addressing or both of them in an opportunistic manner. The appropriate algorithm is selected based on criteria which are estimated on a per tasks basis with the number of nodes that might complete the task (based on the node surplus information) and the system's parameters (for example the maximum surplus the systems can deal with). For example, it might select one node using the focused addressing and, in addition, perform the bidding indicating to the bidders that the offers should be sent to the selected node. This responds to the fact that the node might not have the most updated information (at the moment of the decision making) and the selected node might not guarantee the task assignment. The main underlying aspect of which algorithm to select is based on the number of possible nodes (this subset of nodes is constantly calculated) and the surplus they might offer. However, most of the information needed to make a decision is constantly communicated between the members of the network or between the most inter-related subset of the members (to avoid communication overhead). Hence, the most important aspect of this proposal is that the flexible algorithm reasons in a flexible manner about when to select a particular protocol given the particular circumstances in which the decision is taken. The most interesting result of this work is the fact that the flexible algorithm outperforms the others in most cases. This, in turn, provides some insight into the potential benefits of run-time selection of algorithms by considering the system's circumstances and by taking decisions at the time the choice is made.

From the perspective of this thesis, the main drawback of this solution is that agents are assumed to be truly cooperative and to constantly communicate accurate information. Thus, although the decision making might be performed with imprecise information (for example, the focused algorithms might not select the best node), there is always an alternative way to correct a wrong selection. Additionally, only a very small number of metrics affect the decision-making process (just the node surplus) and general parameters of the system that can be tuned. In short, the underlying approach is similar to this research in that the factors involved in the decision-making process can be out of date and uncertain, though in their research all nodes work together to obtain highly consistent information about the factors that are taken into consideration in the reasoning model.

There has also been a large body of work concerned with flexible reasoning that analyses the coordination problem from a more general perspective. For example, Jennings (1993) introduced a level of flexibility in his cooperative problem solving framework. The main hypothesis of his work is *that all coordination mechanisms can ultimately be reduced to (joint) commitments and their associated (social) conventions*. In arguing for this hypothesis, he showed that most of the possible interactions and the flexibility needed to deal with changing environments can be covered by considering commitments (pledges to undertake a specified course of action) and conventions (means of monitoring commitments in changing circumstances). This analysis is important in two senses. Firstly, it supports this thesis's claim in the sense that if all coordination mechanisms can be reduced to commitments and conventions, then it is possible to make a unified evaluation of them. That is, it is possible for all mechanisms to be analysed, measured and evaluated under similar terms. Secondly, however, the problem with his work is that the system designer is responsible for determining which conventions are present in the system, which kinds of interactions particular conventions might be used for and when to use them. This is likely to be an extremely difficult (if not impossible) task in complex and dynamic environments which is why the work described in this thesis is concerned with automating this activity.

In a similar line of discussion about flexibility, the work of Generalised Partial Global Planning framework (GPGP) (Decker, 1995; Decker & Lesser, 1997; Lesser *et al.*, 2002) sought to design and evaluate a family of coordination algorithms. This work is a generalisation of PGP (Section 2.1.3) in the sense that PGP was tied to the particular domain of distributed interpretation of vehicle tracks, whereas GPGP can be applied to any cooperative domain. GPGP allows a group of agents to coordinate their complex interactions through a set of coordination algorithms. In particular, it focuses on two key factors: i) the representation and reasoning about the features of the task environment (TAEMS) and ii) the coordination mechanisms that can respond to those task environment structures. In this framework, agents first solve their own local scheduling problem (by assigning time and resources to local tasks) and then, due to the CM activation, the task interactions are discovered and consequently handled. Here, the global problem consists of solving the interactions constrained by the activities of the other agents. Hence, GPGP can be seen as an optimization framework in which the agent's local optimization solutions are combined (because of task relationships) into a global problem constrained by quality, time or resource constraints.

In more detail, a coordination mechanism in GPGP consists of a protocol that detects and activates actions (communication, information gathering or proposal of commitments) in response to inter-agent task activities. The most important of these activities are the ones that facilitate the scheduling process (i.e. those that generate commitments). These commitments are at the core of the problem solving process because they solve the task constraints related to time and resources (agents commit to tasks by specifying the time by which the task will be satisfied and the quality with which the action will be done). To this end, GPGP uses TEAMS to represent the task structure and the aspects taken into consideration when solving the global constraint problem are modelled in TEAMS.

Thus, the set of protocols in GPGP are one-shot mechanisms that discover where commitments can be applied (or can be broken) when the agents' task structures are revised (because the coordination mechanisms indicate the time at which the structures are revised). For example, one agent might discover that it can benefit another task by satisfying its task by a specific time. Thus, it commits to do the task in the time determined and this information is then communicated to the agents involved. Hence, the CMs are triggered to generate or update those commitments which represent the central feature of solving a coordination problem (as argued by Jennings above). And, as can be seen, the representation of the tasks are fundamental to the CMs for dealing with coordinating activities.

Although the authors foresaw the necessity of having alternative mechanisms to deal with commitments, they assumed that the application of the CM is only carried out at specific moments in the coordinating process. For example, when a new relationship between tasks is detected, then the CM "updating non-local viewpoints" is triggered and, as a consequence, it might exchange information of the task structure with other agents [12]. However, this is the only moment at which this communicative action is activated. This point of view is different from assuming that two or more CMs might generate the same action (the communication of the task structures) and then the question to answer is which of them should be selected in which circumstances (which is the approach used in this thesis). This point is where the main difference lies with the research of this thesis. In this

[12]Another example is the contracting protocol (introduced into GPGP in recent times to deal with resource consumption (Lesser *et al.*, 2002)) which might be used only when more than one agent "produces" the same resource (rather than every time a coordination problem is faced).

research, the agent's take decisions about when to use a particular CM based on the particular circumstances when the decision is taken. In contrast, in GPGP the CMs are activated at pre-defined situations which might occur during execution (no more than one CM can be activated in a given situation which is why GPGP does not provide a reasoning mechanism to discriminate between CMs). In summary, GPGP sees the CMs as exclusive protocols whose execution generates alternative outcomes, and consequently, different benefits to the scheduling. Whereas in the research of this thesis, the CMs may produce the same outcome and then the problem is how to decide which one is more appropriate.

Thus far, two aspects have been identified as permitting flexibility in the coordination of problem solving. On the one hand, there is the necessity of introducing various "mechanisms" to solve the coordination problem, and, on the other, the incorporation of a mechanism to reason over the selection of such mechanisms. As previously discussed, some research has investigated the former (the GPGP framework and Jennings's conventions) and some the latter (Ramamritham's research). However, there is one piece of work that studies both aspects and this is (Barber *et al.*, 2000). The aim of Barber *et al.*'s work is to dynamically select the most appropriate coordination mechanism in a given situation. To this end, they present a software engineering framework that enables agents to vary their coordination mechanisms according to the prevailing circumstances. They also identify criteria for determining when particular mechanisms are appropriate and the decision procedures for actually trading-off these criteria. They analyse the following coordination mechanisms: negotiation (Section 2.1.5), voting (agents obey the decisions of the majority of the participants), arbitration (a central coordinator evaluates and decides on solutions for other agents) and self-modification (agents prefer to change their behaviour rather than request changes from others when a conflict is present). For each mechanism, a classification along the following dimensions is undertaken: a) constraints associated with the mechanism [13], b) cost of communication and execution and c) quality of solution. For example, for them, a negotiation mechanism has as constraints the fact that communication is required and that all the agents have the authority to take decisions, the cost associated depends on the number of agents involved in the negotiation and, finally, regarding the quality of the solution, this mechanism cannot guarantee that a solution will be found, however if a solution is reached, its quality will be high.

[13]This involves the particular mechanism's requirements; for example, whether communication is fundamental, whether some specific roles need to be played, whether any authority hierarchy is necessary and so on.

Agents then associate a weight to each of these characteristics and calculate the cost of each strategy by adding the cost of all the above mentioned features. The strategy that is then selected is the one with the minimum cost. The formulation used to calculate this cost is:

$$Cost^{i}_{strategyk} = \sum_{j=1}^{m}(w_j \times Cost_{strategyk}(y_j))$$

where y_i is the attribute under consideration (in this case, constraints, cost or quality), w_j is the weight associated to the feature j and $Cost_{strategyk}(y_j)$ is the cost value of strategy k for feature j.

An obvious drawback of their formulation is that calculating the weights for the features is not straightforward. Moreover, it is difficult to assess whether a given weight is good for an individual or for the overall performance. Further, they do not demonstrate whether agents do indeed perform any better by having the capability of selecting CMs (as opposed to not being able to do it). Furthermore, their agents do not reason in terms of the other agents in the environment, nor the environment itself and, therefore, the decisions about when to select a strategy are somewhat arbitrary. For example nothing is mentioned about the other agents' disposition during interaction and during the decision making process. This means it is assumed that all the agents agree about the strategy selected even though they are never asked about it. Finally, even though the authors have analysed each of the exemplar strategies, more work is needed to provide a more systematic decision procedure for actually trading off these criteria.

The final approach considered here of reasoning between different strategies is that of (Boutilier, 1999). He presents a decision making framework, based on multiagent Markov decision processes (MMDP), that reasons about the state of a coordination mechanism. He proposes the use of coordination mechanisms as protocols and introduces the concept of states of coordination to incorporate them in the MMDP. Each state summarises some aspect of the agent's previous experiences. For example, assume that two robots aim to move boxes (independently from any other actions) from one position to another in a common grid-world scenario. Agents could be happily moving their boxes, however, a problem arises when they need to coordinate their actions to avoid being together in a "dangerous" zone in the scenario (for which they are penalized). Thus, the idea is that when robots face the coordination problem they should endeavor to take a decision in terms of the actions which benefits them both.

To this end, the protocols are represented as a finite state machine which models the coordination mechanism with states and with possible coordination interactions (decision rules). The decision rules allow agents to constrain their selection of actions. The key insight in this proposal is to allow agents to reason not only in terms of optimal joint actions, but with the state space of coordination in such a way that agents decide the possible next action based upon the particular state of coordination. Continuing with the running example, in a corner of the grid (a specific state), the robots calculate the expected utility of each possible action in that state considering whether they had previously coordinated or not in the that state (recall that the MMDP is expanded to incorporate these states). The coordination mechanism detailed was called "Randomization" and it consists of selecting the actions based on profitable past actions and on their chances of reoccurrence. Hence, given the decision rules of Randomization, the action with the best expected reward is selected. However, agents with this framework do not reason in terms of their local states; rather, they observe and take decisions based on the global state that they are always assumed to have access to. Furthermore, 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.

2.2.3 Flexible commitments

Until now, the review undertaken in this chapter has focused on two aspects: the approaches to coordinate agents' interactions and the introduction of flexibility into coordination problems. Nevertheless, most of these techniques exploit commitments to define the agent's behaviour (i.e. agents establish their course of action by using commitments). Being more precise, by making commitments agents decide what to do to pursue their goals (through creating agreements to perform certain tasks). Thus, another perspective on the problem is that agents achieve flexible deliberation with the use of commitment. In the work that is discussed in the remainder of this section, the role that commitments play in contributing toward flexible deliberation is analysed.

Several studies have been undertaken to understand the role of commitments in multiagent environments. One of the earliest and most commonly cited strands of research in this area was that concerned with the study of the philosophical aspects involved in agents' rational reasoning (Cohen & Levesque, 1990). Cohen and

Levesque's main contribution is the definition of a logical theory in which the relationship between intentions, beliefs, goals and commitments is clearly established. While this work concentrates on the individual point of view, Jennings's research (1993) is an example of how these concepts can be used in a social setting (see Section 2.2.2). Following the same concerns of improving flexibility in complex multiagent systems, more recent studies have analysed the introduction of *rights* (actions that agents can legally perform [14]) as an important component of agents' interactions (Norman & Jennings, 1998; Norman *et al.*, 1998). In particular, Norman *et al.* argue that existing theories (such as the one of Cohen and Levesque) must be extended to include agreements as a combination of rights and actions to define additional properties such as capabilities, delegation and morality. They claim that such extensions are needed to provide greater flexibility in ever more demanding cooperative settings.

The aforementioned research investigates an important and complementary aspect of commitments as they relate to the broad area of multiagent systems. Here, however, commitments play a more specific role: to define the agent's course of actions (Jennings, 1993) and to establish situations in which this course might be modified. Concretely, these concerns can be rephrased as dealing with particular courses of actions through the establishment of contracts between agents (Rosenschein & Zlotkin, 1994; Sandholm & Lesser, 1996; Kraus, 1993; Sen & Durfee, 1994) [15] and the situations in which these contracts can be relinquished. Hence, flexible deliberation can be accomplished with the use of commitment to specify an agent's agreement of performing a task at certain time [16] and a concomitant indication of when this agreement can be broken (by decommiting). Since the former aspect is dealt with in Chapter 8 the rest of this subsection concentrates on the work of decommitment.

The standpoint of this thesis is that agents should have the ability to decide at run-time to relinquish existing commitments in order to participate in more prof-

[14] Under this perspective, an agent might have the capability to perform an action but not necessarily the right to execute it.

[15] While commitments might be the result of other mechanisms (e.g. though a planning approach in which agents agree about the action to perform (Section 2.1.3)), here, the attention is on them as the result of an explicit agreement or contract about a task to undertake.

[16] Although this interpretation does not take into account the whole theory of commitment (Cohen & Levesque, 1990; Jennings, 1993; Raiffa, 1982), it does integrate the use of commitments and penalties into an agent's decision making framework. In other words, they do not address the question of what and when to commit. This does not mean that these theories are inapplicable here, but rather that further work is needed to incorporate them into the apparatus that agents use for making decisions.

itable activities [17]. To this end, a significant amount of work has been concerned with identifying the particular situations in which commitments should be reconsidered (Sandholm & Lesser, 1995; Sen & Durfee, 1994; Jennings, 1993; Kinny & Georgeff, 1991; Cohen & Levesque, 1990). However, generally speaking, this work does not deal with *decision procedures that agents can use at run-time in order to reconsider their current commitments.* Thus, for example, the formalisation of Cohen and Levesque builds upon the definition of a *persistent goal* which represents a level of commitment to construct more complex propositions. A persistent goal is one that will not be dropped unless it is no longer achievable, it has been already satisfied or the agent's motivation to achieve it changes. However, this description is not connected to a reasoning model that is able to determine whether any of these situation has occurred. In contrast, Kinny and Georgeff (1991) do evaluate an agent's decision making with respect to commitments [18] based on the changes that occur in the environment. In this evaluation, agents range from never reconsidering their plans (*bold* agents) to ones which reconsider them every plan step (*cautious* agents). They define the term *rational commitment* to allow agents to react and to reconsider their current plan given environmental factors. They detect and assess specific situations (reaction strategies) in which an agent has to deliberate about what is occurring in the environment. In general, their commitments are all related to an individual and an agent's internal aims and they do not provide a model for determining in which situations such reaction strategies are effective.

Closer to the aims of this thesis is the work of Sen and Durfee (1994) in the domain of distributed scheduling. They focus on the use of two strategies: committed and non-committed to carry out contracts [19]. They evaluate the impact of various environmental factors on the strategies' effectiveness. Their results show that unequivocal commitment to a task leads to poor performance and that having the ability to re-assess commitments is important in improving the agent's effectiveness. However, though they developed a matrix of choices and discover the rules about when each commitment and non commitment should be applied (given a particular set of qualitative values of the environmental factors considered), their analysis was performed off-line. This, in turn, limits the run-time flexibility of the agents.

[17] This contrasts with many coordination protocols in which commitments are taken to be unbreakable (e.g., (Rosenschein & Zlotkin, 1994; Kraus, 1993)).

[18] Here commitment refers to the fact that once an agent has adopted a plan, it does not consider re-planning.

[19] In this research, an agent commits when it blocks its calendar so that no other meeting can be scheduled.

The final strand of work in this area is that of Sandholm and Lesser who propose a novel mechanism, based on the Contract-Net protocol, in which agents can reconsider and drop their existing contracts by paying a pre-agreed penalty (Chapter 1). In particular, this work develops the concept of levelled commitment contracts that builds upon the basic intuition that agents should be able to unilaterally decommit from a contract, for whatever reason, as long as they pay some penalty. Given the dynamic and unpredictable nature of the environment considered in this thesis and the self motivated nature of the coordination participants this appears to be the most suitable approach. However, Sandholm and Lesser's model has a number of shortcomings that need to be rectified to be applicable to this context. Firstly, their aim is not to investigate general aspects of coordination nor dynamic selection of coordination mechanisms. Thus, their model only allows agents to reason about commitments and decommitments. Secondly, levelled commitment contracts assume a fixed penalty for decommitment that ignores the current costs of the ongoing coordination activity. Thirdly, Sandholm's original proposal contained no algorithms (decision procedures) for agents to compute when they should decommit from a given contract. This was rectified in (Sandholm *et al.*, 1999; Sandholm, 2001), however only in a limited manner. In particular, his algorithm for computing the Nash equilibrium decommitment threshold relies on the fact that agents have information about the actual and likely alternative options (as well as their probability distribution functions) that may be presented to the agents with which they are coordinating.

In order to clarify the analysis of commitments carried out in this section, it is pertinent to remark on some points which are the underlying justification for dealing with commitments in this research. Firstly, commitments are important because they allow agents to model the expected future actions of others and reason under those terms. Thus they are an essential form of reasoning about coordination. However, it was also discussed that unbreakable commitments limit flexibility to respond to new situations and, hence, decommitments need to be dealt with. Secondly, from the above, it is clear that a significant amount of research considers the fact that some level of flexibility can be achieved by having the ability to relinquish current contracts in order to engage on other more effective ones. However, the mechanisms to deal with such decommitments are varied. For example from Kinny and Georgeff's perspective, decommitments are important in order to have more reactive planning agents (with different degrees of commitments) to be able of considering changes in the environment, however, their agents do not have a mechanism to deliberate when a particular strategy must be applied. Sen and Durfee undertook a close study of the environmental factors that should be taken into consideration in order to commit to a schedule or not. However, the shortcoming in this analysis is that agents do not have degrees of commitments as suggested in (Kinny & Georgeff, 1991; Sandholm & Lesser, 1995). Thirdly, turning to the penalty for decommitment, except for (Sandholm & Lesser, 1995), none of the previous studies use a flexible penalty mechanism that can be associated with the contract. Furthermore, neither (Kinny & Georgeff, 1991) nor (Sen & Durfee, 1994) consider compensation for decommiting, agents do so because others wanted to decommit and cancel the contract. To this end, it is believed that sanctions must be a fundamental element when dealing with competitive agents so a new model of commitment needs to be developed (see Chapter 8).

2.2.4 Discussion

The research community in the area broadly agrees about the need of flexibility as a key element when reasoning about coordination. Thus, a variety of research positions have investigated how flexibility can be introduced in different aspects and at different levels of coordination. However, from the perspective of this research, these can be classified as introducing flexibility into particular cases of coordination mechanisms (Section 2.2.1), in a restricted manner (Section 2.2.2) or by only dealing with a limited portion of the problem (Section 2.2.3). Thus, although the research of this thesis agrees about the fundamental use of flexible deliberation related to coordination mechanisms, it argues that an integrated framework covering the aforementioned components needs to be developed.

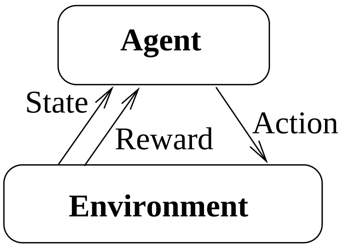
2.3 Multiagent learning

Thus far, it has been assumed that flexibility is achieved by incorporating more information in the agent's deliberation, by delaying as far as possible the choices about coordination or by reconsidering when to decommit from established contracts (Section 2.2). However, in most of these examples, the agent's behaviour remains constant throughout the whole decision making process. More specifically, agents do not change the way they reason even if things change in the environment that should affect their future decision-making process. For example, consider the case of a manager that announces a request to coordinate a task using the Contract-Net protocol. If, most of the time, it does not receive answers (bids) to its request for coordination, it would be better off if it avoids selecting this means of coordinating in the future. Similarly, in the situation in which the participating agents often request reasonable bids and receive replies to their requests to coordinate, this protocol should become the dominant means of achieving coordination. Thus, another means of incorporating flexibility into agents' coordinating behaviour is to allow them to observe the environment and use this perception to take adequate coordination decisions. In the previous example, agents could take advantage of their experience and detect in which of the two cases it is worth applying the Contract-Net protocol and when it is not profitable to do so. Thus, generally speaking, in open and dynamic environments agents need the ability to react, adapt, model and learn from what is occurring in their environment in order to know what to do in face of unexpected situations. In short, agents can also achieve flexible deliberation by learning.

The field that deals with learning from experience in a multiple agent setting is known as *multiagent learning* [20]. There are many strategies that can be employed by the agents to achieve such learning; including learning from examples, rote learning and learning by analogy (Sen & Weiss, 1999). However, most of the MAS work in this area has used Reinforcement Learning (RL) (see (Kaelbling *et al.*, 1996; Sutton & Barto, 1998; Mitchel, 1997b; Russell & Norvig, 1995e) for a general overview and characterisation of reinforcement learning techniques). RL is a technique in which agents improve their behaviour by relying upon their past experience. Specifically, this strategy uses some form of feedback to indicate the level of efficacy acquired during learning process. A RL technique is appropriate in this context because this thesis is concerned with agents pursuing goals and obtaining rewards according to how effectively those goals are accomplished.

To give an idea of the procedure the agents follow when learning by reinforcement, consider the cycle of actions that an agent performs (see Figure 2.1). The basic problems to be addressed in modeling a RL technique are: to recognise the possible and different situations that the agents might experience (step [1]); to identify the group of actions that each agent can perform in each situation and select one (using an *action-selection* procedure (step [2])); and to decide when it

[20]The subfield of Artificial Intelligence (AI) that focuses on learning from an individual point of view (i.e. agents do not possess social awareness) is called Machine Learning (ML) (Russell & Norvig, 1995c; Mitchel, 1997a; Weiss & Dillenbourg, 1999; Sen & Weiss, 1999; Stone & Veloso, 2000).



[1] The agent senses the environment

[2] The agent makes a decision about the next action to execute by analysing its current circumstances (state).

[3] The agent executes the selected action.

[4] The agent obtains a reinforcement reward for the action it performed.

[5] Go to [1].

Figure 2.1: Agent's procedure when learning by reinforcement (taken from (Mitchel, 1997b).

is appropriate to give the reinforcement (step [4]). The learning process consist of performing steps [1] to [4] until the actions *converge* to their optimal values. In general terms, there is a function that approximates (from now on called an approximation function) the expected utility of the actions given their previous effectiveness. Thus, actions are evaluated with an expected utility which is updated and maintained with the reinforcement received in step [4] after executing the action (step [3]) given a particular state.

From the above, two concepts need further explanation (the action-selection procedure and the convergence state). The action-selection procedure is normally referred to as the "Exploitation versus Exploration" choice (Sutton & Barto, 1998) and it basically chooses the action to be executed. Agents with this procedure balance their decision between selecting an action that, when performed in the past, brought about a positive reward, and an action that has not yet been performed, and, therefore, is associated with an uncertain reward. There are several algorithms that suggest how and when it is better to take the risk and explore new possibilities and when it is better to turn to experience and exploit solutions that have already been tried (e.g. greedy strategies, random strategies (Kaelbling *et al.*, 1996, sec. 2)). The decision of which one to use depends on the particular of the RL technique being used. Turning to the convergence state. This refers to the condition of an RL algorithm to reach a termination point. This point represents the state where the agent attains the best values for its available actions (or the pairs state-actions) in a given state. In other words, while the algorithm is executing, the approximation values are being refined and when this situation is reached, then it is said that the algorithm has converged.

As already stated RL techniques represent a category of methods related to the steps shown in Figure 2.1, but that differ in how each step is performed. For example, some methods give an indication of whether agents always perform the same action given the same situation, or whether agents have a guide (e.g. a probability matrix) to discriminate some situations from others, or whether agents have and use a model to represent their environment (an algorithm which does not use a model is called an online algorithm), or whether they should take the future into account in their decisions (Kaelbling *et al.*, 1996). Moreover, these methods are also concerned with how frequently the reinforcement is received and how the actions and states are represented. The formulation used to represent and calculate the expected utility is indeed related to each choice selected.

A significant literature on multiagent learning has been produced in recent years concerning the use of RL techniques and, in particular Q-learning (Watkins & Dayan, 1992) (see Section 9.2 for details of this particular technique). As it relates to this thesis, this research can be viewed as concentrating mainly on three aspects. In the first case, Q-learning has been applied to learn how agents can coordinate or cooperate to achieve common goals by using specific strategies (Section 2.3.1). In the second case, an agent's goal is to learn about the other agents or their environment in order to predict their behaviour or to produce a model of them (Section 2.3.2). And, in the third one, the agent's aim is to learn how to discriminating between the appropriate coordination mechanism to specific situations (Section 2.3.3).

2.3.1 Learning to cooperate

Generally speaking, the aim of this line of research is to improve the cooperation or coordination between the agents in the environment. The most representative research concerning coordination between agents is briefly discussed through the analysis of communicating and non-communicating agents (Tan, 1993; Sen *et al.*, 1994). The efficacy of cooperation in this research is measured by learning the tasks to achieve cooperative actions.

As an early example of improving cooperation by using communication, Tan's work (1993) uses a Q-learning algorithm to evaluate the effect of sharing perception information between independent learning agents. In this work, cooperative agents exchange different levels of information: instantaneous information (actions and reinforcements), episodic experiences (sequences of actions and their respective rewards received) and policies (not only the actions, but the situations in which the actions are performed) in order to perform joint tasks. Tan shows in a simulated prey/hunter environment that agents do learn to cooperate by sharing some information about themselves. However, there can be a drawback in the agents' behaviour. In some cases, communicating extra information can interfere with the agents' learning process because of the large number of states agents have to deal with as well as the communication cost. In contrast, Sen *et al.* (1994) explored an agent's capacity to coordinate without performing any communication action. That is, agents do not share any kind of information, but only use environmental feedback. Using such an approach, they were able to show that two robots could jointly push a block by only sensing the block position.

The research of this thesis assumes that independent agents learn to cooperate by taking decisions about their coordination mechanisms. In particular, each agent starts its own learning process independently of what the others do, and, consequently, they do not share any kind of information. To this end, the line of research advocated by Tan cannot be applied in this context (at least not as a first point to explore learning in the highest level of the decision making). However, the research carried out in this thesis builds on Sen *et al.*'s approach in the sense that agents do not explicitly communicate information to benefit the learning process and it is through external factors that agents retrieve information of the actions taken.

2.3.2 Modelling others

The easiest way to address the problem of how an agent can model another is to assume the actions performed by other agents alter the environment that the agent is perceiving and sensing (this is the approach followed with the non-communicative agents in Section 2.3.1). In such cases, agents do not model explicitly the behaviour of others. These agents are usually called *0-level agents* (using the terminology of Vidal and Durfee (1997)). To be precise, the approximation function of a *0-level agent* is not affected by other agents' actions. The next level is called *1-level agents*. These agents represent the others as *0-level agents*. In such cases, agents analyse the others' past actions and try to predict their preferences in one of two ways. Firstly, explicitly representing their actions in their formulations (by doing so, they represent the fact that the others' actions affect the feedback the agent receives). Secondly, updating the knowledge that *1*-level agents have of the others and then refining its own decision making procedures based on these new predictions (with no alteration of the formulation). The most complex kind of modeling is done by representing the other agents as *1-level agents*. These kind of agents are called *2-level agents*. This is especially complicated because here agents have knowledge of how the others select their actions. Thus *2-level agents* represent the others, not only by the actions they perform, but by their policies (i.e. the actions they perform given a particular state).

Given this nomenclature, the rest of this section considers how learning techniques can be applied to develop such agents. To this end, Nagayuki *et al.* (2000) propose a Q-learning algorithm where agents approximate the actions of the others by using a function that estimates the other agents' actions. In this work, the approximation function is represented not only in terms of the particular agent's actions, but also in terms of the actions of the others. This work is an example of modelling others as *1*-level agents and it is also interesting because communication is not allowed between agents. Another example in modeling other agents, in market domains in this case, is the work of Hu and Wellman (1998). In particular, they analyse the efficacy of the different levels of modelling in this kind of application domain. Their results show that *0-level agents* perform better most of the time because they make minimal assumptions about the others. However, their most important conclusion is that agents which use certain information about the others always perform better than those which have to operate with uncertain information. Finally, the work of Claus and Boutilier (1998) compares learning agents which represent the other agents as part of the environment (*0-level agents*) against learning agents which associate the value of the other agents' actions as part of their modeling (*1-level agents*). This enables them to compare the advantages of the two levels of modeling the others. However, in contrast with previous work, they associate others' actions with probabilistic distributions. This work also analyses in detail one of the fundamental aspects in RL algorithms; namely, the exploitation-exploration problem by considering different strategies to select the actions. They focus on analysing the convergence properties in these two kinds of scenarios. In particular, their results emphasise the complex properties of modeling agents in MASs and suggest that exploration-exploitation techniques have a great influence on the results of learning agents.

Generally speaking, the work discussed in this subsection deals with creating an explicit representation of other agents in order to predict their actions so that an agent can take more informed decisions in the future. However, this body of work, also showed that its efficiency is based on the level of modelling used, the amount and certainty of the information shared by the agents and the particulars of the learning algorithm employed. Against this background, in the research of this thesis, modelling others plays a particular role because it seems practicable to have agents that can reason as a result of their interactions with others. In particular, it is believed that agents could improve their decision making if they explicitly represent the effect of their interactions. Thus, for example, if an agent has previously received positive answers to a past request for cooperation, it could produce a model of the other's attitudes as perhaps "cooperative". Hence, when the agent faces a situation about which acquaintances to interact with this information could be incorporated into its decision making. Thus, as a point of departure, agents in this research model the elements on which they base their decision making regarding coordination problems (see Section 9.3 for more detail).

2.3.3 Learning to select a CM

To date there has been comparatively little work concerned with learning which CM to select in a given context. However, there are two systems in which such learning is exploited; namely, COLLAGE (Prasad & Lesser, 1996, 1999) and LODES (Sugawara & Lesser, 1998). The objective in both systems is to improve coordination by learning to select a coordination strategy in appropriate situations. However the aspects each system addresses are different and their findings are complementary.

LODES is more interested in having agents that are capable of learning the key information that is necessary to improve coordination in specific situations. In COLLAGE agents learn how to choose the most appropriate coordination strategy given a particular situation. Thus, LODES focuses on "what information to learn" and COLLAGE on "learning the situation where to use a coordination strategy". It is important to notice that both systems are concerned with the detailed activities of coordination as part of the learning process. For agents to solve a particular coordination problem, they have to solve all the interrelations and dependencies between their actions. Thus agents first plan the actions to perform and then execute them. To solve this, both systems have to handle explicit knowledge about the domain in the case of LODES and about coordination strategies in the case of COLLAGE.

In the case of this thesis, however, the research aim is broadly similar, but the assumptions are somewhat different and the problem is tackled using alternative solutions. In this thesis's framework, agents are endowed with a set of decision making procedures to select adequate coordination mechanisms. By dealing with an abstract set of such mechanisms, it is considered more important to have agents that have the capacity to take decisions about coordination, rather than dealing with all the interactions between them. The latter (the coordinating algorithm in the generic description of a CM) is left to the details of the subsequent tasks of the associated protocol. Furthermore, as agents are increasingly being required to deal with more dynamic environments then online learning methods (such as Q-learning) will become more important. COLLAGE, by contrast, uses instance based learning techniques in which there is a phase of recovery of examples and one of training. Consequently, the system has well defined moments in which these phases are performed which gives the additional problem of determining when each phase should finish.

2.4 Discussion

This section has further elaborated upon the basic idea that building complex systems requires the coordination aspects to be flexible and dealt with at an abstract level. In particular, it was argued that flexible deliberation is a key characteristic when dealing with agents' interactions in dynamic, open and unpredictable environments.

To this end, this chapter investigated a range of research in which some form of flexibility was introduced. Such work basically employed sophisticated techniques to deal with a variety of the problems associated with attaining flexible coordination. In particular, this analysis highlighted the fact that to embody flexible deliberation about coordination into the agent's decision making requires work in the following directions:

- Take an abstract view of the coordination problem and separate out the reasoning about the coordination problem from the techniques that are actually used to achieve coordination.
- Provide agents with flexible reasoning about coordination at the particular time at which they face the coordination problem (at run-time).
- Incorporate flexible reasoning about commitments and penalties into the agents' decision making process.

- Endow agents with learning abilities so they can improve their decision making about coordination.

Regarding the first point, some classical work was presented to illustrate how particular coordination mechanisms deal with agents' interactions. It was also shown that each of these mechanisms has its own particular characteristics and advantages. Furthermore, it was shown how they could be represented using the generic template developed in Chapter 1.

Regarding the second point, this thesis builds on the fact that agents should reason about coordination at the highest level of abstraction of the coordination problem. Moreover, in order to obtain the desired flexibility in this reasoning, agents need to constantly update their appreciation about the elements involved in the coordination. Thus, agents need to take into consideration what is occurring in the environment and with the others agents, and their decision making should be performed at point at which coordination is needed.

Turning to the third point, this thesis builds on the work of (Sandholm & Lesser, 1996) and introduces commitments through contracts as an integral part of the agents' reasoning about coordination at run-time. In particular, this work amends the shortcoming of levelled variable commitments by considering different degrees of commitments. Moreover, it is also believe that Sandholm and Lesser make somewhat unrealistic assumptions about having probability distribution available of agents' choices. Finally, it is believed that it is necessary to introduce variable penalty contracts as a more realistic model for assessing the cost of reneging. Thus, putting all this together, the research of this thesis claims that it is necessary to manage variable degrees of commitment and variable penalty contracts to better model coordination activity.

Finally, regarding the fourth point, learning and adaptation are important features when dealing with dynamic systems and reinforcement learning is an appropriate technique to apply in this context. Its main areas of application in the context of this work should be to allow agents to choose the coordination strategy which has previously been effective in similar circumstances (Section 2.3.3) and to adapt the factors on which an agent bases its reasoning to have a stronger degree of certainty about its values in the prevailing environmental in order to reduce the degree of uncertainty in its decision making (Section 2.3.2). In particular, one of the key determining factors in this reasoning relates to the likely actions or responses of the other agents in the group. To this end, agents should learn these actions from previous encounters and build a model of the likely actions of the others. Thus, as a first step of learning to model others, this research explores how to endow agents with the capacity to learn about the others as *1-level agents*.

Chapter 3

The Coordination Scenario

This chapter presents the characteristics of the scenario in which the agents coordinate their activities. It is organized into two sections; Section 3.1 introduces the particulars of the testbed domain and presents the protocol the agents follow to interact with each other and Section 3.2 justifies and discusses some of the design decisions of this scenario.

3.1 Scenario Description

The testbed domain takes the form of a grid world in which a number of autonomous *agents* (A_i) perform tasks for which they receive units of *reward* (R_i). Each agent has a *specific task* (ST_i) which only it can perform; there are other tasks which require several agents to perform them, called *cooperative tasks* (CTs). Each task has a reward associated with it, the rewards for the CTs are higher than those for STs since they must be divided among the m coordinating agents.

The agents move around the grid one step at a time, up, down, left or right, or stay still. At any one time, each agent has a single *goal*, either its ST or a CT over which coordination needs to be achieved. On arrival at a square containing its goal, the agent receives the associated reward. In the case of STs, a new one appears, randomly, somewhere in the grid, visible only to the appropriate agent. In the case of CTs, a new one appears, randomly, somewhere in the grid, but this is only visible to an agent who subsequently arrives at that square. If an agent encounters a CT, while pursuing its current goal (i.e., its ST), it takes charge of the CT [1] and must decide on both whether to initiate coordination with other agents over this task, and which coordination mechanism (CM) it should use. In this context, each agent has a predefined range of CMs at its disposal. Each CM is parameterised by the two key attributes of the meta-data (discussed in Section 1.1): set up cost (in terms of time-steps) and chance of success. For example, a CM may take t time-steps to set up (modelled by the agent waiting that number of time-steps before requesting bids from other agents) and have a probability, p, of success (thus when the other agent(s) arrive at the CT square, the reward will be allocated with probability p, with zero reward otherwise). An agent may well decide that attempting to coordinate is not a viable option, in which case it adopts the null CM (i.e. the agent rejects adopting the CT as its goal).

The Agent-in-Charge (AiC) of the coordination selects a CM and, after waiting for the set up period, broadcasts a request for other agents to engage in coordination. The other agents respond with bids composed of the amount of reward they would require in order to participate in the CT and how many time-steps away from the CT square they are situated. If an agent's bid is successful, then it is termed Agent-in-Cooperation (AiCoop) to denote the fact that it is a participant (not AiC) for a CT task. The role Agent-in-ST (AiS) is used to denote the situation where an agent is working towards a ST. Within this broad framework, Figure 3.1 highlights the specific decisions which have to be made (see Chapter 4 for more details) and gives the protocol the agents follow at each time-step.

Agents might receive more than one proposal at the same time step, in which case they reply with as many bids as the proposals they receive. However, they will only accept one CT contract at a time. Agreements between AiCs and AiCoops to achieve a particular CT are established via a contracting protocol. This Contract-Net-like protocol consists of three steps. In the first step, AiC broadcasts a proposal to all agents. It then waits for the bids. The second step involves selecting the bids and contracts from AiCs and AiSs respectively (evaluation phase). Finally, the third step consists of the commitment about the terms of the contract and the time step at which AiCoops will arrive at the CT square. Figures 3.2 and 3.3 describe in detail the reasoning each agent performs during the protocol in the evaluation step.

[1] If several agents arrive at a CT square at the same time, one of them is arbitrarily deemed to be in charge and, if an agent finds more than one CT in a given cell, it randomly selects one of them for further analysis.

[1] Agents arrive at a square. If AiS arrives at its ST cell, its goal is attained, it receives the reward and updates its goal. If AiCoop arrives at the CT cell, it notifies the AiC that it has arrived. The CT is achieved and the rewards are paid to AiCoops.

[2] If AiS finds a CT it must decide if it wants to become AiC and, if so, which CM= (t, p) it should use. If $t > 0$ it must wait t time-steps before broadcasting a request for coordination. If AiC finds a new CT, it ignores it.

[3] If AiS receives a request for coordination, it decides whether and what to bid to participate in the CT. The AiC then evaluates all bids. If AiS's bid is accepted, it adopts CT as its new goal. AiC does not respond to requests for coordination.

[4] Each agent decides on its next move according to its current goal and all agents move simultaneously.

Figure 3.1: Basic protocol followed by agents.

[1] The AiC receives and evaluates agents' bids indicating the task reward and the number of time steps they would take to arrive to the CT cell.

[2] If AiC does not receive the number of bids needed to achieve a CT or no one answers its request for coordination, it recovers the ST as its goal.

[3] From the bids received, the AiC selects the number of bids needed to achieve its CT and awards contracts to those agents and waits for the corresponding agreements.

[4] If the AiC receives the number of agreements required, the AiCoops and the AiC commit to pursue the associated CT as their current goal. If the chosen contratees do not accept the contract, the AiC will try to award the contract to the remaining potential AiCoops (from the set of unselected bids).

[5] If the AiC does not find enough bids available from the unselected bids or the bids do not generate a surplus, it recovers its previous ST as its goal.

Figure 3.2: Steps followed by AiC in the evaluation phase.

[1] AiS submit a bid when it receives a request for coordination from AiCs (the AiS replies to as many requests for coordination as it receives from the AiCs).

[2] If more than one bid is awarded with a contract, the AiS has to choose which one to agree upon. It commits to pursue a CT of the selected contract with an agreement message to the corresponding AiC and it notifies the other AiCs that it does not accept their offers.

Figure 3.3: Steps followed by AiS in the evaluation phase.

To clarify the protocols associated to each role and the previous description of the scenario, Figure 3.4 shows a [10×10] grid size at a specific time step with 5 agents in the grid and three CTs. The CT in position (1,9) requires 3 agents to be achieved, the one at (2,6) needs 4 agents and the one at (9,8) requires 2 agents. In the specific moment shown, the AiC-A_1 (at (1,9)) negotiated and is in agreement with two AiCoops (A_4 at (4,6) and A_3 at (4,8)) to achieve its CT at (1,9). A_0 and A_2 are AiSs (at (4,4) and (6,5) respectively) that are working towards their respective specific tasks at (2,2) and (6,5). No agents have found the CTs at (2,6) and (9,8).

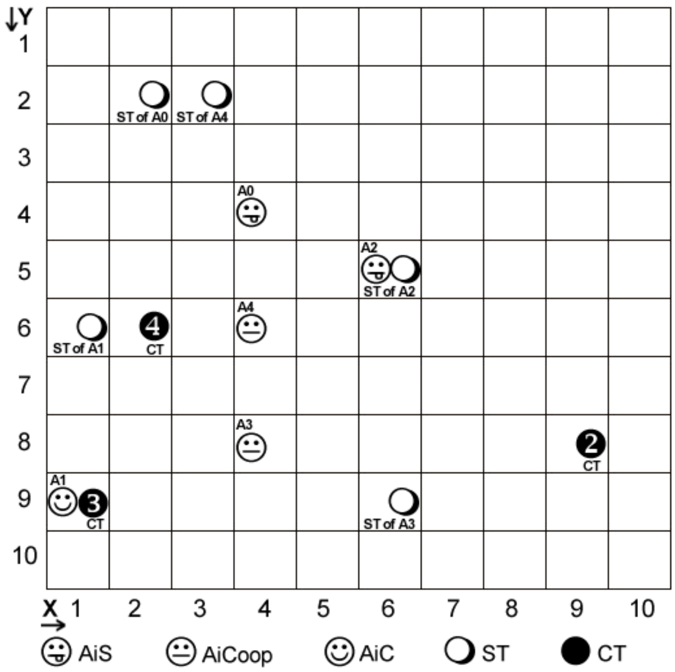


Figure 3.4: Scenario with agent roles.

3.2 Discussion

This initial presentation involves several simplifying assumptions; in particular common knowledge, a deterministic environment and straightforward coordination mechanisms. However, the framework is also intended to be flexible so that these and other assumptions will be relaxed in future work (see Chapters 8, 9 and 10). To model dynamism, unpredictability and open features (as per Section 1.1.3) in this grid world, the elements in the environment change their values at execution time. Some examples are the values associated to the tasks' rewards (both for

STs and CTs); the frequency with which tasks appear and disappear in the grid; the changing number of agents in the environment; and the number of agents needed to achieve a CT. The main consequence of these variations is that they generate an environment in which agents face difficulty in estimating the decisions of other agents. Thus, agents have to take decisions based on factors that cannot be predetermined.

In choosing a scenario in which to evaluate the model this work faces one of the perennial problems of empirical research (see (Cohen, 1995) for a fuller discussion): should this research use a concrete real world domain or should it work in an abstract environment? Choosing the former means concerns are raised about the generality of the results. Choosing the latter means there are concerns about the applicability of the developed models or the simplicity of the scenario (see (Hanks *et al.*, 1993) for a discussion of the relative merits of such a choice). The choice made here, a grid world scenario, obviously falls into the abstract environment category. This decision was made because the primary objective is to focus on the essential aspects of an agent's decision making about coordination and it was felt that this is best achieved without the extraneous constraints of a real-world application. Nevertheless, it is believed that the scenario models the key features related to making coordination decisions that are present in many real world scenarios and that it incorporates the necessary degree of dynamism and uncertainty to fully evaluate the coordination model. To this end, the scenario has been deliberately set up to concentrate on the decision making involved in coordination. Thus the remainder of the agent's decision making capabilities were minimised so that differences in performance are solely attributable to decisions about coordination and not anything else. For example, an agent can only pursue one goal at a time (meaning the results are not influenced by how effectively an agent can interleave execution of multiple concurrent goals) and that agents cannot renege on commitments (meaning the model does not have to reason about commitment strategies and types of sanctions; however, see Chapter 8 where this is relaxed). While adding such functionality would inevitably improve the agents' performance, and increase the degree of realism, it may also make the effect of the coordination decisions more difficult to determine.

Moreover, the parameterised behaviour of the environment means that the experimental conditions can be fully reproduced in order to allow meaningful comparison between the different coordination techniques. However, to help show that the concept of this scenario (and the subsequent reasoning model) are applicable in realistic settings, Section 4.5 discusses additional situations (to those introduced here) that agents might encounter as a result of possible interactions and Chapter 5 provides a mapping into the domains of transportation management and coordinated information retrieval.

Chapter 4

The Agent's Decision Making Procedures

This chapter formalises the basic decision procedures of the agents; extensions of this model to deal with decommiting from agreements and learning about the most suitable CM to coordinate with are dealt with in Chapters 8 and 9 respectively. To study the average impact of coordination mechanisms, an infinite horizon model of decision making (Filar, 1997) was adopted because this work is concerned with the long-term performance of agents; a finite horizon model may lead to erratic behaviour as the last time-step approaches (Axelrod, 1985). However, there are still two ways to model the agents' decisions: by using average reward per unit time or by discounting future rewards. Here, the former was chosen, since it simplifies the decision analysis.

The agents' aims are to maximise their reward; in particular their average reward per unit time. Each agent keeps track of its own average reward, termed its *reward rate*, and it uses this rate to decide how much to charge for its own services and, occasionally to approximate the expected rates of other agents (when it is not able to build up a picture of them (see Section 9.3)). Specifically, each agent uses its reward rate to evaluate and compare the different actions available to it; if it can maintain or improve this rate, it chooses to do so.

Agents may have various dispositions with respect to cooperation and the characterisation of sociality used here is captured by an agent's *willingness to cooperate* (WtC) factor (based on (Hogg & Jennings, 2001)). This factor, ω, represents the weight an agent puts on opting to cooperate, relative to collecting its usual reward. When reward units, effectively the agent's utility, are of equal currency, a Neutral agent ($\omega = 1$) only needs to receive the same reward from a CT as it would from its ST. Thus, if $\omega > 1$ it can be described as Greedy, asking for more reward than it would normally expect to receive and if $\omega < 1$ it can be described as selfless or Altruistic, asking for less than it would normally expect to receive. The decision procedures described in this section will typically assume that agents are neutral, but will include ω to indicate where this factor comes into the calculations.

In the model there are four types of decisions that agents are required to make: (i) the direction to move in; (ii) which CM to adopt, if any; (iii) how much to bid when a request for coordination is received; and, (iv) how to determine which bid to accept, if any. Each of these is now dealt with in turn.

4.1 Deciding the direction to move

An agent always has a target square in which its current goal is located. The agent decides to move towards its goal by selecting the direction, up, down, left, or right, probabilistically according to the ratio of up/down to left/right squares away from the goal it is. Formally, if the agent is at square $[x1, y1]$ and its goal is at $[x2, y2]$, the probabilities (pmove) that it will move in any given direction (up/down, left/right) are given by:

$$\mathsf{pmove}(\text{up/down}) = \frac{|y1 - y2|}{|x1 - x2| + |y1 - y2|} \qquad \text{(up if } y2 < y1\text{, down otherwise)}$$

$$\mathsf{pmove}(\text{left/right}) = \frac{|x1 - x2|}{|x1 - x2| + |y1 - y2|} \qquad \text{(left if } x2 < x1\text{, right otherwise)}$$

4.2 Deciding which CM to select

An agent which, while pursuing its current goal, encounters a CT must decide whether to initiate coordination with other agents in order to perform it. To do this, the agent must determine whether there is any advantage in so doing. This depends not only on the reward that is being offered, but also on the CMs available, as well as on various environmental factors which effect the expected demands of the potential coordinating agents.

To model the *expected demands* of the other agents, the AiC assumes they are randomly distributed throughout the grid, and that their current goals are similarly distributed. Thus some agents may be near the CT while others may be far away; likewise, for some agents there would be a significant deviation from their ST to reach the CT, while others may be able to coordinate over the CT en route to their own goals. The agent then assesses the possible CMs on the basis of how long before the task can be performed and how much reward it is likely to obtain after deducting the expected reward requirement of the other agents. In the former case, it considers both the set up time and the average distance away each agent is situated, whereas the latter value is based on the amount of time agents must spend deviating from their path and the CM's probability of success. This assessment determines the amount of surplus reward the agent can expect, over and above what it expects to obtain during its normal course of operation (i.e., its own average reward per time-step, r). The agent then selects the CM that maximises this surplus [1].

To formalise this decision procedure, consider an $[M \times N]$ grid with reward size S for STs, and R for CTs, a coordination mechanism, CM=(t, p), which costs t time-steps to set up and has a probability of success p. In this grid world of known size, the agent can calculate the expected average distance ($\mathsf{ave\_dist}$) away of any randomly situated agent from the CT square as well as the likely average deviation ($\mathsf{ave\_dev}$) such agents would have to make to get there.

First, the average distance in each direction of a random square from a point $[x, y]$ is given by:

$$\mathsf{x\_distance}(x) = \frac{2x(x-1) + M(M+1-2x)}{2M}$$

$$\mathsf{y\_distance}(y) = \frac{2y(y-1) + N(N+1-2y)}{2N}$$

Hence the $\mathsf{ave\_dist}$ of any given agent from $[x, y]$ is:

[1] Though this may not be a globally optimal criterion for deciding which CM to use, it makes sense from a self-interested agent's point of view.

$$\mathsf{ave\_dist}(x, y) = \mathsf{x\_distance}(x) + \mathsf{y\_distance}(y)$$

The average distance, $\mathsf{ave\_dist}$, of an agent from its ST is the average distance between two random points on the grid. This is given by averaging $\mathsf{ave\_dist}(\mathsf{x}, \mathsf{y})$, over all x and y:

$$\mathsf{ave\_dist} = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \mathsf{ave\_dist}(x, y)}{M \times N}$$

Finally, the average deviation of an agent to assist in a CT at square $[x, y]$ and then go on to its ST, as compared with going straight to its ST, is given by:

$$\mathsf{ave\_dev}(x, y) = 2 \times \mathsf{ave\_dist}(x, y) - \mathsf{ave\_dist}$$

Based on these figures, the agent can assess the average surplus reward from coordinating over the CT at (x,y) using $\mathrm{CM}_j = (t_j, p_j)$. First, it must estimate its own cost in terms of how long the CM will take to set up and how long it expects to wait for the other agents to arrive. Since the AiC would expect to receive S reward units per ST, the average reward per time step would be (see Appendix A for alternative ways of calculating this rate):

$$r = \frac{S}{\mathsf{ave\_dist}} \tag{4.1}$$

The cost of CM_j is then given by:

$$\mathsf{cost}_j(x, y) = r \times (t_j + \mathsf{ave\_dist}(x, y)) \tag{4.2}$$

Second, the AiC must estimate the average amount of reward the other m agents will require. To distinguish an agent's own average reward (r) from that of the others, r_AiCoop is used to refer to the average reward of all the other agents in the environment. When AiC does not have any knowledge of r_AiCoop it uses its own average reward as an approximation (see Chapter 9 for details of how this can be learnt from past encounters and Appendix A to evaluate the effect of various ways of calculating this factor):

$$\mathsf{ave\_bid}_j(x, y) = \frac{r\_\mathrm{AiCoop} \times \omega \times \mathsf{ave\_dev}(x, y)}{p_j} \tag{4.3}$$

Third, the AiC estimates the expected surplus ($\mathsf{ave\_payoff}$) of CM_j from adopting the CT by taking into account the probability of success of the task:

$$\mathsf{ave\_payoff}_j(x, y) = p_j \times R \tag{4.4}$$

Using these estimates, the AiC can evaluate the expected surplus reward of adopting CM_j [2]:

$$\begin{aligned} \mathsf{ave\_surplus}_j(x, y) = {} & \mathsf{ave\_payoff}_j(x, y) - \\ & (\mathsf{cost}_j(x, y) + (m \times \mathsf{ave\_bid}_j(x, y))) \end{aligned} \tag{4.5}$$

When deciding which of its CMs to adopt, the agent computes its expected surplus reward from each of them and selects the one that maximises this value. If the surplus associated with all CMs is negative, the agent adopts the option of the null CM (which is defined to have zero surplus).

$y \downarrow$					
1	AiS_2				
2			CT		
3					AiS_1
4		ST_1			
5				ST_2	
$x \rightarrow$	1	2	3	4	5

Figure 4.1: Example of a coordination world grid.

To exemplify this decision procedure, consider the simple scenario of Figure 4.1 at one instant in time with two agents (AiS_1 and AiS_2), two STs, one CT and two CMs: $\mathrm{CM}_1(3, 0.9)$ and $\mathrm{CM}_2(6, 1.0)$. AiS_2 occupies a $[5 \times 5]$ grid and finds a CT

[2] Note that in order to estimate $\mathsf{ave\_surplus}$ it is assumed that m is determined in advance or is part of the agent's knowledge. However, this assumption may not always be valid for cases in which the number of cooperative agents depends on the particulars of the coordination's objective. In such cases, the agents will need to predict this number based on previous experiences or some how estimate this information (e.g., the straightforward solution is that agents maintain an average of the number of helpers each time they accomplish coordination; more complex solutions might involve building a model for each agent each time there is an interaction).

requiring one other agent with $R = 6$ at square $[3, 2]$. Assume all agents have a WtC factor of $\omega = 1$. The average distance of other agents from $[3, 2]$ is 2.6. Since the average distance between two random squares is 3.2, the average deviation of any agent from $[3, 2]$ is 2. Assume that each ST has a reward $S = 2$, then the average reward per time-step of all agents is $\frac{2}{3.2} = 0.625$. The expected surplus reward of adopting each CM is given by:

$$\begin{aligned}
\mathsf{cost}_1(3,2) &= (0.625 \times (3 + 2.6)) = 3.5 \\
\mathsf{ave_bid}_1(3,2) &= \frac{(0.625 \times 1 \times 2))}{0.9} = 1.389 \\
\mathsf{ave_payoff}_1(3,2) &= (0.9 \times 6) = 5.4 \\
\mathsf{ave_surplus}_1(3,2) &= 0.511
\end{aligned}$$

$$\begin{aligned}
\mathsf{cost}_2(3,2) &= (0.625 \times (6 + 2.6)) = 5.375 \\
\mathsf{ave_bid}_2(3,2) &= \frac{(0.625 \times 1 \times 2))}{1.0} = 1.25 \\
\mathsf{ave_payoff}_2(3,2) &= (1.0 \times 6) = 6 \\
\mathsf{ave_surplus}_2(3,2) &= -0.625
\end{aligned}$$

Under these circumstances, AiS_2 decides to attempt coordination with CM_1 (becoming AiC) because it expects to obtain a profit. Note this is not the case with CM_2, where the negative result indicates there is not likely to be a surplus. Thus, in this case, if AiS_2 only had CM_2 at its disposal it would choose the null CM (expected surplus zero) and it would continue towards its ST.

4.3 Deciding what to bid to become an AiCoop

When agents receive a request to participate in a CT they submit a bid based on the amount of reward that they would require to compensate them for deviating from their current goal. Thus, an agent's required reward is determined by the amount of time spent in deviating from the CT square, its average reward per time-step and the probability of success of the CM being proposed [3].

To formalise this, consider an agent, A_i, with ω_i and average reward per time-step r_i. The agent calculates its *deviation* (i.e., the number of extra time-steps it

[3] Note that the AiSs use the actual values of the concepts discussed, whereas the AiC's task is to make a good approximation of these components through equation (4.3).

requires to reach its ST if it goes via the CT square). Note that if, for example, the CT square lies directly on a path to the ST, the agent's deviation would be zero. Clearly, such an agent will be in a position to submit a very attractive bid, since the cost of coordinating is effectively zero.

Again by means of illustration consider the agents depicted in Figure 4.1. AiS_1 at $[5, 3]$ would take 4 time-steps to reach ST_1 at $[2, 4]$ directly, but 6 steps going via the CT at $[3, 2]$, a deviation of 2 time-steps. However, AiS_2 at $[1, 1]$ would take 7 time-steps to reach ST_2 at $[4, 5]$ directly, and also 7 steps going via the CT at $[3, 2]$; AiS_2 therefore has a deviation of 0.

To compute the reward AiS_i requires from engaging in coordination over the CT, it takes into account the compensation both for its deviation and for the possibility that the CM might fail; it also takes into account its willingness to cooperate. Thus, the estimation of bid is given by:

$$\mathsf{bid}_{ij} = \frac{r_i \times \omega_i \times deviation_i}{p_j} \tag{4.6}$$

The agent submits its bid to coordinate and its distance from the CT square. If an agent is selected to coordinate, it adopts the CT as its current goal. Its ST is only re-adopted after the CT has been accomplished.

4.4 Deciding which AiS bids to accept

Once the AiC has received bids from all agents, it selects the set that maximises its surplus reward, given the new (definite) information it has received (cf. the approximation in section 4.2). For each agent, A_i, the AiC knows the amount of reward it will require (bid_{ij}) and the time it will take to arrive (T_i).

The AiC's selection bid process is based on the calculation of the cost of each bid received. However, when more than two agents are required to achieve a CT, it is necessary to deal with the fact that an AiCoop may have to wait in the CT cell while the remaining AiCoops arrive (because agents have to travel different distances). There are many ways of dealing with this situation (see discussion below). However to simplify the estimates of expected reward undertaken by the various agents, it is assumed the AiC pays an additional reward for the time elapsed. Thus, AiC knows the number of time steps that each AiCoop is likely to have to wait (specified in the bid) and the amount it will pay for waiting time at a specific predefined waiting rate (q). The CT is achieved only when the AiC has received the confirmation of all m agents involved in the cooperation. When an AiCoop notifies the AiC of its arrival at the CT cell, it either receives its share of the CT reward or the waiting rate followed by its share of the CT reward.

Thus, to decide which bids to accept, the general idea is that AiC selects the m proposals with least cost (from the total bids received $\mathcal{B}$). It does this by considering the reward requested in the bid and the waiting time cost ($\mathsf{cost\_bid}$) and then it estimates its expected reward given this cost and its investment. Formally, AiC calculates the cost of each subset b of $\mathcal{B}$ with m elements of the form (bid_{ij}, T_i). From b, AiC selects the agent that will take the longest time to arrive (i.e., $maxT_b = \max_{(\mathsf{bid}_{ij},T_i)\in b}[T_i]$), then it can determine the maximum time that each agent will spend in the cell. Finally, it approximates the cost of each bid based on the reward and the waiting time an AiC has to pay:

$$\mathsf{cost\_bid}_b = \sum_{(\mathsf{bid}_{ij},T_i)\in b} (\mathsf{bid}_{ij} + (maxT_b - T_i) \times q) \tag{4.7}$$

Bringing all this together, AiC estimates the surplus it expects to obtain by taking into account the cost of the selected bids and its own investment to wait for the last AiCoop to arrive. The bids selected belong to the subset b of $\mathcal{B}$ that maximises the $\mathsf{surplus}$ given by:

$$\mathsf{surplus}_j = p_j \times R - \mathsf{cost\_bid}_b - r \times (t_j + maxT_b) \tag{4.8}$$

Now, it may be the case that no bids are received which give a positive surplus. Even though the chosen CM had an expected surplus, by chance it may be that no agents are sufficiently near to provide reasonable bids. In such a situation, the AiC abandons the CT and returns to its ST.

4.5 Discussion

The decision making framework covers the progression of the participants from being independent agents that seek to maximise their individual gain (competitive behaviour), through to the formation of a collaborating group that cooperates to achieve a common task (cooperative behaviour). Moreover, the degree of competition/collaboration in the various stages of the protocol can be varied through the willingness-to-cooperate factor (ω) (see Sections 7.4 and A.2 for experiments involving this parameter).

Turning now to the applicability of the decision making framework (as per the discussion in Section 3.2). It is important to emphasise the fact that the protocol specified in Figure 3.1 indicates only the general steps agents follow in this scenario. In order to formulate the agent's decision making procedures, some specific design choices had to be taken. For example, the agent's decision of which bids to select (as described in Section 4.4) models the situation in which the AiC pays compensation for the AiCoops's waiting time because all AiCoops have to be in the cell to attain the CT. However, there are other ways of dealing with the fact that various cooperative agents are required to achieve a CT (step [1] of the protocol). For instance, in some circumstances, only two agents might be needed to accomplish a CT or the cooperative agents may not need to wait for the others in order to attain the CT. In both of these cases, the AiC would not need to pay compensation for the AiCoops's waiting time. Additional situations might occur if the agents are allowed to negotiate the waiting rate. However, many of these alternatives can be modelled using the components and constituent factors introduced in this section. To demonstrate this, in what follows, alternative formulations to calculate surplus are illustrated.

In the situation in which only one more agent is needed to achieve a CT (the simplest case), the bid selected is the one with the cost ($\mathsf{cost\_bid}_b$) that maximises the surplus reward estimated by equation (4.8). This time, therefore, it does not make sense to use $maxT_b$ and T_i is used instead, and the cost of the bid is:

$$\mathsf{cost\_bid}_b = \mathsf{bid}_{ij} \tag{4.9}$$

If, however, more agents (to be precise m agents) are required to achieve a CT and AiCoops do not need to wait for the rest to achieve the cooperative task, the $\mathsf{cost\_bid}_b$ must reflect the fact that the bids are not affected by the waiting time. In this case, the cost of each subset b of m elements is approximated by:

$$\mathsf{cost\_bid}_b = \sum_{(\mathsf{bid}_{ij}, T_i) \in b} \mathsf{bid}_{ij} \tag{4.10}$$

Note that for the calculation of surplus, although AiC does not pay compensation for the waiting time, it still has to wait for the furthest AiCoop to arrive ($maxT_b$). Once again, with this new cost calculation, the surplus equation considers the changes needed and the formulation is used transparently.

In short, the alternative formulations presented for calculating surplus illustrate that although there are many situations that agents might encounter as a result of possible interactions, the main components and constituent factors taken into consideration in the agent's decision making are still valid. Moreover, the same concepts can be used to formulate the particulars of alternative applications domains. In particular, the modifications mainly occur in the calculation of cost_bid_b rather than a change in the whole evaluation of surplus. To illustrate this point further, the next chapter takes two different applications domains (transportation and information retrieval) and shows how the decision making framework outlined in this chapter can be used to model variations in the way that the domain specific components of the framework are calculated.

Chapter 5

Applications of the Coordination Scenario

To help demonstrate the applicability of the coordination scenario and the decision making framework presented in Chapters 3 and 4, this chapter shows how two commonly used examples in the MAS literature can be described in these terms. The scenarios relate to transportation and coordinated information retrieval. Specifically, the purpose here is to show how the constituent factors of the agent's decision making framework can be grounded in these application domains. In what follows, a general description of the problem is first given and then the relevant features are instantiated in the agent's decision procedures.

5.1 The Transportation Problem

The domain of transportation has been used by a number of researchers in the multiagent system community to demonstrate their ideas. A common form of this problem involves agents moving in an interconnected network from one place to another; modelling the rescue of evacuees to safe points (Durfee, 2001), modelling shipping companies (Fisher *et al.*, 1996) and the package delivery problem (Rosenschein & Zlotkin, 1994). Here the focus is on the final example. In this case, the truck's goal is to deliver a number of parcels to specific locations (post offices or delivery offices). The more parcels delivered by a truck, the more profit it receives. There are special packages (a package being a group of parcels) that have to be delivered by more than one truck (because of their size).

Now mapping this into the abstract scenario of Chapter 3. Trucks are the agents that move around the grid (mail vans in Figure 5.1) and the final destinations for their parcels correspond to the agents' specific tasks (post offices in Figure 5.1). Agents start moving around the grid with a *package* to deliver and with a post office to reach (target square). If they find more parcels to dispatch en route (which increases their benefit) they will try to incorporate them into their plan. As soon as they arrive at their target square, all the carried packages are deemed delivered and another destination is specified and new parcels are requested to be delivered. There are intermediate points (additional distributed outlets) where an agent can pick up packages to be dispatched to the final destination. These correspond to the CT locations and are represented as post boxes in Figure 5.1. At these points, agents are asked if they wish to carry the additional parcels (from now on to distinguish the original parcel to the possible additional ones, the latter are referred to as *packages*). If they have sufficient space in their truck (and if it is profitable) they accept the proposal and deliver the packages by themselves. If, however, they can only carry out part of the delivery, they have to decide whether to accept the whole package (and enlist the help of others) or to refuse the delivery. In this context, the number of packages to be delivered in a coordinated action corresponds to the CT's reward (R) and the parcels requested in the destination point to start a new journey represent the ST reward (S) [1].

In this application domain, the main decision an agent faces is whether it should accept the challenge of being responsible for the whole package (i.e. becoming the AiC). Before doing this, however, the agent would expect to find other possible trucks (AiSs) with sufficient space to help it carry the delivery. An AiS receives a request to assist in this group delivery process; evaluates its own capacity and proposes the number of parcels it is able to carry. This information corresponds to the bid the AiC will evaluate. Thus, if the AiC identifies an appropriate group (with sufficient capacity) from the AiSs' bids, these become AiCoops for this package. If, however, the AiC cannot find sufficient trucks to carry out the delivery, it simply informs the intermediate distribution outlet of this fact and returns to its original task of getting to its delivery target.

Having mapped the delivery problem into the coordination testbed, it can be seen that the trucks take decisions in terms of how profitable their actions will be by trying to deliver as many parcels as possible. However, the main difficulty in

[1] To match better with the abstract scenario, it is assumed that each truck starts with a predetermined number of packages (although this premise is not fundamental to the analysis).

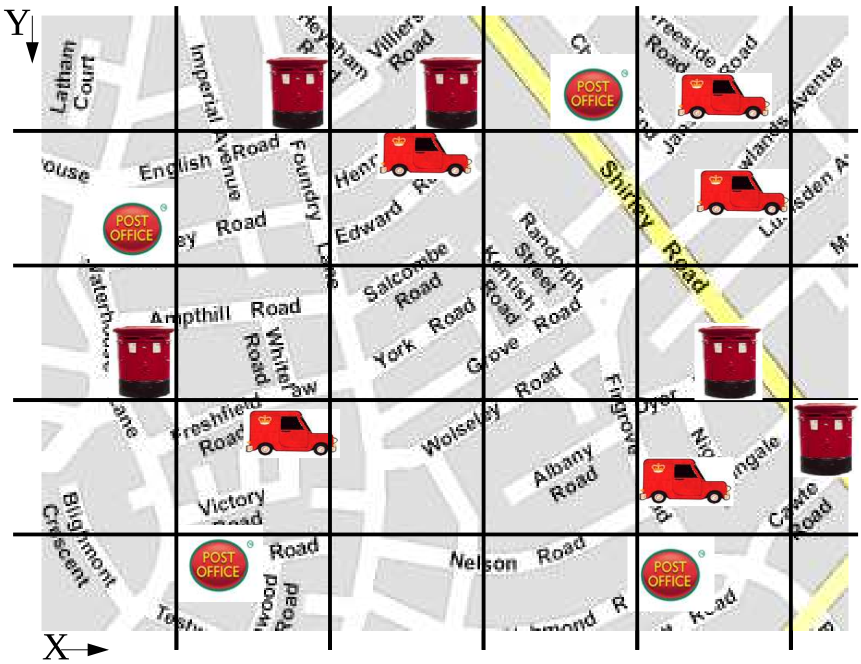


Figure 5.1: Transportation map example.

making these decisions is that they have to be based not only on the information of the individual trucks themselves, but also on the agents' beliefs about the other trucks in the environment. In what follows the main components of the agent's decision making framework (as outlined in Sections 4.1 to 4.4) are instantiated for this application domain.

To begin with, however, there are a number of aspects in which this example differs from the coordination testbed :

1. The number of agents required to achieve the CTs cannot be determined in advance. This is because this information is based on the amount of space available in the various trucks and this is not known until the AiC receives the AiSs' proposals.

2. An AiS's bid in the transportation domain incorporates the truck's space. When an AiS receives a proposal for coordination, it responds with a bid (bid_{ij}) that represents the cost for its services and the time to arrive (T_i) to the CT location. Here, in addition, the agent indicates the space available in its truck (S_i).

3. The AiC does not consider the potential AiCoops' waiting time. This is because the AiCoops do not all need to be in the cell to accomplish the

cooperative task. Rather, each AiCoop reaches the CT destination, picks up their set of parcels from the package and continues its travels onto the target location. The possibility of this situation was discussed in Section 4.5.

Having said this, in what follows, the agent's decision procedures discussed are: which CM to select?, how much to bid? and which bids to accept?.

5.1.1 Which CM to select?

This decision procedure is estimated with equation (4.5). The general constituent factors of this procedure are: ave_dist, ave_dev and r. The first two factors can be modelled with the same principles as in the abstract scenario because the interconnected transportation network can be mapped transparently into the coordination grid. The remaining factor r is calculated with the ST reward (S) and the ave_dist. As before, the agents base their decisions on this average and they are inclined toward those decisions that maximise it. Because the agents' ST destinations are their final delivery point, they would expect at least to deliver the original parcel from one destination to the final one. Thus, equation (4.1) to approximate r is employed. Given that, all elements of the CM selection procedure (equation (4.5)) can be estimated: cost (equation (4.2)), ave_bid (equation (4.3)) and ave_payoff (equation (4.4)). There is, however, one aspect that needs further discussion. This is, how do agents estimate the number of acquaintances needed (m) to achieve a CT since this information cannot be determined in advance? In this transportation example, m can take values in the range from two to the total number of agents in the environment (when $m = 1$ the AiC does not need to evaluate the ave_surplus because no-one else is needed to complete the CT and it attends to the whole package by itself). The straightforward solution to estimate m is that agents maintain an average of the number of helpers each time they accomplish coordination and use this average as m. The advantage of this simple solution is that it is based on what has actually occurred in the environment. However, in highly dynamic situations, this average might vary considerably. More complex solutions would involve building a model for each agent each time they interact (see Chapter 9 for a discussion of this approach). Thus, when the AiC needs other agents to carry the package, it evaluates the ave_surplus using the formulation detailed in equation (4.5). With this equation, the trucks are able to make the most important decision: whether it is worth accepting the whole package and which coordination strategy to use in order to coordinate with the other agents.

5.1.2 How much to bid?

This corresponds to how much the agents should bid to participate in a coordinated action (equation (4.6)). Again, it is assumed that r is updated at each step and this is used by the AiSs when they receive proposals for cooperation. So, the formulation to evaluate bid is not modified. Rather, as discussed before, AiSs now propose to the AiC the triple: bid, time to arrive and space available $(\mathsf{bid}_{ij}, T_i, S_i)$.

5.1.3 Which bids to accept?

This corresponds to the AiC's selection of which bids should be used to make agreements. The AiC makes this decision by calculating the $\mathsf{surplus}$ (equation (4.8)). Here, however, the AiC's reasoning needs to incorporate the following two additional considerations: the space in the AiS's trucks and the omission of the payment for the AiCoops' waiting time. To do this, a new $\mathsf{surplus}$ formulation is needed. This follows the same broad intuition as that of the general scenario. Formally speaking, let P be the AiC's space available in its truck, $\mathcal{B}$ be the set of bids received of the form $(\mathsf{bid}_{ij}, T_i, S_i)$, b be a subset of $\mathcal{B}$ (excluding the empty set) and $maxT_b$ be the furthest agent from b ($maxT_b = \max_{(\mathsf{bid}_{ij}, T_i, S_i) \in b}[T_i]$). The idea is then to estimate the cost of a subset b:

$$\mathsf{cost\_bid}_b = \sum_{(\mathsf{bid}_{ij}, T_i, S_i) \in b} \mathsf{bid}_{ij}$$

The subset that maximises the $\mathsf{surplus}$ are the bids that become agreements. In this case, however, it is required, in addition, to satisfy the constraint that the space offered by the bids of b:

$$P + \sum_{(\mathsf{bid}_{ij}, T_i, S_i) \in b} S_i >= R$$

Thus, AiC has to calculate whether it expects to receive a profit given this distribution. It does so based on the $\mathsf{surplus}$ formulation of equation (4.8), but this time using the calculation of $\mathsf{cost\_bid}_b$ detailed above. However there is still the possibility of not having sufficient bids to assign the packages to or that the cost asked by the AiCoops might be too high to expect a profit. In such cases, the procedure ends having analysed all subsets in $\mathcal{B}$. Then, the AiC does not accept any proposals and, as previously mentioned, it abandons the CT.

Note that the only difference between the calculation of $\mathsf{cost\_bid}_b$ discussed here and the ones discussed in Chapter 4 is that in this domain, AiC is simply trying to accept an unspecified number of bids (the cardinality of the set b). In contrast, in the examples of Chapter 4, this number was known in advance by the agents.

As can be seen, the majority of the decision procedures work as specified in Chapter 4. The alterations that are needed are relatively minor in nature and they are a natural consequence of the particulars of the transportation domain in which the number of trucks is not determined in advance and the truck's space needs to become part of the decision making process.

Turning now to the run-time selection of the CM. This thesis claims that agents need to have a set of CMs to solve their coordination problems in an effective manner. Thus, the emphasis is on how the AiC decides whether it is worth accepting the whole package and which coordination strategy should be used to coordinate with other agents. However, it is not always obvious exactly what a selection of a CM implies. In Chapter 2 a number of examples of different strategies to solve coordination problems were introduced. These coordination strategies represent the CM instances the agents reason about. By means of illustration, assume that in this domain agents use multiagent planning as CM_1 and the Contract-Net protocol as CM_2. Thus, when an agent faces the problem of deciding whether to deliver the parcel with others, it has to select between doing so using planning or through the Contract-Net protocol. Using CM_1 would mainly involve the agents having to agree about the order in which the related agents' actions need to be performed. In other words, by making an analogy with PGP (Section 2.1.3), each agent maintains its partial plan and helps to build the global one in which all the waiting times are solved. The subset of AiSs that reply to the request for coordination represent the group of agents that work together to agree about the actions each one has to perform to satisfy the time and cost constraints. Thus, CM_1 involves a complex mechanism to solve the interactions that guarantee a successful global plan. However, the price to pay is the time to set up the plans of the agents involved. Regarding CM_2 (as described in Table 1.2), the requirements to establish the protocol are less complex than for CM_1, but the outcome is also less likely to be achieved. This is because agents do not negotiate about the cost of their bids. Thus, the AiC receives the bids, selects the ones it considers convenient for its purposes and refuses the others. When an AiS receives a refusal, it does not counterpropose a new offer, even though it could perhaps reduce its cost to get its bid accepted. On the basis of the previous discussion, it is clear that the characteristics of these two examples of CMs vary in how likely success is going to be and how much effort needs to be invested. What is more, the particulars of each CM change from situation to situation. In this transportation example, agents would probably prefer to coordinate through the Contract-Net in some circumstances and use the planning approach in others. For example, when the frequency with which agents find post boxes is high and losing some opportunities for cooperation might not be too expensive, the Contract-Net would seem to be more appropriate; whereas PGP would be more suitable finding post boxes are seldom found and agents might prefer to invest more time and ensure they do not loose the opportunity to deliver the additional parcels.

5.2 Coordinated Information Retrieval

This problem consists of having a number of agents with the task of downloading documents from specific locations in the Internet (Huhns & Stephens, 1999). The action of downloading has an associated cost that represents the price paid for the use of the server [2]. The agent's objective is to reduce, as far as possible, the cost of downloading. Each time an agent has a document to retrieve, it might download it by itself or it could minimise the cost by coordinating its activities with those of other agents that are also interested in the same document.

The Internet downloading domain has a number of characteristics that can be found in many other multiagent application domains. To mention some: loaning a book from a library or querying a database (Rosenschein & Zlotkin, 1994). In the former case, an agent's aim is to reduce the cost of loaning the book between the interested agents. In the latter case, each agent has a set of SQL queries to perform in a database. Executing a query means having access to the database and spending time performing the query. Agents would prefer to find which others are interested in performing the same query in order to avoid the cost of doing it by themselves.

Thus, returning to the problem of coordinated retrieval in the abstract scenario, there are two kinds of document to retrieve: private and public. The former are

[2] Actually this cost might also involve paying for the copyright of the document or perhaps the price asked by the company for maintaining the server.

only of interest to an individual agent, while the latter might have a broader appeal. Thus, the ST of the abstract scenario models retrieving private documents and the CT represents the public ones. Thus, Figure 5.2 illustrates an AiS agent which deals with its private document by itself and, on the other hand, an AiC and a number of AiCoops (5) which have agreed to share the cost of downloading a public document. Agreements between agents in this context means that the AiC gives the permission to the AiCoops to access the server from which the document can be downloaded.

In the abstract scenario, the agent's aim is to maximise the expected utility, whereas in this domain it is to minimise the expected costs. Thus, there is a correspondence between the reward obtained by achieving tasks in the abstract scenario and the cost of downloading a document on its own. In other words, R in this domain represents the cost of downloading a public document and S represents the cost of downloading a private one.

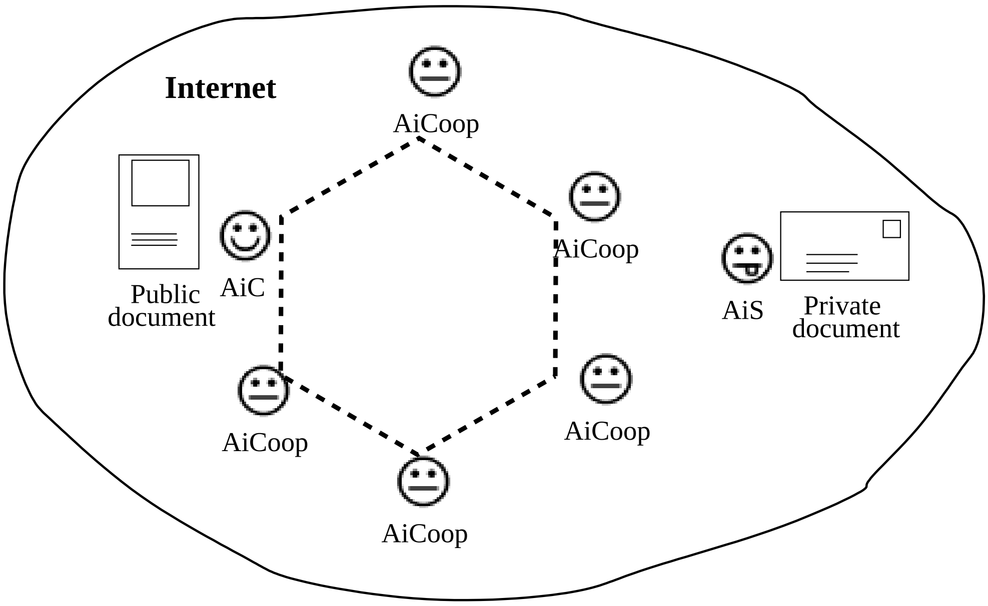


Figure 5.2: Internet document retrieval.

Agents start with a list of private and public documents to download. The agent's general behaviour is associated to the actions they perform when they recover a private document or a public one from its list. If it is a private document, the agent locates the server and simply retrieves the document. Once downloaded, the agent recovers the next document from its list. If it is a public document then it has to decide whether it is worth trying to cooperate with others (becoming an AiC) or not. If it finds a CM with the least expected cost of downloading, then the

AiC decides to attempt coordination. To this end, the AiC waits until the time to set-up the CM selected has elapsed and it then communicates to the others its intention of downloading a document and the coordination process starts (as per Figures 3.2, 3.3).

To better model the fact that an agent's objective is to minimise costs, rather than maximise rewards, the decision making formulation is mainly redefined in its meaning (not in its components). Thus, for example, ave_surplus (equation 4.5) represents the average reward agents expect to obtain with a particular CM, while in this domain, it becomes ave_cost which corresponds to the average cost an agent expects to pay for downloading a document. Similar actions are taken with the remainder of the agent's decision procedures. Each of these will now be discussed in turn.

5.2.1 Which CM to select?

In the abstract scenario, agents employ the concept of distance between grid cells to model the average distance and deviation (ave_dist and ave_dev). In this domain, an agent has to "travel" from one point to another on the Internet (to be precise from one server or node to another) because it has to be at the server location to download the documents. On this basis, the concept of distance is modelled using the notion of Internet routing cost (Baccala, 1997). This simply consists of calculating the number of servers an agent has to pass through in order to make the connection between nodes [3] to move from one source destination to the final one. Regarding the average reward per time step r, here, it is necessary to model the average cost per time step spent by the agents. Therefore, with the new semantics for S (the cost of downloading a private document) this average is estimated with equation (4.1).

Building on previous factors, the three main components to select a CM (equation (4.5)) cost, ave_bid and ave_payoff are now analysed. Once again, the objective is to minimise cost. In this context, the calculation of the expected surplus of achieving a CT (ave_payoff) requires a change in its meaning to reflect the expected cost (exp_cost) agents have to pay for downloading the document given the

[3] In the Internet domain there are many solutions to calculate the routing cost (Baccala, 1997). This cost involves building on and assigning costs to network paths and then the routing protocols select the path to the destination using some specific criteria. This criteria, in most cases, is the path with the least cost.

specific CM being used. The remaining components including the cost the AiC has to invest in setting up the CM (cost) and the average bid ($\mathsf{ave\_bid}$) it recovers from others' help are estimated using the standard equations of (4.2) and (4.3) respectively [4]. Thus, the average cost an agent expects to pay (instead of $\mathsf{ave\_surplus}$) for downloading a document taking into account $\mathsf{exp\_cost}$ is given by:

$$\mathsf{exp\_cost}_j(x, y) = p_j \times R$$

$$\mathsf{ave\_cost}_j(x, y) = \mathsf{exp\_cost}_j(x, y) + \mathsf{cost}_j(x, y) - (m \times \mathsf{ave\_bid}_j(x, y)))$$

Note that the main alteration in this new formulation is that some components change their role in terms of reducing or increasing the total cost. Being more precise, cost increases the total cost rather than decreasing the gain and $\mathsf{ave\_bid}$ decreases the total expected payment rather than reducing the expected reward obtained by achieving the CT. In summary, an agent calculates the $\mathsf{ave\_cost}$ of each of the CMs at its disposal and then selects the one that minimises the cost.

5.2.2 How much to bid?

When an AiS receives a coordination proposal, it replies with the amount it can contribute (bid_{ij}) and the time it would take to attend the request (T_i). In other words, bid is based on its own average cost per time step (equation (4.6)) and T_i represents the fact that an agent that is connected to one server downloading a document cannot attend to the coordination request until it finishes and closes on the present server [5].

5.2.3 Which bids to accept?

This decision procedure corresponds to the AiC's selection of bids (equation (4.8)) from the total set of bids received ($\mathcal{B}$). In the formulation presented below, the

[4]Once again, the AiC cannot estimate in advance the number of helpers. This is because it does not have a way of knowing the number of agents that are interested in the same document or how many of them will agree to share the cost of downloading. However, it is possible to follow the same solution discussed in the transportation domain.

[5]It is important to notice that in the abstract scenario only AiSs answer coordination requests. However, in this domain it is clear that AiCoops or AiCs might also respond. The importance of having agents that reconsider their commitments to their current tasks to attend to more profitable ones (in this domain, that which might reduce the costs) is discussed further in Chapter 8.

surplus (surplus) becomes the total cost of downloading (def_cost) a given document. It is interesting to notice that in this domain, in principle, the more bids an agent receives, the better off it is likely to be. This is because all bids contribute to reducing the total cost of downloading. However, consider the situation in which the AiC waits for the furthest agent. This waiting time is likely to increase its total cost rather than reduce it. Consequently, the AiC has to balance the waiting time of the AiCoop and the amount it gains with the bid. Thus, it calculates the reward of the subset of bids (reward_bid instead of cost_bid) so that it minimises the cost spent in waiting time.

In more detail, let b be the subset of $\mathcal{B}$ (excluding the empty set). From this, AiC approximates the reward from a subset of $\mathcal{B}$: $\mathsf{reward\_bid}_b = \sum_{(\mathsf{bid}_{ij}, T_i) \in b} \mathsf{bid}_{ij}$. Then it finds the furthest bid from that subset: $maxT_b = \max_{(\mathsf{bid}_{ij}, T_i) \in b}[T_i]$. Finally, it estimates the definitive cost to be paid based on the furthest bid to arrive and its own investment. The subset of bids selected is the one that minimises def_cost:

$$\mathsf{def\_cost}_{ij} = p_j \times R - \mathsf{reward\_bid}_b + r \times (t_j + maxT_b)$$

Note that the estimation of reward_bid is similar to the formulation described by equation (4.10), in which cost_bid is the sum of the m bids needed, but in this case, it is based on the cardinality of the set b. An additional observation is that def_cost increases with the waiting time and decreases with the reward of the set of bids.

5.3 Discussion

This chapter described two different application domains to show that the concepts of the decision making framework of Chapter 4 are not specific to the grid world scenario outlined in Chapter 3. With minor changes, it was shown how the key constituent factors of the framework were mapped into the application domains of transportation and coordinated information retrieval (this helps address some of the concerns in Section 3.2).

Furthermore, this endeavour highlights the fact that the scenario and framework introduced in Chapters 3 and 4 portrays and describes the key coordinating processes that can be found in concrete applications domains (as well as the more generic testbed). The agent interactions described in this chapter clearly show the need for a degree of flexibility in the decision making with respect to coordination. To this end, however, it is also evident that the basic model has a number of limitations. For example, the necessity of dropping commitments and paying sanctions were present in both domains and yet these are not considered in the initial version of the framework. This is because it was felt that that it is initially important to highlight the fact that the framework's basic assumptions were maintained, and, more importantly, that all the framework's constituent factors found a correspondence in the more realistic domains. To this end, having incorporated the main factors into the formulations of Sections 5.1 and 5.2 means that the framework does indeed have a broader applicability. Most probably, depending on the domain, some additional concepts (the truck space for example in the transportation domain) will need to be incorporated or modelled more precisely (in the coordinated retrieval information domain, the use of real time could have been employed instead of the routing connection). However, those that have already introduced represent the major ones in which many examples can be mapped and tested.

Another important observation from both exemplar applications is that agents need to base their decisions about coordination on predicted information about the environment and about the other agents. For example, in both application domains, the number of possible agents expected to participate in the coordination activity is not specified in advance. Hence, the prediction of the expected reward to gain from others is imprecise. In general, the use of definitive information when agents take the decision of engagement or not in coordination activities is indispensable; the more precise the predictions, the better the decision about coordination. However, in open scenarios, for example, where agents can enter and leave at different times, the estimation of these factors is extremely challenging or when the cost of making a good prediction is expensive it is sometimes better to have a reasonable approximation. This issue is examined in more depth in Chapter 9.

Chapter 6

Evaluation Methodology

In order to employ a formal and systematic evaluation of the work in this thesis, a set of experiments has been designed to evaluate and measure the agent's performance. In particular, statistical inference methods are used to generalise conclusions from samples of data. The underlying idea consists of assessing a system by verifying hypotheses over the data. There are several statistical procedures to perform this task (including analysis of variance (ANOVA), regression and path analysis) each of which represents an alternative method for analysing the sample. For purposes of this work, however, ANOVA is used because it is a general and a robust technique: *"analysis of variance is very robust against violations of the normality and equal variance assumptions, especially if the group sizes are equal."* (Cohen, 1995, pp. 194).

In concrete terms, designing an experiment in this thesis consists of identifying the variables of interest (hypothesis formation), making observations of those variables after the application of a particular process, testing the significance of the observations and revising, accepting or rejecting the hypothesis. Additionally, it is necessary to establish the periodicity with which the data will be collected and the frequency with which the experiments will run. Thus, designing an experiment is categorised by the following items that will be dealt in turn.

1. Experimental procedure,

2. Simulation or dependent variables,

3. Experimental or independent variables and

4. Hypothesis formation

Experimental procedure. This describes in detail the procedure for performing and analysing the experiments. In the context of this thesis, the experimental procedure captures the agent's reasoning which varies from modelling a new feature (e.g., whether an agent possesses learning abilities or not), changing a particular process (e.g., when agents use alternative decision making processes for bidding) or introducing a new variable or parameter (e.g., when the decision making procedures incorporate a new factor in their formulation). In particular, the main interest in this case is the observation of the procedures that model the agent's decision making about the dynamic selection of coordination mechanisms.

Simulation variables. These summarise the attributes and properties of the environment in which the experimental procedure is tested. The simulation variables are given specific values to define a particular instantiation of a scenario. Thus, it could be said that the domain in which the simulation's variables take their values generates all possible scenarios. Additionally, these variables can be thought of as the static (because once the value is set, it remains unchanged in a particular experiment) and dependent variables. In this context, there are two kinds of simulation variables, those related with the scenario itself and those related to the agent reasoning. Examples of the former are the total number of time steps (duration time) the experiment is going to run, the grid size and the reward associated to the CT. Examples of the latter include the specific characteristics of the CM, the alternative ways in which the decision making procedures are used and the elements taken into consideration in such decisions.

In general terms, there are two main classes of experiments used in this thesis to validate hypotheses. The first class corresponds to those experiments that provide evidence of how a particular procedure behaves in a variety of situations and is achieved by fixing the experimental procedure of interest and changing the scenarios (a combination of simulation variables). The second class are those that indicate the benefits of alternative agent behaviours and these are accomplished by having a scenario with fixed features and varying the experimental procedure. For instance, testing whether an agent that dynamically selects its CM performs the best in all situations is an example of the former class of experiment (the agent's behaviour is fixed and the grid sizes and CT reward are explored). In contrast, probing whether a learning agent performs better than a non-learning one in a specific scenario is an example of the latter (here the agent's behaviour corresponds to a learning and a non-learning agent and there is one set environment).

In more detail, Table 6.1 presents the simulation variables and their assigned values for most of the experiments used in this thesis. Unless otherwise specified, these values are used in all the hypothesis tested in this thesis.

Simulation Variable	**Value**
Number of simulation runs over which the data is collected.	10
Duration of time. Number of time steps in a given simulation run (horizon)	10,000
Size of Grid $[N \times M]$	$[10 \times 10]$
Number of agents in the environment	5
Number of Cooperative Tasks in the grid at any one time	1
Maximum number of agents needed to achieve a CT (m)	3
Reward of a single CT instance (R)	20
Reward of single ST instance (S)	1
Number of CMs an agent knows about	5
Probability of success and number of time steps to set up in a given CM=(t, p)	CM_1=(0,0.6) CM_2=(15,0.7) CM_3=(30,0.8) CM_4=(45,0.9) CM_5=(60,1.0)
Willingness factor for a single agent (ω)	1.0

Table 6.1: Simulation Variables

On the basis of these variables, it is important to note that the main conclusions regarding the efficacy of the dynamic selection of CMs (see Chapter 7) test the whole range of possible settings for the key simulation variables. The values introduced here simply indicate the situation in which the experimental procedures are tested in a fixed environment. In such cases, some assignments need additional explanation. For instance, to set the duration variable, a utility rate (the agents' total reward divided by the horizon) was calculated for the experiment where the duration was varied from 10,000 to 100,000 time units. The statistical measures of the utility rate in this case had a standard deviation of 2.10E-02 and a variance of 8.11E-04. Given this result, it was concluded that the duration of the experiment in this range does not have a significant effect on the rate of utility since regardless of the horizon, the utility rate is maintained. Therefore the lower level was used since this meant the experiments could be conducted more quickly. The same result was obtained with ANOVA; in this test, the hypothesis of equal means of the utility rate given alternative duration times was accepted. Although most of the fundamental exploration regarding the variable setting was performed using the methodology described here, those results are not reported because this thesis only describes the experiments and hypothesis used to make relevant conclusions. With respect to the CMs, these were chosen to be consistent with the observations that there is no best CM and the CMs more likely to succeed take longer to set up (see Chapter 1).

Experimental Variables. These describe the data of interest; that is, the data that should be examined and measured from the scenario. In the context of this thesis, this means focusing on the specific variables that show the agent's performance. These are generally concerned with measuring how much an agent's individual utility is improved by an enhancement of the model [1]. To this end, the main experimental variables are specified in Table 6.2 and, as can be seen, these measure how much the agents' performance improves depending on a specific environment or a refined ability. All experimental variables are calculated by averaging the totals obtained by all simulations and by all agents in the system. Thus, for example, the TCT achieved is the average number of CT tasks achieved by the agents in the system.

Experimental Variable	**Variable Description**
AU	Total Agent Reward obtained from the accomplishment of its ST and CT tasks (termed Agent Utility).
TCT	Total number of CTs accomplished by an agent.
AiS	Total agent reward obtained by agents in the Agent-in-ST role.
AiC	Total agent reward obtained by agents in the Agent-in-Charge role.
AiCoop	Total agent reward obtained by agents in the Agent-in-Cooperation role.

Table 6.2: Experimental Variables

Hypothesis formation. This consists of identifying a claim to direct the experimentation. Hypotheses are formulated based on the experimental variables to test the execution of a particular experimental procedure under a particular

[1]Notice that some experimental variables are only used in some of the experimental procedures. For instance, in the case of flexible commitment varying penalties (Chapter 8) it is necessary to introduce new simulation variables and, correspondingly, new experimental variables are needed.

environment. Recalling the examples of the two classes of experiments, the first one could test the hypothesis of whether the TCT (experimental variable) is constant when agents dynamically select the CMs in all possible scenarios (varying the simulation variables of grid sizes (simulation variable $[N \times M]$, CT reward, etc.)). On the other hand, the example of the second class of experiments could verify whether the AU (experimental variable) obtained by learning agents is the same as that obtained by non-learning ones (alternative experimental procedures) in a fixed environment (fixed setting of simulation variables).

6.1 Evaluating hypotheses

Given the above, designing an experiment consists of formulating and testing hypotheses taking into account data collected (from experimental variables) as the result of the application of the experimental procedures to particular scenarios (combination of values of simulation variables). In other words, the execution of an experimental procedure in a specific scenario generates a set of values for the experimental variables that can be analysed under the same circumstances and situations in order to probe hypotheses.

The basic idea of testing a hypothesis consists of comparing the means (from the experimental variables of the sample) and supporting a hypothesis by a certain confidence level. Formally speaking, the procedure is the following (Cohen, 1995; Ott & Mendenhall, 1995; Lane, 2001):

1. Formulate a hypothesis (termed the null hypothesis and represented by H0)
2. Show that the probability (p) of obtaining a given result given H0 is above a certain threshold (this threshold is known as the significance level and 0.05 is the standard measure associated to it [2])
3. Conclude H0 (with a percentage of confidence level)

By means of an example, assume that the hypothesis to test that the AU obtained by procedures A and B are the same (to clarify the description think of

[2] The significance level is used to compute the confidence level. This confidence level equals 100*(1-significance level), or in other words, a significance level of 0.05 indicates a 95 percent of confidence level.

A as the procedure that represents a learning agent and B a non learning one). Thus, testing the hypothesis consists of the following steps:

1. H0: AU_A=AU_B [3].

2. Assume that the result of the statistical test applied is p = 0.980 (using a significance level of 0.05)

3. Hypothesis H0 is then accepted and it can be concluded that H0 is certain with a 95% of confidence level. Being more precise, it can be said that learning and non learning agents perfom the same with a 95% confidence level.

Following the above example, the null hypothesis is defined (step 1) and tested with ANOVA (step 2). If it reveals that the differences among means are significant (the value of p is less than 0.05) then the hypothesis of equal means is rejected or, in the contrary case, the hypothesis is accepted. That is, if the hypothesis is accepted (step 3), it means that there is no evidence to accept the proposition that the experimental procedure (A or B) has any effect on the independent variables (AU).

ANOVA explains the relationships between groups by analysing all possible interactions among them. However, though it provides an answer to the hypothetical questions by indicating if the mean of the groups are equal or not, it does not indicate which groups are better (for example, if $\mathsf{AU}_\mathsf{A} > \mathsf{AU}_\mathsf{B}$). Thus, in most cases, it is necessary to go a step further (post-analysis) to determine where the exact differences among the means occur between groups. This procedure consists of running a post-test to explore the data collected on a case by case basis (this is termed pairwise analysis) because it tests the difference between each pair of means [4]. This pairwise analysis is particularly important in those cases where more than two procedures are being tested (or one procedure is tested in more than two scenarios). For example, to test whether A, B, C and D perform the same, H0 is rejected if the equality of means is not maintained. However, this says nothing about how A compares with B, C and D, nor B with A, C and D,

[3]Throughout this thesis the convention of using the element of comparison as the subindex is followed. In this example, A and B represent the experimental procedures.

[4]Several statistical tests exist to perform this analysis. The one used here is called Tukey's honestly significant difference (HSD). This was chosen because it lies in the middle of the spectrum of alternatives; between LSD (which stands for *least significant differences*) and Scheffé tests which are the extreme in the conservative methods (Cohen, 1995; Lane, 2001).

and so on. In concrete terms, the post-test makes a comparison between the data collected and builds groups (as many as necessary) that have statistically homogeneous values. Each group is generated with an associated value (the p value) that indicates the degree of confidence from which each group was built (the higher the number (in a range of 0.0 to 1.0), the greater the confidence in the grouping). For example, Table 6.4 shows the result of this cluster analysis. Here, three groups were generated (labelled 1,2 and 3) and the respective p values were 1.000, 0.8728 and 0.9959.

6.2 Example: Testing hypotheses

To clarify the methodology, assume there are five agents (A_1, A_2, A_3, A_4 and A_5), each of which has at its disposal different CMs (i.e. A_1 has CM_1, A_2 has CM_2, and so on). The agent's performance is evaluated by the experimental variable AU in a specific environment defined by CT reward of 10 and a grid size of $[5 \times 5]$ with the following hypothesis formulation:

H0: the AU obtained by A_1 is the same as that obtained by A_2, A_3, A_4 and A_5 in a given environment.

CT reward=10 $[M \times N]$=$[5 \times 5]$		
Hypothesis to evaluate	**p**	**Outcome**
H0: $\mathsf{AU}_{\mathsf{A}_1}$=$\mathsf{AU}_{\mathsf{A}_2}$=$\mathsf{AU}_{\mathsf{A}_3}$=$\mathsf{AU}_{\mathsf{A}_4}$=$\mathsf{AU}_{\mathsf{A}_5}$	0.000	Rejected

Table 6.3: Example: result of ANOVA.

The result of ANOVA (Table 6.3) shows a significant effect on the AU given the environment (p value is 0.000 and H0 is rejected). This result means that the agents' performance does indeed have a statistically significant effect on the AU obtained. However, apart from knowing that the hypothesis is rejected it is not possible to know about how the various kinds of agents compare relative to each other. Thus, Table 6.4 shows the groups formed by the post-analysis test. Here, three groups were formed and the agents in each group can be regarded as having broadly the same level of performance. However, each group indicates that though agents perform more or less the same, the data obtained by each group has a statistically significant difference from the others. Additionally, by observing the results, the agents with better results (which can be said to have a dominant performance [5]) are A_3 and A_1. The second best performing group is A_4 and A_5 and the worst performance was obtained by A_2. In general, such a full justification needs to accompany each hypothesis test. Thus, in this case, it is necessary to explain why A_1 and A_3 have similar performance characteristics and why these lead to them to be more successful than the next group of agents.

Agent	**AU**		
	1	2	3
A_2	3266.3		
A_5		3337.7	
A_4		3353.3	
A_1			3794.3
A_3			3800.3
p	1.000	0.8728	0.9959

Table 6.4: Example: post-analysis of ANOVA

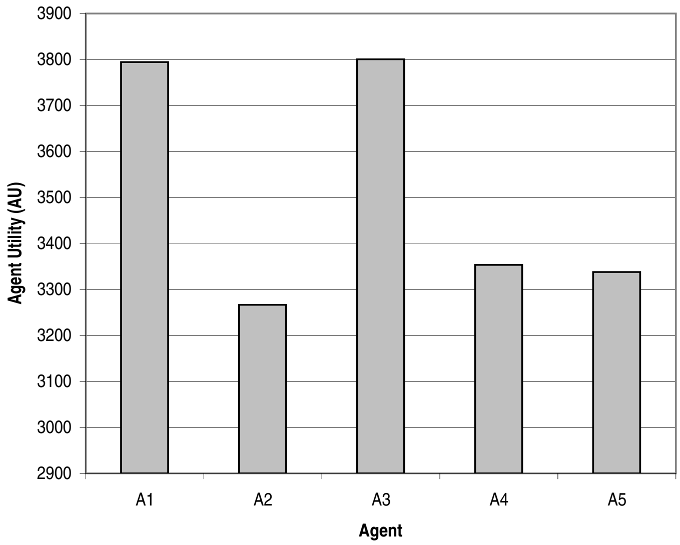


Figure 6.1: Example: Agents' performance.

Although Table 6.4 clarifies the results obtained with ANOVA, in some experiments in this thesis, figures and graphs are used instead of this post-analysis. The reason for this is because in these cases the figures more clearly present the results obtained. For example, Figure 6.1 shows the same information as that of Table 6.4 (but using a bar graph). It is important to emphasise the fact that graphs might be used to make conclusions (for example, to locate the agents with best performance) in a graphical way (as a replacement of the post-analysis) not

[5] Dominant behaviour in the context of this thesis means that one agent has an AU value that is higher by a statistically significant amount than the others.

as a substitute for ANOVA. The problem of using graphs as the only mechanism to draw conclusions is that it is not straightforward to detect all the details obtained with the post-analysis and, more importantly, it is not obvious whether the hypothesis might be rejected or accepted. For example, by looking at Figure 6.1, it is not graphically obvious that there are three groups with different levels of performance (as indicated by Table 6.4). In short, ANOVA evaluates the hypotheses and provides statistical measures with a degree of confidence about rejecting or accepting the hypothesis, and after that a post-analysis or a graph is used to understand and justify the results.

Moreover, to facilitate the task of testing hypothesis, a statistical software, SPSS for Windows [6], has been used to undertake all the fundamental explorations and to draw the conclusions reported in this research. This is particularly important in the scenario used in this thesis because the number of simulation and experimental variables represented by environmental factors and the agent's reasoning features are large.

[6]SPSS Inc. Chicago, Illinois (1999). *SPSS for Windows release 10.5.5.*

Chapter 7

Decision Making Evaluation

Having presented the basic formal framework in Chapter 4, this chapter deals with its evaluation according to the methodology described in Chapter 6. In particular, this chapter concentrates predominantly on the fundamental hypothesis underlying this thesis; namely that being able to select the CM at run-time is beneficial. It also explores the impact of the model's main parameters on the performance of the individual agents. Subsequent chapters (8 and 9 respectively) deal with the evaluation of extensions to the basic model with respect to flexible commitments and penalties and with respect to learning. Additional experimentation focusing on several heterogeneous aspects and their impact on the agent's performance is detailed in Appendix A.

7.1 Experimental setting

The experiments were set-up using the values specified in Table 6.1. The experimental variables were the size of the grid and the reward for CTs. The variables that measure the agent's effectiveness are: AU and TCT. Moreover, in some experiments it is interesting to observe the reward an agent obtains in each of its different roles: AiS, AiC and AiCoop (see Table 6.2 to recall the description of the experimental variables).

7.2 Selecting different CMs

The first thing to test is that agents do indeed select different CMs in different circumstances. To this end, Figure 7.1 shows which CMs were selected in which grid position. Here, the grid size was $[20 \times 20]$ (the remaining three quadrants are simply a mirror of the upper left portion shown) and the reward for CTs is 10. In the centre of the grid, $[10, 10]$, the agents choose CMs that minimise the set up cost (even though they have a significant chance of failing to ensure coordination). However, as the agents move further away from the centre, so they increasingly prefer mechanisms that are more likely to succeed (even though they have a correspondingly higher set up cost). The explanation for this behaviour is that as the distance from the centre increases, so does the expected time for another agent to reach the CT square. Thus, to justify its choice of a CT over its ST, the AiC needs to ensure that the cooperations it does enter into do succeed. Whereas, towards the centre of the grid, the time the AiC typically has to wait for another agent to arrive is much smaller and so it can afford to have more cooperations fail. In between are the points where success and set up time are traded off. Figure 7.2 shows the corresponding expected utility for the various CMs (ave_surplus, equation (4.5)). Notice that the CM's surplus expectation decreases as the agent moves away from the center of the grid.

y↓ \ x→	1	2	3	4	5	6	7	8	9	10
1	5	5	5	5	5	5	5	5	5	5
2	5	5	5	5	5	5	4	4	4	4
3	5	5	5	5	4	4	4	4	4	4
4	5	5	5	4	4	4	3	3	3	3
5	5	5	4	4	4	3	3	3	3	3
6	5	5	4	4	3	3	3	2	2	2
7	5	4	4	3	3	3	2	2	2	2
8	5	4	4	3	3	2	2	2	2	2
9	5	4	4	3	3	2	2	2	1	1
10	5	4	4	3	3	2	2	2	1	1

Figure 7.1: Terrain map showing where the various CMs are selected.

$y \downarrow$										
1	2.20	2.41	2.59	2.74	2.88	2.99	3.08	3.15	3.20	3.22
2	2.41	2.61	2.79	2.95	3.08	3.20	3.29	3.36	3.41	3.43
3	2.59	2.79	2.97	3.13	3.26	3.38	3.48	3.55	3.60	3.63
4	2.74	2.95	3.13	3.29	3.43	3.55	3.65	3.73	3.78	3.81
5	2.88	3.08	3.26	3.43	3.58	3.70	3.81	3.89	3.94	3.97
6	2.99	3.30	3.38	3.55	3.70	3.84	3.94	4.02	4.08	4.11
7	3.08	3.29	3.48	3.65	3.81	3.94	4.05	4.14	4.20	4.23
8	3.15	3.36	3.55	3.73	3.89	4.02	4.14	4.23	4.28	4.31
9	3.20	3.41	3.60	3.78	3.94	4.08	4.20	4.28	4.34	4.38
10	3.22	3.43	3.63	3.81	3.97	4.11	4.23	4.31	4.38	4.41
$x \rightarrow$	1	2	3	4	5	6	7	8	9	10

Figure 7.2: CM's expected utility in the terrain map.

7.3 Amount of cooperation

To measure the number of times that coordination is attempted, Figure 7.3 shows the number of cooperative tasks achieved (TCT) by an agent as a function of the reward for achieving a CT and the grid size. The figure shows that once the reward for CTs is sufficiently high (above 4 in this case) then CTs start getting initiated (i.e. this value is needed before ave_surplus becomes positive). At the same time, the TCT increases as the grid size decreases because in small grids agents simply have more chance of finding the CT.

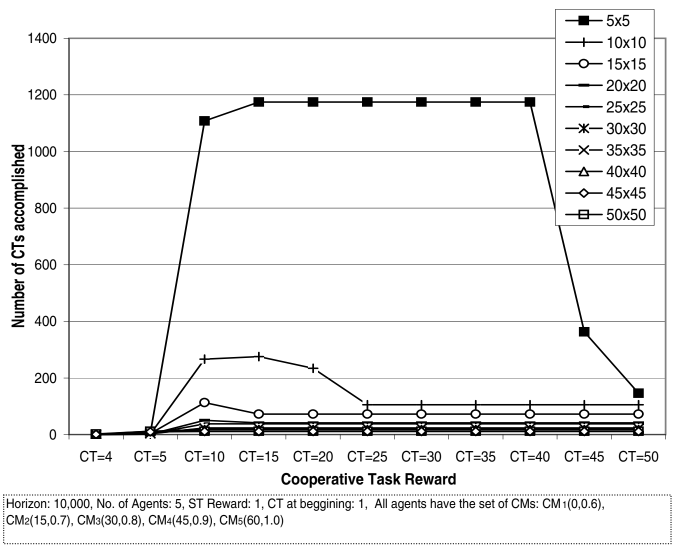


Figure 7.3: TCT achieved per agent

However, it is important to notice that the combination of size of grid and CT reward does have an implication on the selection of the CM and, consequently, on the number of CTs achieved. In the [5 × 5] grid, for example, the TCT declines when the CT reward is higher than 40. This is because although the agents initiate coordination in exactly the same circumstances, they select a CM that takes more time to set up [1]. The reason for this behaviour is that since the reward is so high, the agents select a CM that guarantees success, regardless of the time invested. As they spend more time establishing coordination, they have less opportunity to find CTs and so with a fixed time horizon they achieve fewer CTs. Similar behaviour is observed in the other grid sizes, although the CT reward level at which coordination starts and the level at which the number of CTs achieved starts to fall obviously varies. While Figure 7.3 indicates how often the agents cooperated, Figure 7.4 shows how profitable those decisions are. It is clear that the AU decreases as the grid size increases. Again this is simply because the agents have less opportunity to engage in CTs. Following the same example of the [5 × 5] grid, it is observed that the agent's utility stops increasing when the CT reward is above 40. Once again, this is explained by the fact that agents do not engage in coordination as often as they do with smaller values of CT reward.

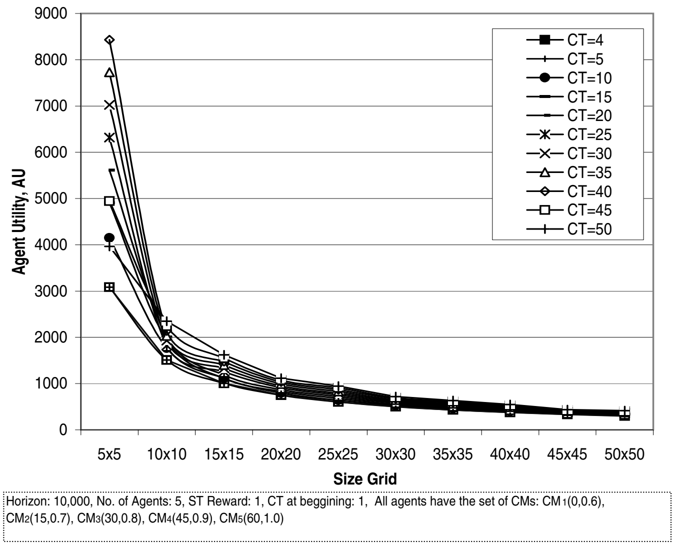


Figure 7.4: Agent Utility, AU

[1]Being precise, agents in this situation selected the five CMs in the following grid positions:

y↓					
1	4	3	2	3	4
2	3	1	1	1	3
3	2	1	1	1	2
4	3	1	1	1	3
5	4	3	2	3	4
x→	1	2	3	4	5

Finally, to have a better understanding of the model as a whole it is important to generalise the results discussed regarding the selection of a CM. To do this, the best performance of an agent that has only a single CM at its disposal is compared with that of an agent that has the set of CMs at its disposal. First, the best performing CM in a given environment needs to be determined. This is achieved by giving each of the agents in the system a single CM (the same one for each agent) and measuring the average AU. This is repeated for each CM. Then, the system is run with each agent having the full set of CM at their disposal. The arithmetic difference between the best performing single CM agent and the agent with the full set of CMs is then calculated. Here, this difference is termed the *error rate*. Now in environments where there is a higher error rate (see Figure 7.5 which shows the error distribution per grid size and CT reward [2]) are those in which it is most likely that there is a single agent with a particular CM which obtains better AU than that obtained by the agents with the whole set of CMs at their disposal. In precise terms, an agent with a single CM obtained a better AU in the environments with higher error rates values. Thus, higher discrepancies are situated in smaller grids and when there are higher CT rewards. These observations corroborate the results illustrated in Figures 7.3 and 7.4 where the zones in which AU drastically declined are the ones with higher error rates. Less marked differences in others zones of the same figures are again validated by the error rates values.

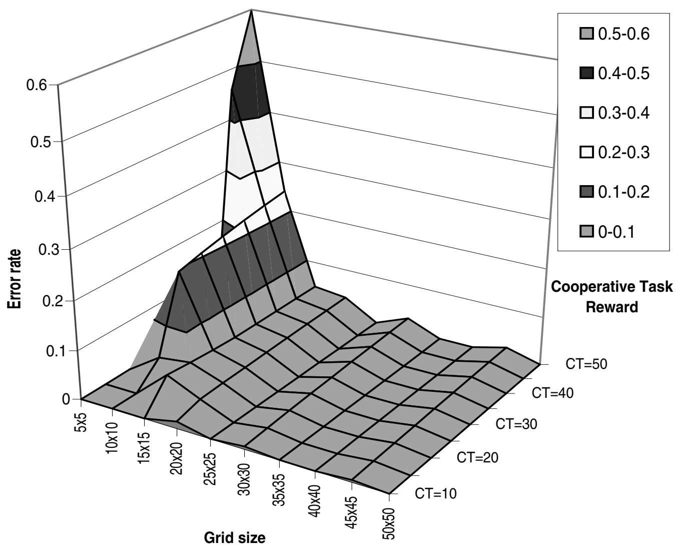


Figure 7.5: Cases in which dynamic selection is more effective.

[2]Given the results obtained, higher error rates are above 0.3 and lower ones are below this value.

7.4 Willingness to cooperate

The final key determinant of the amount of cooperation that occurs in the system is the WtC factor (experimental variable) [3]. To this end, Figure 7.6 shows the effect of this factor (the reward for CTs is here taken to be 10 and the grid size is [20×20]) on the cooperative tasks accomplished. As expected, the more greedy the agents become (increasing ω), the fewer CTs that get initiated and achieved. The same figure also illustrates the relation between the TCT achieved and the agent's reward. In this scenario, the more CTs that are accomplished, the more AU that is gained. However, this hides the fact that the various constituents that go towards making AU vary in their relative importance as ω changes. This is because ω affects the amount asked to become an AiCoop (equation (4.6)) and, consequently, the surplus available for the AiC (equation (4.5)). Thus, for instance, the more selfless an agent becomes, the more reward is gained through the AiC role (the percentage of reward obtained by the AiC and AiCoop roles with $\omega = 0.25$ was 97% and 3% respectively). On the other hand, the more greedy the agents become, the higher the percentage that is obtained by being AiCoop (with $\omega = 3.0$, AiC received 69% of the reward gained through cooperative situations and 31% through being AiCoop).

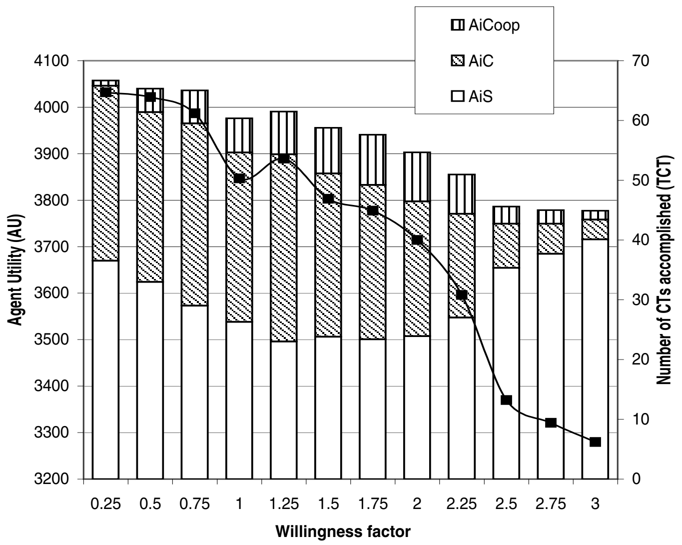


Figure 7.6: Willingness to Cooperate (ω).

[3] Appendix A explores the effect of having agents with different dispositions to cooperation (varying ω factor) in the same simulation run.

7.5 Effectiveness of the agent's decision making

Having analysed the effect of the decision making framework's basic parameters, it is now time to consider the impact of being able to dynamically select a CM that is deemed to be appropriate to the prevailing circumstances. This time the performance of an agent that employs a single CM is contrasted with the agent that has the full range of CMs at its disposal. Note that in this experiment the key element is that agents with different CMs share the environment in a given simulation run [4]. Thus, A_1 only has the $CM_1(0, 0.6)$ to select from, A_2 the $CM_2(15, 0.7)$, A_4 the $CM_4(45, 0.9)$, A_5 the $CM_5(60, 1.0)$, and A_S the whole set of CMs [5].

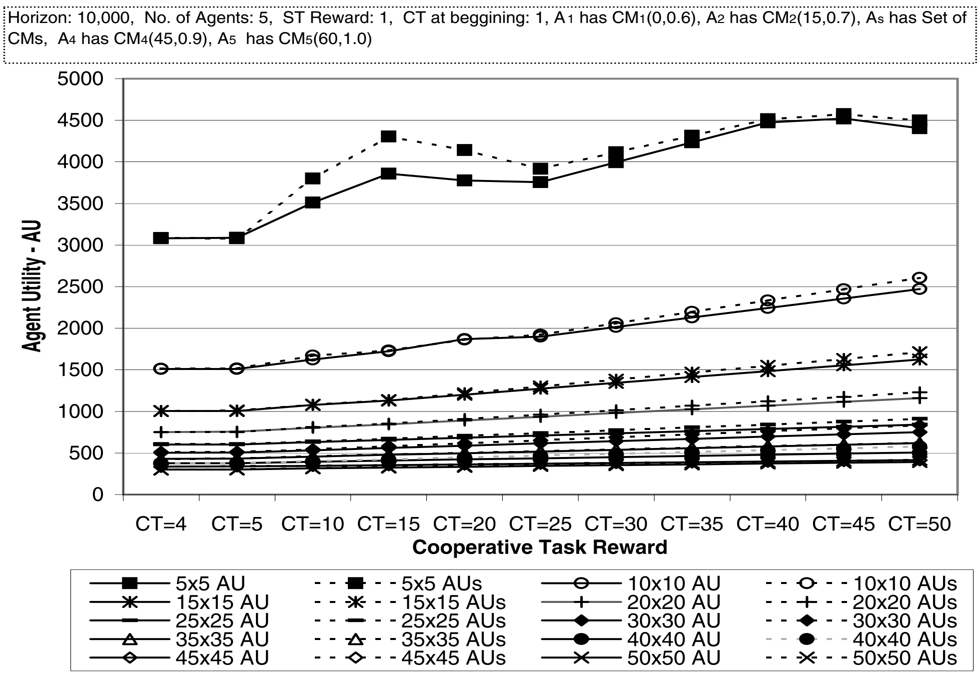


Figure 7.7: Agent Utility: Dynamic versus static selection of CMs.

Figure 7.7 presents the average utility of agents in the environment (AU) and that obtained by A_S (AU_S) for varying grid sizes and CT's reward. In general, it can

[4]In the comparison performed at the end of Section 7.3, in a given simulation run, all the agents had the same CM at their disposal. Thus, for example, all the agents had only CM_1 or only CM_2, whereas in this scenario one agent has CM_1, another has CM_2 and so on.

[5]Note that $CM_3(30, 0.8)$ was not associated to any agent. This is because it was decided to maintain a constant number of agents in the environment (for reasons of comparison) rather than change the experimental settings. CM_3 was omitted because it lies in the middle of the range. However, the same experimentation as described here was performed with each of the CMs missing. Although the number of cases rejected were different in each case, the general trends and conclusions remained unchanged.

be seen that A_S obtains higher utility than the corresponding average of the agents. This improvement is due to the fact that A_S not only attempts coordination more frequently, but that it also makes the right decision about which CM to select. Further support for this conclusion can be obtained by observing Figure 7.8. This shows the TCT achieved by the agents that have a single CM to select from and the agent that has the set of CMs (TCT_S). Here, in most cases, it can be seen that TCT_S is larger than the corresponding TCT. Moreover, in cases where A_S obtains a similar number of CTs to its counterpart (for example, CT reward of 40 and grid size of $[10 \times 10]$) the AU is higher (the AU_S represented by dotted lines are superior to the corresponding solid ones). Regarding the total amount of cooperation in this experiment, the TCT achieved is substantially lower than that reported in Section 7.3. This is because agents attempt coordination based on their particular CM and each of them starts coordinating in different places and with a different reward level for the CT. Thus the coordination activity is simply less frequent.

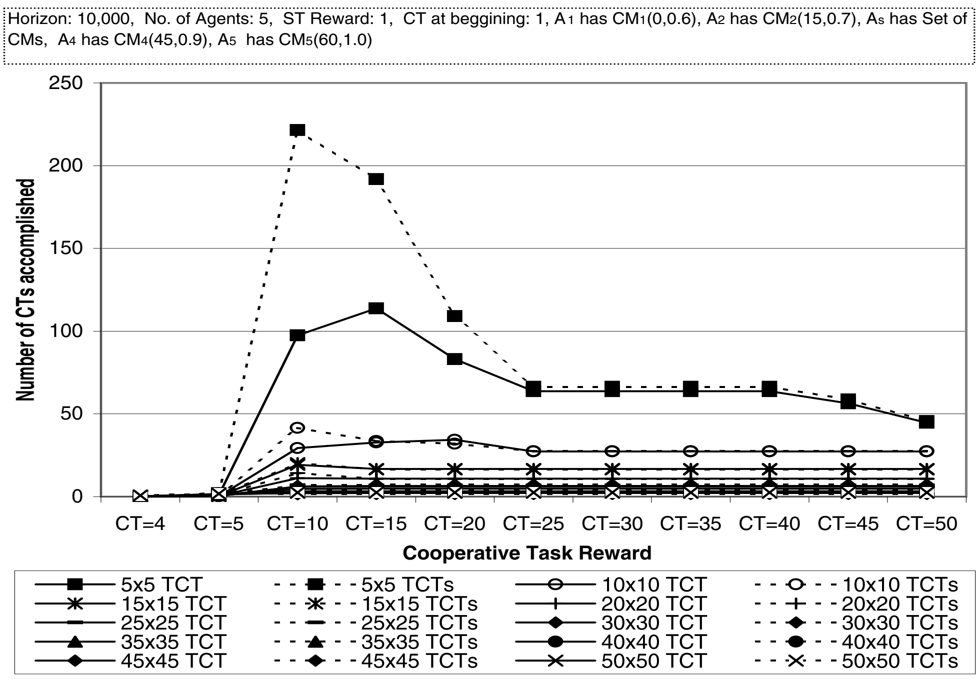


Figure 7.8: TCT achieved. Dynamic vs static selection of CMs.

Turning again to the agent's utility presented in Figure 7.7, AU_S does not appear to dominate the corresponding AU. This begs the question, does A_S's AU have a statistically significant dominance over that of the other agents? To answer this, and to analyse the benefits of A_S in detail, the evaluation methodology introduced in Chapter 6 is employed to test hypotheses. This time the evaluation focuses on the variable AU since this measures how much the agents' performance improves depending on a specific environment. Thus, to evaluate the claim about the benefits of dynamically selecting the CM, the following hypothesis (H0) needs to be tested for each agent, in each environment (defined by a specific CT reward and a grid size), using ANOVA.

H0: the AU obtained by A_S in a given environment is the same as that obtained by A_1, A_2, A_4 and A_5.

CT reward=10 $[M \times N]$=$[5 \times 5]$		
Hypothesis to evaluate	**p**	**Outcome**
H0: $\mathsf{AU}_{\mathsf{A}_1}=\mathsf{AU}_{\mathsf{A}_2}=\mathsf{AU}_{\mathsf{A}_s}=\mathsf{AU}_{\mathsf{A}_4}=\mathsf{AU}_{\mathsf{A}_5}$	0.000	Rejected

Table 7.1: Agent's performance: result of ANOVA.

To start with, Table 7.1 shows the result of evaluating H0 in an environment with a CT reward of 10 and a grid size of $[5 \times 5]$. Intuitively, it is expected that there will be a different level of performance from the agents that have only one CM to select from and the agent that has a set to select from. The ANOVA result is that the hypothesis is rejected ($p < 0.05$) meaning that the agents' performance does indeed have a statistically significant effect on the AU obtained. To justify this result, Table 7.2 shows the groups formed by the post-analysis. The first thing to notice is that A_S and A_1 are the agents that perform best (they have the highest AU values). This is because, in this environment, the less time that is invested in setting up a CM, the better (because the agents have more opportunity to find CTs). Both A_1 and A_S use the CM that takes the least time to set up (thus, A_S selects CM_1 all of the time).

However a good coordination decision maker has to balance the time invested in a CT through the CM selected and the final reward obtained. Thus, to validate this reasoning, it is important to know the total number of CTs achieved by each agent. This is shown in Figure 7.9. From this, it can be seen that A_1 and A_S did accomplish the most CTs and this corresponds with the reward they obtained. The second best performing group was A_4 and A_5 which have the CMs with the longest time to set up. However from Figure 7.9 it can be seen that they did not accomplish any CTs. Instead, they use their time accomplishing STs. This shows that the agent's decision making about when to attempt coordination (and when not to) is as important as selecting the right CM. Thus, A_4 and A_5 gained more AU by not attempting coordination than they would have done by attempting it. Finally, the worst performance was by A_2 which attempted coordination some of the time and achieved some CTs, but the reward it gained was not significant enough to make a difference in its AU [6].

Agent	**AU**		
	1	2	3
A_2	3266.3		
A_5		3337.7	
A_4		3353.3	
A_1			3794.3
A_S			3800.3
p	1.000	0.8728	0.9959

Table 7.2: Agent's performance: post-analysis.

In summary, the post-analysis indicates that the best performance is obtained by the agents which, on the one hand, gain the reward which justifies the time they invest on the cooperative tasks and, on the other, ensures that this reward is better than that they could have obtained by achieving STs alone and having no cooperative attitude.

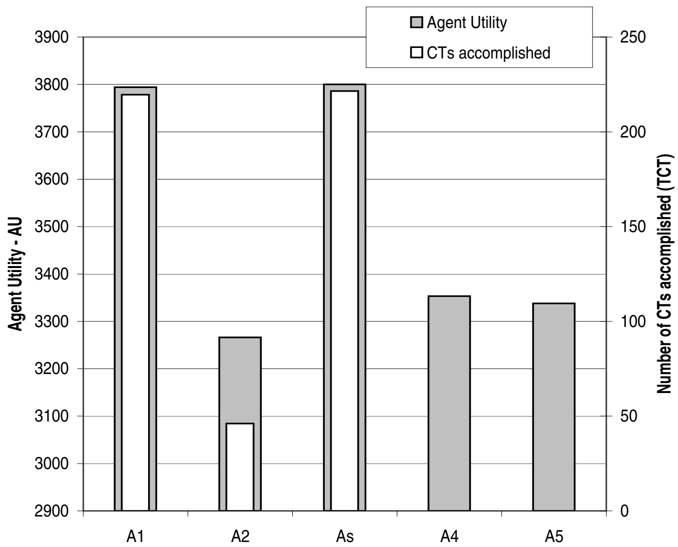


Figure 7.9: Dynamic selection of CMs.

Having shown the benefits of dynamically selecting CMs in one specific environment, the next step is to evaluate how this generalises to other environments.

[6]Notice that the agents' reward obtained is influenced by the probability of success of the CM which represents the final reward the agents obtain as a percentage of the CT reward. Thus, it can be seen that the number of CTs accomplished gives an indication of the agent's overall performance, but it does not necessarily follow that agents will be more productive if they accomplish more CTs.

In this case, 90 different environments are considered; these have CT rewards in the range of [10, 15, 20, 25, 30, 35, 40, 45, 50] (9 cases) and grid sizes in the range of [5×5], [10×10], [15×15], [20×20], [25×25], [30×30], [35×35], [40×40], [45×45], [50×50] (10 cases). To do this, the same statistical test is re-applied to each different environment. The premise is that the various single CM agents will perform well in different environments, but A_S will perform at least as well as (if not better than) the best of the others in all cases. Being more precise, the aim is to find those environments in which H0 is rejected [7] and then check using the post-analysis test in each environment whether A_S belongs to the group with the best performance (called the *winner group* hereafter).

The results are as follows. Firstly, there were 55 cases in which H0 was rejected and 35 cases in which it was accepted. This means that 61% of the time dynamically selecting CMs had a significant effect on the AU in a given environment. In more detail, Figure 7.10 (left section) shows the number of times that each agent belongs to the winner group. The results are grouped by CT reward (i.e. CT=10 represents the group of 10 environments (of different grid sizes) in which CT was 10). Figure 7.10 (right section) shows the same information in percentage terms. These results show that A_S obtained a statistically significant better performance than the other agents (61% of the time), A_5 did this 44% of the time and so on. This provides the evidence of the fact that A_S has the dominant behaviour over the other agents in most of the environments.

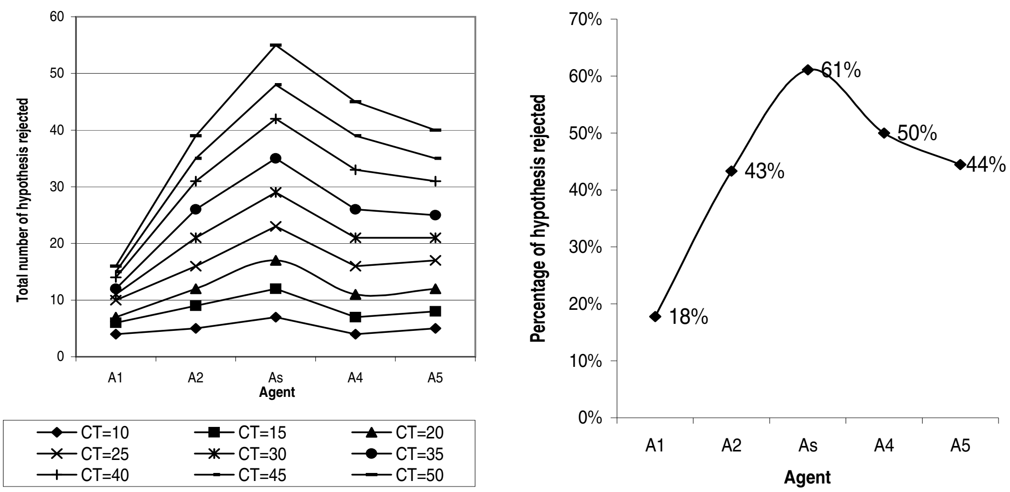


Figure 7.10: Cases in which the various agents' AU appeared in the winner group.

[7] Recall that the cases in which H0 was accepted are those environments in which there is no significant effect of the AU obtained by any agent.

While previous figures demonstrate A_S's dominance over the other agents, they do not show in which environments the different CMs are dominant. Again, taking the CT reward of 10 as an example, Table 7.3 presents the winner agents on a per environment basis [8]. From this it is clear that the CMs with lower times to set up (those associated with A_1 and A_2) are selected in smaller grids, while the CMs with higher values are selected in bigger grids. From the same table, it is also possible to observe the grid sizes in which the agents' selections do not have any effect on the AU (those in which H0 was accepted). This figure helps clearly illustrate (as previously discussed in Section 7.2) that the combination of CT reward and grid size has an effect on the agents' CM selection. Though the figure shows only a few of the environments, the same pattern is followed in the remaining environments. Thus, there are some environments in which some CMs are preferred over others and other environments in which there is no significant difference on the CM selection. However, what is more important is the fact that even though the situations in which some CMs are preferred over others (given the current constituent values of the CMs) are recognised, what has been shown is that A_S obtains a consistently better performance than the other agents.

CT reward = 10					
Grid size	A_1	A_2	A_S	A_4	A_5
[5 × 5]	1		1		
[10 × 10]	1	1	1		
[15 × 15]	1	1	1		1
[20 × 20]					
[25 × 25]					
[30 × 30]		1	1	1	1
[35 × 35]			1	1	1
[40 × 40]	1	1	1	1	1
[45 × 45]					
[50 × 50]		1	1	1	1

Table 7.3: Distribution of cases in which various agents belong to the winner group.

[8] Since the post-analysis generates groups with inferior and superior thresholds, it is possible to have agents belonging to more than one group (i.e. some AU values might belong to two groups). The results presented here take into account all the members of the winner group regardless of whether some of their AU values are closer to the superior limit of the subsequent group.

7.6 Discussion

The experimentation presented in this chapter confirms the motivating hypothesis of this thesis that agents do indeed perform better if they have the chance to select their CMs at run-time. However, some of the results show that agents do not always take the best decision about which CM to select. In Figure 7.3, for example, the agents' TCT drops when the CT reward is 45 or 50 in a $[5 \times 5]$ grid. This occurs because in some grid positions the agents chose a CM other than CM_1, which was selected for the lower values of CT reward [9]. This could be seen as the wrong choice because the agents would have achieved a higher number of CTs (around 1,170) by selecting CM_1 in all grid positions and they could have obtained greater reward (10,500 and 12,000 approximately) than they actually did. In such circumstances, it is emphasised that the agent's decisions are based on uncertain and estimated information about the environment and the other agents. In this particular case, when agents are at the edges of the grid, they assume that it is going to take a long time for the AiCoops to arrive or for them to gain enough reward to recover the investment (even though this is not the case in practice). To improve the effectiveness of the decision making framework, in accordance with the analysis of flexibility of Chapter 2, a number of extensions are proposed:

Deal more flexibly with commitments. In the basic model, once agents commit to a task they remain committed until that task has completed. To increase flexibility agents should be allowed to drop commitments if better opportunities present themselves and they should be able to deal with different types of sanctions associated with such decommits.

Endow agents with adaptive decision making abilities. In the basic model, the agents select the coordination mechanism solely on the basis of the expected surplus of each CM at its disposal. To enable agents to operate more effectively, they should be endowed with the capabilities of learning which CMs to select given specific environmental situations.

Allow agents to construct simple models of their collaboration context. In the basic model, agents make decisions based on their assumptions of other agents and the environment. If these assumptions can be made closer to the actuality of

[9] Recall the position in which the particular CMs were selected as illustrated in footnote 1 of this Chapter.

the other agents' behaviour, then the decision making should be improved. Thus, there is a need to explore the benefits of an agent learning, acquiring or refining the key factors of their collaborators upon which coordination decisions are taken.

To this end, Chapter 8 extends the basic model in terms of dealing more flexibly with commitments and penalties and Chapter 9 deals with making the decision model adaptive. Both chapters first outline their extensions to the basic model and then empirically evaluate their effect on the agents' performance.

Chapter 8

Flexible Commitments and Penalties

This chapter discusses the introduction of new decision procedures to deal with the dropping of contracts in order to better exploit new coordination opportunities. The motivating hypothesis is that enabling agents to dynamically set and re-assess both their degree of commitment to one another and the sanctions for decommitment according to their prevailing circumstances will make the coordination more effective (see the discussions in Sections 2.2.3 and 7.6). This hypothesis is evaluated, empirically, by considering agents that undertake actions with varying degrees of commitments and that have varying types of sanctions imposed whenever they renege.

The ability to renege upon commitments and to claim different types of redress impacts the decision making behaviour of both the AiCs and the AiCoops. In the former case, agents need to be able to attend to and recover from the situation when one of the AiCoops decommits. In the latter case, AiCoops have to assess opportunities to increase their utility by moving to more profitable CTs whenever they are found. To this end, to give maximum flexibility in coordination, agents require the ability to make agreements that involve different levels of commitment and different types of penalties (as motivated in Section 2.2.3). In particular, three types of commitment are considered [1]:

[1] These types of commitments can be considered to cover the full range of possibilities with respect to degree of commitment level. Total and Loose are the end points and Partial covers all possibilities inbetween. In particular, Partial commitment of 0% is equivalent to Total commitment and Partial commitment of 100% is equivalent to Loose commitment

- Total: Once an agent accepts a contract to achieve a CT, it cannot renege upon it (as per the basic model outlined in Chapter 4).
- Loose: An agent always drops a commitment if it finds a better option.
- Partial: Agents commit to achieve a CT, but with a percentage of probability they can drop this commitment if they find a better CT to pursue. For example, if an agent has a commitment level of 50%, then half the time it finds a better CT it will cancel and half the time it will continue with its current agreement.

Associated with the dropping of commitment is the use of a penalty model to compensate the agents that remain in the CT. Penalty payments are made each time an agent cancels a commitment and they are paid to the AiC (see Section 8.1 for details). Here the following types of penalty are analysed:

- Fixed: The amount is fixed at design time and is unrelated to the current coordination context. For example, whenever an agent decommits, it pays a fixed pre-established amount.
- Partially Sanctioned: The amount is specified in the contract when it is agreed. The actual fee depends on the state of the coordination activity and its participants, and the AiC's estimate of the profit that it will receive. For example, an agent could establish larger amounts in situations in which commitments are often dropped, whereas it could use lower values when agents remain committed most of the time.
- Sunk Cost: The amount is based on the effort that has been invested in the CT to date; thus, if the agents are close to achieving their goal they pay a higher fee than if the goal is some way off. This would mean, for example, that agents that have just started a coordination activity are more likely to drop commitments than those that have been working on the task for a long time.

In identifying whether a new CT is more beneficial than the current one, it is rational for AiCoops to include the decommitment penalty from their existing contract in their deliberations in order to realistically assess the new opportunities. Thus, a new CT may offer an intrinsically higher reward than the current one, but when the penalty is incorporated the agent may be better off sticking with its existing commitment. The protocol the agents follow (Figure 8.1) is an updated version of the one in Figure 3.1. There are two main differences (those differences are underlined in the referred figure). Firstly, AiCoops now reason about participating in new coordination activities when they find a CT (step [2] second part) and when they receive new requests for coordination when they are already engaged in a cooperative activity (step [3]). Secondly, the AiC now has to take corrective actions when contracts are dropped (step [4]).

[1] Agents arrive at a square. If AiS arrives at its ST cell, its goal is attained, it receives the reward and updates its goal. If AiCoop arrives at the CT cell, it notifies the AiC that it has arrived. It might have to wait in the cell until the remaining AiCoops arrive. If AiC receives confirmations from all AiCoops, the CT is achieved and the rewards are paid to AiCoops.

[2] If AiS finds a CT it must decide if it wants to become AiC and, if so, which CM= (t, p) it should use. If $t > 0$ it must wait t time-steps before broadcasting a request for coordination. <u>If AiCoop finds a new CT, it must decide if it should become AiC or continue with its present aim.</u> If AiC finds a new CT, it ignores it.

[3] If AiS or AiCoop receive a request for coordination, they decide whether and what to bid to participate in the CT. <u>If AiCoop decides to submit a bid, it factors in the penalty fee (if present) for dropping its current contract. The AiC then evaluates all bids.</u> If AiS's bid is accepted, it adopts CT as its new goal. <u>If AiCoop's bid is accepted, it drops its current contract (paying the associated penalty to the AiC) and becomes AiCoop of the new CT.</u> AiC does not respond to requests for coordination.

[4] <u>If AiC receives a decommitment message, it tries to find a replacement for the reneging agent by reproposing the CT. If it does not receive appropriate bids, it cancels the current CT by paying the contracted penalty to the remaining AiCoops.</u>

[5] Each agent decides on its next move according to its current goal and all agents move simultaneously.

Figure 8.1: Decommitment protocol followed by agents.

It is now necessary to extend the agents' decision making procedures to deal with varying commitment levels and penalties. In order to do this, the basic idea is to use a variable that in some way models the frequency with which decommitments might occur. From the AiC's point of view, if it receives a decommitment message it has to act to recover the ongoing coordination activity and ask for the penalty from the decommiter. If no recovery is possible, the worst case, the whole coordination activity is ruined. This generates loses for the AiC in terms of the time invested and for the other participants that do not receive the payment they were promised in their contracts. Thus, the first consequence of introducing a level of commitment is that an agent's expected rewards must be factored by this probability of failure. To this end, as an initial step toward having more refined methods and a better approximation of this probability of failure, the agent assumes that the others' attitudes to commitment are the same as its own (i.e. an agent that drops commitments very often will assume the rest do the same and, correspondingly, the possibility of receiving a reward will be lower) [2].

In what follows, let the current CT be subscripted by k and the potential new one by j. So, the probability of success (p_j given a particular CM_j) is affected by degree_commit [3]. Thus, when AiC decides to attempt coordination, equation (4.3) needs to be updated with the level of commitment in the following way [4]:

$$\mathsf{ave\_bid}_j(x,y) = \frac{r\_\mathrm{AiCoop} \times \mathsf{ave\_dev}(x,y)}{p_j \times \mathsf{degree\_commit}} \tag{8.1}$$

Equation (4.4) also needs to be modified because the agent needs to consider the total cost of decommitting. This involves the agent estimating the surplus it expects to obtain from adopting the CT, taking into account the probability of success of the task, the probability of commitment and discounting the decommitment payment, $\mathsf{decommit\_cost}_k$ (this includes the current contract value it expects to receive and the penalty it has to pay), from its existing CT_k, if there is one. Thus equation ave_payoff becomes:

$$\mathsf{ave\_payoff}_j(x,y) = p_j \times R \times \mathsf{degree\_commit} - \mathsf{decommit\_cost}_k \tag{8.2}$$

Following the same reasoning, the formulation employed for deciding how much to bid (equation (4.6)) and which bids to accept (equation (4.8)) need to be updated by the degree_commit and the cost of decommitment. Thus:

$$\mathsf{bid}_{ij} = \frac{r_i \times deviation_i}{p_j \times \mathsf{degree\_commit}} + \mathsf{decommit\_cost}_k \tag{8.3}$$

[2] This is a reasonable approximation given no additional information. This situation can clearly be improved upon by having the agents vary this perception in the light of their actual experience (see Section 9.2 for more details).

[3] This is represented as a percentage of probability that agents will renege during the course of the coordination episode.

[4] Agents are assumed to be neutral with respect to their willingness to cooperate (i.e. $\omega = 1$). Thus, to focus only on varying commitment levels and penalties, this factor has been removed from the subsequent equations in this chapter.

If the agent is AiS (rather than a AiCoop), bid_{ij} does not need to add the $\mathsf{decommit\_cost}_k$.

$$\mathsf{surplus}_{ij} = p_j \times R \times \mathsf{degree\_commit} - \mathsf{cost\_bid}_b - r \times (t_j + maxT) \tag{8.4}$$

In addition to these modifications, the agents now need to make decisions about two new situations: how to set a penalty fee (Section 8.1) and when to drop a commitment (Section 8.2).

8.1 Deciding how to set the penalty fee

The penalty fee can be set independently or dependently of the actual state of the coordination activity. In the former case, the fixed penalty is set as a pre-specified percentage (percentage_penalty) of the goal reward (the CT reward) over which the coordination activity is taking place (the actual percentage is an experimental variable that allows penalties to be high, medium or low).

$$\mathsf{penalty\_fixed}_j = \mathsf{percentage\_penalty} \times R$$

In the latter case, there are two possibilities for modelling the current state of coordination: by considering the surplus agreed in the contract (Partially Sanctioned penalty) and by incorporating a ratio of the time invested in the current CT (Sunk Cost penalty). For Partially Sanctioned penalties, the fee is based on the current expected surplus of the CT (again in a proportional manner). In this case, however, the proportion is set according to the degree of commitment within the group; if decommitment is likely, then a high penalty is set since the AiC needs to recoup its costs in setting up and running the group (mutatis mutandis when decommitment is unlikely).

$$\mathsf{penalty\_partially\_sanctioned}_j = (100 - \mathsf{degree\_commit}) \times \mathsf{surplus}_j$$

With Sunk Cost penalties, the fee is calculated with the percentage time_invested of the CT reward. In contrast to fixed penalties, this percentage is calculated as a ratio of the time spent on the CT and the time that the agent believes needs to be spent in order to complete the CT.

$$\mathsf{penalty\_sunk\_cost}_j = \mathsf{time\_invested} \times R$$

Partially Sanctioned and Sunk Cost penalties are both variable penalty cases because the sanction changes dynamically based on the state of the coordination activity, whereas the Fixed penalty represents a static penalty situation that is independent of current context of the coordination activity [5].

8.2 Deciding when to drop a commitment

The introduction of loose and partial commitments allows agents to consider the possibility of decommitting. This can occur in two situations:

- the agent may go from an AiCoop to an AiC or
- from an AiCoop on its current contract to an AiCoop on a new one (because the bid_{ij} it proposed was accepted).

Moreover, let $\mathsf{expected\_reward}_k$ represent the reward the agent is expecting to obtain from its current activity (this is calculated as the reward rate it will receive from the current goal). An AiCoop will drop its current contract to become an AiC on a new one if it finds a CT such that its expected average surplus is higher than the expected reward of its current contract [6]:

$$\mathsf{ave\_surplus}_j(x, y) > \mathsf{expected\_reward}_k$$

Similarly, an AiCoop will drop its current contract in favour of becoming the AiCoop on a new CT if:

$$\mathsf{bid}_{ij} > \mathsf{bid}_{ik}$$

It is important to emphasise that ave_surplus and bid_{ij} were calculated with equations (8.3) and (8.2) incorporating the decommitment cost.

[5] Note that for both variable penalties there are other possible ways to use the values of degree_commit and time_invested. For example, the ratio could be used to factorise either the CT Reward, the contract value or the expected surplus. However, since this is not the main aim of this research, only the selected methods are explored.

[6] Note that bid_{ij} is not used as means of comparison to maintain the same rate units.

8.3 Experimental evaluation

8.3.1 Experimental setting

The experiments were set-up using a single coordination mechanism CM=$(1, 1.0)$, the reward of the CT was randomly generated in a range of [20, 40, 60], 3 CTs were in the grid at any one time and the maximum number of agents needed to achieve a CT was 4. The changes in the environmental set up are such that the focus is on those aspects that are specifically related to commitments and penalties; everything else has been stripped out so that the results are not affected by extraneous factors. In particular, this setting does not incorporate a variety of coordination mechanisms because the aim is to concentrate on the particular decision making processes related to the commitment and penalties (given that the effectiveness of the dynamic selection has been demonstrated in Chapter 7). However, to incorporate more dynamic features in the environment, the CT reward (R) and number of agents needed (m) to achieve a CT are randomly changed during the execution.

As previously discussed, this new setting introduces new variables in order to analyse the percentage of commitment (degree_commit) and the percentage of fixed penalty (Fixed) which were tested in a range of 0, 25, 50, 75 and 100%. Correspondingly, the new experimental variables were the average penalty fee by penalty type, the number of contracts dropped (CTs decommitted) and how many of them were successfully recovered.

8.3.2 Results

The experiments seek to explore the following basic hypothesis: incorporating various levels of commitment and penalties into the coordination framework will improve the agent's effectiveness. The experiments address two main hypotheses that must be tested: those that probe the effect of the levels of commitment and those that deal with the effect of penalties. The former case is investigated first:

H1: The AU obtained by any agent role using Total commitment is the same as that obtained by agents that use Partial or Loose commitment.

H2: The number of CTs achieved (TCT) by agents using Total commitment is identical to that of agents that use Partial or Loose commitment.

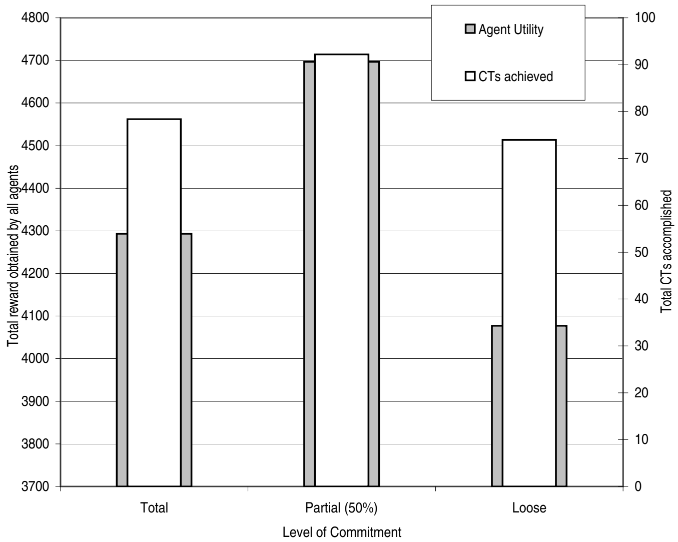


Figure 8.2: Reward distribution by agent role varying level of commitments.

Hypothesis to evaluate	**p**	**Outcome**
H1: $AU_{Total}=AU_{Partial50\%}=AU_{Loose}$	0.000	Rejected
H2: $TCT_{Total}=TCT_{Partial50\%}=TCT_{Loose}$	0.000	Rejected

Table 8.1: Comparing level of commitments: result of ANOVA.

Table 8.1 shows the results for agent utility and TCT achieved for the different levels of commitment (here the penalty is set as 25% of CT reward). As can be seen, both of the related hypotheses were rejected, meaning that the means obtained by the different levels of commitment are different by a statistically significant amount. To see how the levels compare, it is necessary to do a further analysis. Agents with partial commitment obtained the best result with AU of 4,696.83, while agents with Total and Loose commitments obtained values of 4,293.06 and 4,077.24 respectively. To clarify the results, Figure 8.2 shows the variance in the reward distribution by agent for the different levels of commitment. Specifically, the total reward obtained by the agents increases as they have more opportunities for decommitment. This is because agents can drop commitments to take up more profitable ones as they arise (even accounting for the fact that they have to pay a penalty). However, looser commitments do not necessarily improve agent performance; if agents can drop commitments easily, then a greater percentage of started CTs fail to finish because agents are continually attracted onto newer more profitable activities. Thus having flexibility to drop commitments is good, but the best performance is achieved by also having some degree of loyalty to existing contracts. The differences between the various levels of commitment are

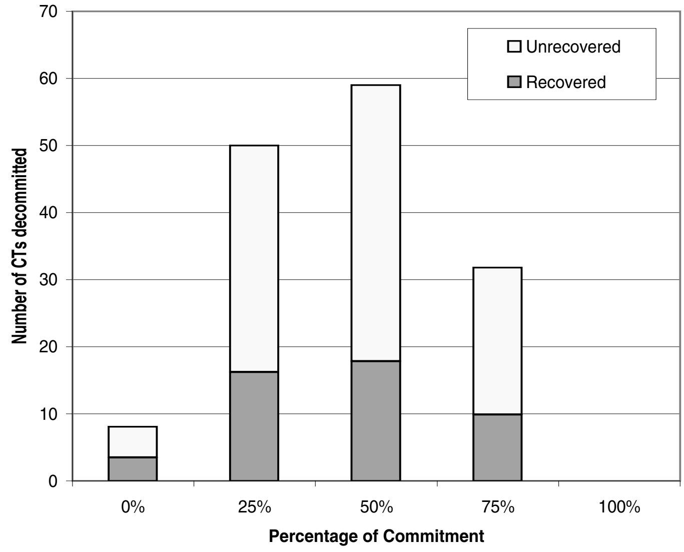


Figure 8.3: Contracts dropped by partial commitment grade.

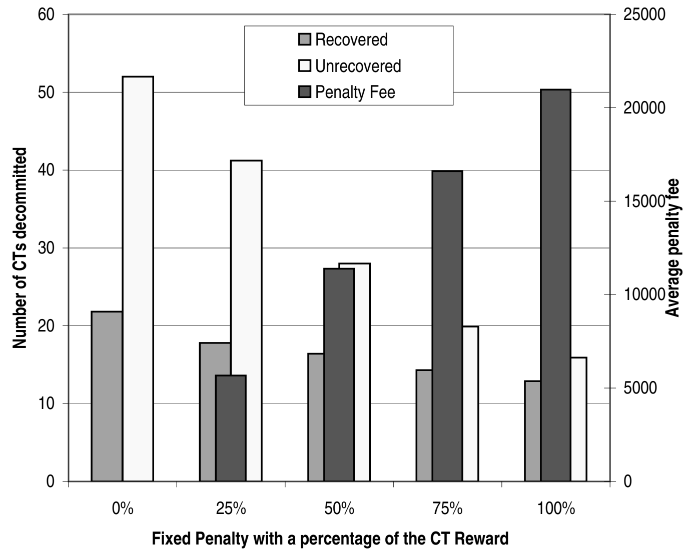


Figure 8.4: Fixed penalties.

mainly due to the CTs accomplished (H2). Here, as expected, the more CTs agents accomplish the more reward they obtain. H2 was rejected since agents with Partial commitment get more reward because they achieve more and more profitable CTs (TCT=92.2). Thus these agents perform better on average. The number of CTs obtained by agents that were partially committed was statistically better than the others, $\mathsf{TCT}_{\mathsf{Total}}$ got 78.4 and $\mathsf{TCT}_{\mathsf{Loose}}$ accomplished only 73.9. Thus the difference in AUs is based on the difference between TCTs; Partial agents obtain more reward than Total agents and, in the same way, Total agents gain more than the Loose ones.

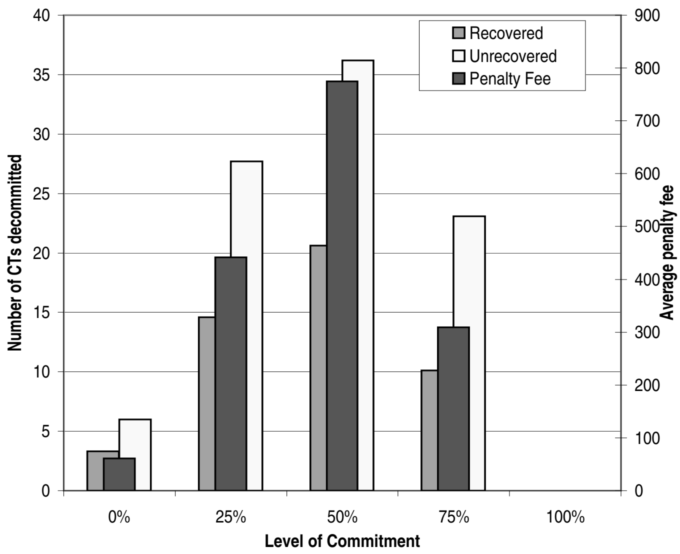


Figure 8.5: Partially sanctioned penalties.

As a consequence of introducing the degree of commitment into the agent's decision making procedures, agents with low levels of commitment have few opportunities to find CTs with high expected surplus rewards (equations (8.1) and (8.2)) and their bids are too high to ever be contracted (equation (8.3)). Correspondingly, agents with a high degree of loyalty to CTs have more chances to attempt coordination and their bids have a higher probability of being accepted. Additionally, the degree of commitment affects the frequency with which agents drop contracts, a higher degree means fewer decommitments are performed. Figure 8.3 clearly illustrates this behaviour. The number of CTs dropped by AiCoops varies with the degree of commitment; AiCoops with a commitment level of 25%, for example, decommit more often than those with 75%. In contrast, AiCoops with a commitment level of 0%, decommit much less frequently (even though they have more opportunities to do so) because their bids to participate in new CTs are too high (as previously noted).

The second group of experiments deals with the penalty aspect of decommitment decisions. Figures 8.4, 8.5 and 8.6 show the effects of the number of contracts decommitted using Fixed, Partially Sanctioned and Sunk Cost penalties, respectively. With Fixed penalties, Figure 8.4 indicates that the number of contracts dropped by AiCoops (here AiCoops have a commitment level of 50%) gradually decreases as the penalty fee increases (because agents cannot afford a high penalty fee). With Partially Sanctioned (Figure 8.5) and Sunk Cost (Figure 8.6) penalties, the same broad trends can be observed, fewer decommitments occur with low and high

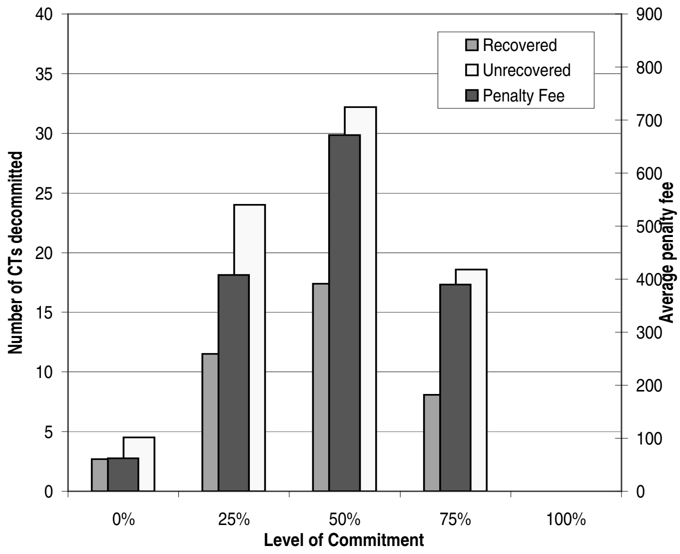


Figure 8.6: Sunk cost penalties.

degrees of commitments. This is because of the high penalties for decommitment and as a consequence of the fewer opportunities to decommit that occur when an agent has a low degree of commitment. However, both figures show similar values for the average penalty fee (second Y axis) and for the number of decommitments performed. In short, AiCoops with lower penalties perform more decommitments independently of the kind of sanction that is in place.

Turning now to the more interesting experiments, H3 tests the degree of improvement (AU) in the agents' coordination when various types of sanction are in place.

H3: The AU obtained by AiC agents (independently of their level of commitment) that use Sunk Cost is the same as that one obtained by agents that use Fixed penalty (of 0% and 50%) and Partially Sanctioned penalties.

Hypothesis to evaluate (AiC agents)	**p**	**Outcome**
H3: $AU_{SunkCost}=AU_{Fixed0\%}=AU_{Fixed50\%}=AU_{P.Sanctioned}$	0.047	Rejected

Table 8.2: AiC's AU given various types of sanctions: result of ANOVA.

By observing the ANOVA result (Table 8.2) and its corresponding post-analysis (Table 8.3), it can be seen that the kind of sanction does have an effect on the AU obtained by AiC agents (H3 is rejected). Moreover, Table 8.3 indicates that the most successful types of sanctions are the Fixed 50% and Sunk Cost. Nevertheless, these experiments do not say with which level of commitment the agents obtain

Kind of Sanction	AU	
	1	2
Fixed 0%	2236.41	
Partially Sanctioned	2296.68	
Fixed 50%		2332.20
Sunk Cost		2391.46
p	0.3472	0.3568

Table 8.3: Type of penalty: post-analysis.

the best results. In order to do this, it is necessary to be more specific and evaluate the agent's improvement given the kind of sanction in combination with varying levels of commitment:

H4: The AU obtained by AiC agents with a Partial level of commitment is identical whatever kind of penalty is employed.

H5: AiCs agents that have Loose commitment with a Fixed penalty obtain the same AU as those agents that use Sunk Cost and Partially Sanctioned penalties.

Recall that because agents with Total commitment are not affected by the kind of sanction employed, there is no hypothesis associated with this type of agents. In fact, testing the hypothesis with Total AiC agents ANOVA obtains the following result:

Hypothesis to evaluate (Total AiC agents)	**p**	**Outcome**
H6: $AU_{Fixed0\%}=AU_{Fixed50\%}=AU_{P.Sanctioned}=AU_{SunkCost}$	1.000	Accepted

Table 8.4: AiC's AU with Total commitment: result of ANOVA.

Table 8.5 shows the result of testing H4 and H5 in terms of the AiC total reward with respect to commitment level and penalty type. As can be seen, the type of penalty employed by Loose agents does not affect the AiC reward obtained (meaning H5 is accepted). However, H4 is rejected because there is a significant impact on the AiC reward gained by Partial agents (as shown in Figure 8.7). The best reward is obtained by combining a Sunk Cost penalty with Partial commitment (50%). With Sunk Cost and Partially Sanctioned penalties, agents decommit less frequently than they do with a 0% Fixed penalty, but more often than they do with a 50% Fixed penalty. However the reward obtained by AiCs using a fixed penalty of 50% is better than that of the Partially Sanctioned penalties. From this, it is noted that using a sanction that changes dynamically with the state of the coordination, specifically, sunk cost, seems a good model because it sets a more appropriate payment for decommitment when compared with the partially sanctioned penalties.

Hypothesis to evaluate	**p**	**Outcome**
Partial AiC agents H4: $AU_{Fixed0\%}=AU_{Fixed50\%}=AU_{P.Sanctioned}=AU_{SunkCost}$	0.000	Rejected
Loose AiC agents H5: $AU_{Fixed0\%}=AU_{Fixed50\%}=AU_{P.Sanctioned}=AU_{SunkCost}$	0.167	Accepted

Table 8.5: Comparing penalties: result of ANOVA.

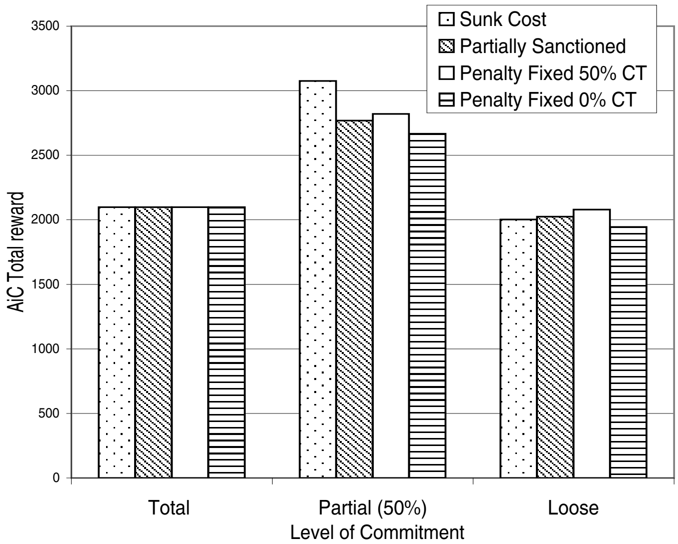


Figure 8.7: Comparing fixed, partially sanctioned and sunk cost penalties.

To summarise, as was the case with the commitment related experiments, sunk cost and partial commitment level agents represent the most effective combination between loyalty to the existing situation and flexibility to discover better situations and to submit better proposals.

8.4 Discussion

The extension of the basic framework with decommitments and penalties was undertaken with aim of enabling agents to coordinate in a more flexible manner. In fact, it was believed that this extension would allow autonomous agents to make more rational choices about coordinating their actions and when to relinquish their current commitments. To this end, the experiments in this chapter have shown that adding such features does indeed improve the performance of the agents with respect to coordination. In particular, it was shown that allowing decommitments increases the agents' rewards and that high penalties for decommitment discourage decommitment. Thus, a certain degree of loyalty to existing contracts leads to better overall performance than continually jumping to new opportunities as they arise. In contrast, to the previous works in this area (as discussed in Section 2.2.3), these results advance the state of the art in this field in several ways. Firstly, a general decision making framework is introduced that can both dynamically select CMs and can reason about how and when to relinquish existing commitments in order to participate in more profitable activities. Secondly, through empirical evaluation, it was demonstrated that the efficiency of coordination activities could be improved by this framework.

Chapter 9

Learning Extensions

This chapter investigates a number of extensions to the basic framework of Chapter 4 that are centred around the issue of introducing learning capabilities into an agent so that it can improve its decision making about coordination. The motivating hypothesis is that to achieve the necessary degree of flexibility in coordination requires an agent to make decisions about when to coordinate and which coordination mechanism to use (as argued in Section 2.3). So far, however, the empirical evaluation has highlighted the importance (as well as the difficulty) of making good approximations about the behaviour of other agents (see Section 7.6). Moreover, such approximations are especially challenging as the environment becomes more dynamic and unpredictable.

Against this background, a natural extension of the framework is to enable the agents to acquire knowledge through run-time adaptation (as argued in Section 2.3). Thus, the agents need to be capable of learning to make the right decisions about their coordination problem. In particular, those aspects of the model where such adaptation could be beneficial are that agents *can learn the right situations in which to attempt to coordinate* (Section 9.2) *and the right method to use in those situations* (Section 9.3).

The agent's decision procedures presented in Section 4 and, among them, the procedure outlined in Section 4.2, allows agents to take decisions about when and which coordination mechanism to select in order to achieve a CT. Since this procedure is the major one with respect to reasoning about coordination mechanisms, it is the one which is analysed in terms of evaluating the role of learning [1].

To this end, this chapter is structured in the following way. First, the formal model of the Q-learning algorithm is introduced (Section 9.1). After this, the agents' behaviours are extended (and subsequently evaluated) to use the learning techniques to learn which coordination mechanism is appropriate in which circumstances (Section 9.2) and the constituent components of the decision making procedures (Section 9.3).

9.1 Q-learning

In this study, each reinforcement learning agent uses a Q-learning algorithm. In general terms, an agent's objective is to learn a decision policy that is determined by the state/action value function. The classical model of Q-learning consists of:

- a finite set S of states s of the world ($\mathsf{s} \in \mathsf{S}$);
- a finite set A of actions a that can be performed ($\mathsf{a} \in \mathsf{A}$);
- a reward function $\mathsf{R} : \mathsf{S} \times \mathsf{A} \rightarrow \mathsf{r}$.

An agent's goal consists of learning a policy $\pi : \mathsf{S} \rightarrow \mathsf{A}$ that maximises the expected sum of discounted rewards [2] V:

$$V[\mathsf{r}_t + \gamma \mathsf{r}_{t+1} + \gamma^2 \mathsf{r}_{t+2} + ...] = V \sum_{i=0}^{\infty} \gamma^i \mathsf{r}_{t+i}$$

where $0 \leq \gamma < 1$ is the discount factor [3]. Thus, the agent's task is to learn the optimal policy π (i.e. $arg\max_\pi V^\pi(\mathsf{s}), \forall(\mathsf{s})$).

[1] There are clearly other places where learning could play a role, for example, an agent might learn the decision about how much to bid to become an AiCoop (equation (4.6)) and which bids to accept (equation (4.8)). However, this is left as future work (see Section 10.3 for more details).

[2] Discounting rewards is a function that considers rewards received in future steps to be less valuable than those received in the current time step. The standard model of discounting rewards uses a utility function and a discount factor. In the context of Q-learning, this means that an agent selects actions such that the sum of the discounted rewards it receives over the future is maximised.

[3] Formally speaking, the discount factor determines the value of future rewards in the following way: a reward r received t time steps in the future is worth only γ^t times what it would be worth if it were received immediately. As γ approximates 1, the function takes future rewards into account more strongly.

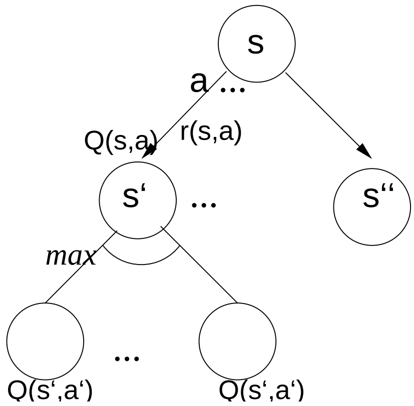


Figure 9.1: Exemplar Q-learning tree.

In more detail, assume that an agent always perform the cycle (illustrated by Figure 9.1) of being in particular state s, then selecting and performing an action a that causes the agent to enter a new state s' and receive an immediate payoff (reward $\mathsf{r}(\mathsf{s},\mathsf{a})$). The Q-learning algorithm is based on the estimated values of the agent's state (s)-action (a) pairs, called $\mathsf{Q}(\mathsf{s},\mathsf{a})$ values. Based on this experience, the agent updates its $\mathsf{Q}(\mathsf{s},\mathsf{a})$ values using the formula:

$$\mathsf{Q}(\mathsf{s},\mathsf{a}) \leftarrow (1-\alpha) \times \mathsf{Q}(\mathsf{s},\mathsf{a}) + \alpha[\mathsf{r} + \gamma \times max_{a'}\mathsf{Q}(\mathsf{s}',\mathsf{a}')]$$

where α is the learning rate (decreasing with time by calculating it with the number of times a $\mathsf{Q}(\mathsf{s},\mathsf{a})$ value is visited $visits(s,a)$: $\alpha = \frac{1}{1+visits(s,a)}$). And $max_{a'}\mathsf{Q}(\mathsf{s}',\mathsf{a}')$ is the value of the action that maximises the Q function at state s.

When agents select their next action to execute, they have to balance their decision between selecting an action that, when exploited in the past, brought about a positive reward, and an action that has not yet been explored and that consequently has an unknown reward ("exploitation versus exploration" Section 2.3). For experimental evaluation purposes in this work, $\gamma = 0.90$ (which means that the agent is reasonably farsighted) and a greedy function is employed as the exploration function [4] until the action has been executed a pre-determined number of times. Formally, the exploration function $f(u,n)$ (Russell & Norvig, 1995e) equates to:

[4]A greedy function means the action with the highest value is the one selected (being greedy) but, once in a while, an action is randomly selected (or using any other mechanism that is independent of the action value estimation).

$$f(u,n) = \begin{cases} R^+ & \text{if } n < N_e \\ u & \text{otherwise} \end{cases} \tag{9.1}$$

which returns the weight of a $\mathsf{Q(s,a)}$ value based on the number of times (n) this policy has been visited ($visits(s,a)$). The action with the best $f(u,n)$ value is selected to be performed. R^+ is the best possible reward that an agent can obtain in a given state, N_e corresponds to the number of times that agents should try a particular action-state pair and u represents the utility of a $\mathsf{Q(s,a)}$ value.

To clarify the use of Q-learning in this context, Figure 9.2 instantiates Figure 9.1 for reasoning about which CM to select in a given situation. Here, the agent-state (s) corresponds to the abstraction of the particular situation that agents experience when a CT is found (for example, the agent role (AiS, AiC, AiCoop), position in the grid (x,y) and so on); the agent-action represents the set of options an agent has at its disposal (i.e. the set of coordination mechanisms it can select, including the null CM: CM_1, CM_2, ..., CM_n and null CM), and the reinforcement is modelled as the reward obtained by selecting the particular CM ($\mathsf{r(s,a)}$).

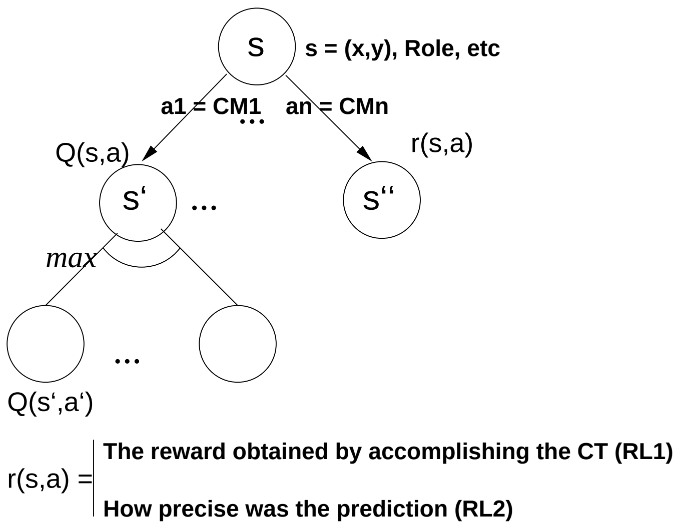


Figure 9.2: Role of Q-learning: Learning a CM.

9.2 Learning to select a CM

The objective of this section is to evaluate the effect of learning on the agents' decision making about CMs. To do this, the performance of agents that use a

Q-learning algorithm (RL) is compared with those that perform no learning (NL). Here the key difference is how the agents select the CM with which they will attempt coordination (step [2] in the protocol specified in Figure 3.1). For the remaining steps of the protocol, both RL and NL agents employ the decision making procedures outlined in Section 4 to make agreements when surplus (equation (4.8)) is positive given the set of bids (equation (4.6)) it received.

In more detail, and by looking at Figure 9.2, when an agent finds a CT, it calculates the expected average surplus (equation (4.5)) of each CM at its disposal. With NL the agent simply chooses the one with the best ave_surplus. With RL it exploits-and-explores (equation (9.1)) the set of CMs. When using RL, the reinforcement is used to measure the benefit of having selected a particular CM which corresponds to the surplus gained by achieving a CT using the CM chosen [5] after paying the AiCoops. Thus, the idea is that with Q-learning the agents will eventually learn the policy (after exploring sufficient situations) that allows them to know which CM to choose given a specific situation/state.

It is clear that the reinforcement is a central element in the process of learning because it is the mechanism to praise or blame if a good or bad action is performed. Thus, it was decided to consider alternative values and moments to provide the reinforcement. Generally speaking, the role of the reinforcement is to assess the evaluation performed on the choice of CM. To this end, the ave_surplus corresponds to the predicted value that an AiC expects to obtain by selecting a particular CM. When the evaluation phase is finished, the AiC receives firm information from the other agents and it is in position to evaluate this prediction. Thus, the AiC compares its predicted ave_surplus with the firm value negotiated (i.e. the surplus, equation (4.8)). If the prediction is close enough ($\pm$ 25%) to the real information, a strong reinforcement is made; but if it is not that close, a negative reinforcement is made [6].

In addition to this basic reinforcement method, it was deemed interesting to see whether the coordination decision making could be improved if the AiC builds a model of the other agents in the environment. That is, can RL agents improve their prediction of ave_surplus if they have a model of the other agents? To evaluate this, a simple representation of the other agents was constructed; namely, the

[5]Actually, accomplishing CTs is the only case considered. Even though agents achieve ST tasks, this information is not considered as reinforcement since it is not relevant to the agent's decisions about CMs.

[6]The absolute value of ave_surplus is used to provide the reinforcement in either the positive or the negative direction.

key variables that are crucial to coordination decisions were simply recorded. In particular, it was decided to explore r_AiCoop in equation (4.3) and so, AiC could calculate the value of r_AiCoop by averaging the bids it receives from the other agents (in contrast to using the AiC's own average reward as per Chapter 4).

In summary, the agents' performance will be analysed using the following algorithms:

- RL1 agents learn to select a particular CM according to the profit gained by accomplishing CTs with a particular CM.
- RL2 agents learn to select a particular CM according to the accuracy with which they predict the ave_surplus (which is based on the tailoring of r_AiCoop to their prevailing circumstances).
- NL agents do not engage in learning activities.

To finish the discussion on the role of learning in the model, it is necessary to specify the features of the environment in which the algorithms will be tested. Two scenarios have been designed: scenario1 in which all AiSs in the environment become AiCoop by submitting a bid that is calculated by equation (4.6) and scenario2 in which AiSs calculate their bids in the same way but they vary the result by a random factor. The reason for this change is that in the general case AiCs face a great deal of uncertainty in predicting this value. Thus the random element mirrors environments in which predictions are less accurate. Together, these two scenarios constitute a reasonably static environment in which good predictions can be made and a more dynamic one in which predictions are inherently less accurate.

9.2.1 Experimental evaluation

The main hypotheses to evaluate is whether agents coordinate more effectively in the scenario using the reinforcement based algorithms. The experimental evaluation is conducted as per previous experiments, namely, testing hypothesis using the methodology introduced in Chapter 6. The following simulation variables were fixed for all learning experiments: duration (50,000 time units) [7] and the rest of variables are the ones introduced in Table 6.2. The experiments reported here take into account the agent's protocol and decision making procedures discussed in Figure 3.1. Note these do not include the introduction of commitment and penalties, either in the protocol or the formulation. Again this is to ensure that differences in the results are related solely to the effect of the different approaches to learning.

Following the same procedure as before, to accept the main hypothesis, the hypotheses presented below must be rejected and the values of the experimental variables of a particular learning algorithm should produce significantly better results than those obtained with NL. Therefore, the following hypotheses must be tested in scenario1 and scenario2:

H1: The agent utility obtained by performing a reinforcement based algorithm is the same as that obtained by agents which use the NL algorithm.

H2: The number of CTs achieved by agents by means of either of the reinforcement learning algorithms is identical to that of agents using NL.

H3: The agent utility obtained by RL1 is the same as that of RL2 (evaluated in the case where H1 rejected).

H4: The number of CTs accomplished by RL1 is identical to that of RL2 (evaluated in the case where H2 is rejected).

To this end, Table 9.1 presents a summary of the results obtained by performing ANOVA on the data collected by each of the algorithms in scenario1. Considering the agent utility hypothesis first. H1 is rejected, meaning that the performance of the algorithms does have a significant effect on the AU obtained. To understand this result, a post-analysis of the AU values obtained by each algorithm was necessary. Here, the interesting conclusion is that the performance of NL is better by a statistically significant amount ($\mathsf{AU}_{\mathsf{NL}} = 10,138.30$) than RL1 and RL2

[7] On this occasion the duration was increased to allow the learning algorithms to converge. This was necessary because this type of learning algorithm needs sufficient time to approximate to optimal values. The convergence time for the RLs is a combination of the learning rate, the exploration and exploitation function, the state representations and so on. However, it was not the objective to hand tune all these parameters to reduce the convergence time in particular cases. Rather, the values of all parameters were fixed and kept constant in both Q-learning implementations.

($AU_{RL1} = 9,413.58$, $AU_{RL2} = 9,399.00$). Furthermore, comparing the performance of RL1 and RL2 in H3 (Accepted), it is concluded that the value and the moment of praising or blaming agents does not have any effect on the AU obtained.

Hypothesis to evaluate	p	Outcome
H1: $AU_{RL1}=AU_{RL2}=AU_{NL}$	0.000	Rejected
H2: $TCT_{RL1}=TCT_{RL2}=TCT_{NL}$	0.876	Accepted
H3: $AU_{RL1}=AU_{RL2}$	0.590	Accepted
H4: Not evaluated		

Table 9.1: Contrasting RL versus NL agent's abilities in scenario1: result of ANOVA.

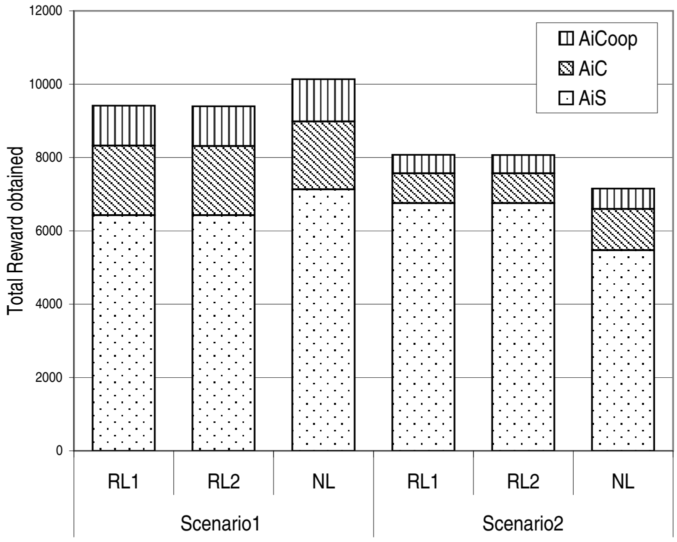


Figure 9.3: Contrasting RL versus NL agent's abilities. Reward obtained by agent role.

Contrary to what was expected, the two versions of Q-learning (RL1, RL2) do not have any effect on the agent's performance. Analysing in detail, it was found that this is because there is no change of information between the time agents make agreements and the time when they achieve the task. That is, no events occur that allow agents to change the reinforcement. Examples of events that could make a difference are if there are any decommitments or delays from AiCoops. The behaviour of the RLs is exploring and exploiting the actions and receiving a reinforcement that praises or blames the actions performed. Thus the two versions of RL are simply reinforcing the same actions but with different values.

Turning now to the more dynamic environment of scenario2. The same set of hypotheses were tested and the results are summarised in Table 9.2. First, the hypotheses related with AU (H1, H3) are analysed. In common with the results obtained in Table 9.1, the conclusion is that applying RL and NL produces distinctive results (H1 is rejected). But, conversely to Table 9.1, in this point RL1 and RL2 get significantly better results ($AU_{RL1} = 8,063.30$ and $AU_{RL2} = 8,067.98$) than NL ($AU_{NL} = 7,151.12$). Additionally, observing the ANOVA result of both RL algorithms, the same conclusion as that in scenario1 is made; namely, giving the reinforcement before and after the evaluation phase makes no difference to the final reward obtained by agents (H3 is accepted).

Hypothesis to evaluate	**p**	**Outcome**
H1: AU_{RL1}=AU_{RL2}=AU_{NL}	0.000	Rejected
H2: TCT_{RL1}=TCT_{RL2}=TCT_{NL}	0.000	Rejected
H3: AU_{RL1}=AU_{RL2}	0.835	Accepted
H4: TCT_{RL1}=TCT_{RL2}	0.940	Accepted

Table 9.2: Contrasting RL versus NL agent's abilities in scenario2: result of ANOVA.

With reference to the values of TCT (H2 and H4), the hypothesis of equal means of H2 is rejected and H4 is accepted. Thus, there is a significant impact on the TCT achieved when performing RLs or NL, where the results are 65 to RLs and 84 to NL. The relevant aspect to discuss now, though, is why NL obtains a lower AU despite achieving more CTs? Being consistent with the previous explanation, and observing Figure 9.3 (right section), it can be seen that the reward gained by achieving CTs for NL-AiCs is bigger than that gained by RL-AiCs because they achieve more CTs. However, the time invested on them was not sufficient to recover the reward that was being gained by RL-AiSs (RL-AiSs obtained in total approximately 84% of the total reward by accomplishing STs and NL-AiSs achieved 76%). The reason for this result is that agents invest a significant amount of time to set up the CM and, in the end, the AiCoops often request higher bids than those in scenario1 (meaning the AiCs' profit is reduced). Thus, RLs perform better because they are more certain about when to invest time in a CT and, more importantly, when not to do it (because it is not worth it). They then use this time to take advantage of pursuing STs.

Before making a general conclusion about the irrelevance of the different options of reinforcement and modelling other agents, Figure 9.4 shows the performance of the algorithms from a different perspective. Here, the performance of agents by evaluating the average surplus reward expected (equation 4.5) is analysed. This analysis is undertaken because it is important to know if it really worthwhile having adaptive agents instead of just hand tuning the agents' decision making procedures about which CM to select. To be precise, NL agents select the CM based on ave_surplus, but RLs agents do not (RL2 uses the result of this evaluation just as a reinforcement). Thus, the idea is to count the number of cases in which the action selected by either RL algorithm coincides with the outcome indicated by equation 4.5. In other words, given a specific situation, RL agents exploit the CMs and evaluate this formulation, then they analyse the ave_surplus of that CM. If it is positive, then a case of TT is encountered (because the CM was exploited and it coincided with the decision based on ave_surplus). On the other hand, if it is negative, the CM was exploited but this did not correspond with the evaluation of equation 4.5 (case TF). Additionally, there are the cases in which the CM was not exploited and this corresponds (FF) or not (FT) with the decision based on ave_surplus. It is clear that with NL all the cases have to be consistent because they attempt coordination based on this evaluation. As for the RLs, it will depend on how this equation is calculated. Both NL and RL1 agents employ equation 4.5 using their own r values for r_AiCoop, whereas RL2 uses the value that it has learnt based on previous encounters.

To this end, Figure 9.4 shows only the TT and FF cases since the others (TF and FT) are not relevant in this discussion. The first thing to notice is the clear difference between RLs, which is due to the different ways in which the ave_surplus is calculated. The second observation is that RL2 has more FF cases, meaning that most of the time the action it performs is to not attempt coordination and in these cases this decision is consistent with that based on the ave_surplus. The reason for this result is that RL2 is modelling others using r_AiCoop which helps it to make a good prediction of the other agents. Given this, the obvious question to ask is could NL perfom better by modelling others in the same way as RL2? Here, the answer is no. The system utility obtained by NL agents in scenario1 degrades considerably when agents use r_AiCoop ($\mathsf{AU}_{\mathsf{NL}} = 7,575.28$) and in scenario2 it raises slightly ($\mathsf{AU}_{\mathsf{NL}} = 7,578.42$). Despite the improvement in scenario2, it is not sufficient to have H1 accepted. If the NL agents' predictions of ave_surplus are too low (being optimistic about the possible future cooperative agents), they will always initiate coordination even in situations where it is not the best decision to make. However, if their predictions are too high (being pessimistic) they will never attempt coordination. Thus, the conclusion is that having learning agents that explore and exploit actions is the best thing to do in dynamic environments because agents cannot be certain about others' actions.

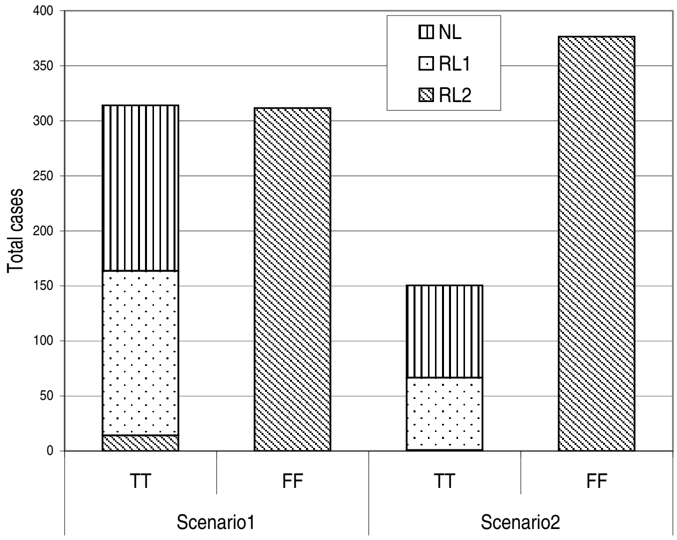


Figure 9.4: RL and ave_surplus action selection

9.3 Learning the decisions' constituent factors

This extension investigates whether the agents can learn about the key factors that influence their decisions about when to attempt coordination. To do this, agents need to be in a situation in which they cannot correctly predict the others' behaviour. Thus, scenario2 of Section 9.2 is considered. In this environment, the hypothesis is evaluated empirically using reinforcement based algorithms as per Section 9.2.

To test the learning hypothesis, the performance of agents that use a Q-learning algorithm (RL) is compared with those that do not (NL). Here the key difference is how agents estimate the ave_bid (this variable represents the prediction of other agents' bids) when selecting the CM with which they will attempt coordination (step [2] in the protocol specified in Figure 3.1). In previous experiments (Section 9.2), agents explored and exploited the CMs (which represent the agent's actions) given a specific situation (which corresponds to the agent's states) and they learnt to choose the most profitable CM. Here, the concern is with learning about the constituent factors that are involved in this decision and, to be more specific, about the decision of when to coordinate. For the remaining steps of the protocol, both RL and NL agents employ the decision making procedures outlined in Section 4 to make agreements when there is a positive surplus (equation (4.8)) given the set of bids (equation (4.6)) received. To this end, Figure 9.5 provides a graphical representation of this situation in terms of the Q-learning algorithm's components.

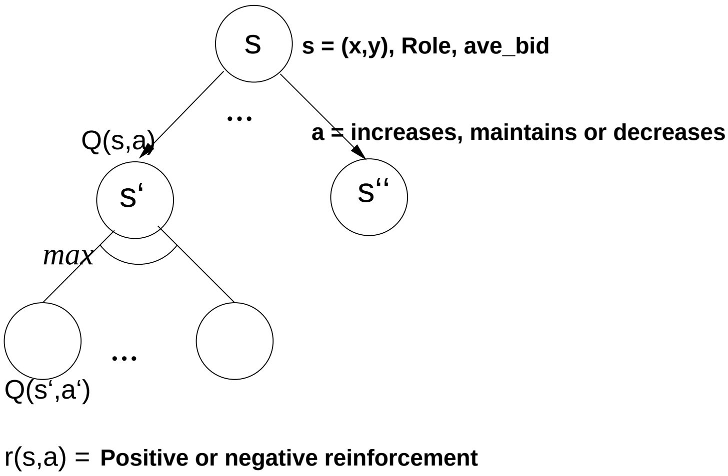


Figure 9.5: Role of Q-learning: Learning decision making factors.

In general, when an agent finds a CT, it calculates the expected surplus (equation (4.5)) of each CM at its disposal and chooses the one with the best ave_surplus. With NL, the agent uses equation (4.3) to calculate ave_bid and then it evaluates ave_surplus of the CMs. In contrast, with RL, the agent computes ave_bid using Q-learning and then it estimates the ave_surplus as before. Here, the objective of the Q-learning is to predict the ave_bid value by increasing, decreasing or maintaining a pre-calculated value and then assessing the action performed depending on the success or failure of the CM selection at the evaluation phase (step [3] in the protocol specified in Figure 3.1). The reason for performing the evaluation after the evaluation phase is because it is at this moment that the agents receive firm information from the potential collaborators. Consequently, they are in a position to evaluate their prediction. If the AiC finishes the evaluation phase with the m collaborators required by the CT, it means that the expected surplus of the CM (based on the ave_bid) was correct and, therefore, the action performed is praised. But, on the other hand, if the AiC finishes the evaluation phase with insufficient AiCoops (for whatever reason) the predicted value of ave_bid was inappropriate and a negative reinforcement is applied to blame the action performed. The process of rewarding and punishing actions continues a sufficient number of times to allow the agents to learn the ave_bid with which the CMs maximise the discounted rewards.

To be more precise, an RL agent initiates ave_bid with a predefined value [8] and

[8]The agent's initial estimation is that the other m agents will require a reward of S, thus ave_bid=$m * S$. This is a reasonable start point since it represents the minimum reward that an agent is likely to bid.

when it faces a coordination problem it exploits-and-explores (equation 9.1) one of the possible actions open to it. It then increases, decreases or maintains the ave_bid according to the reinforcement received about the success of the action. The agent then uses this updated value of ave_bid to evaluate the ave_surplus of all CMs. Thus, in this scenario, the agent-state corresponds to the abstraction of the particular situation that agents experience when a CT is found (for example, the agent role, position in the grid and the value of ave_bid used [9]); the agent-action consists of the operation to perform on ave_bid; and the reinforcement is a positive or negative value based on the result of the evaluation phase. For example, one agent might learn a policy that says, if it is an AiS with an ave_bid higher than 5.0, it should decrease this value by 0.2 to find a CM that maximises the reward.

In summary, the agents' performance will be analysed in scenario2 using the following algorithms:

RL: agents learn the ave_bid value to evaluate the set of CMs.

NL: agents do not engage in learning activities.

9.3.1 Experimental evaluation

In this case, the hypothesis to evaluate is whether learning about the bidding behaviour of the other agents improves the effectiveness of an agent's reasoning about coordination. To measure the benefits of introducing this extension to the framework the same experiments as those conducted in Section 9.2.1 were designed and, similarly, the same experimental and hypotheses variables were used. To substantiate the claim, the hypotheses presented below must be rejected and agents using RL should produce significantly better results than those obtained with NL.

H5: The AU obtained by performing the RL algorithm is the same as that obtained by agents which use the NL algorithm.

H6: The TCT achieved by agents using the RL algorithm is identical to that of agents using NL.

[9]To simplify the state representation, ave_bid is in fact associated with a range of values. In this scenario the ranges are the following: ave_bid< 1, $1 <$ave_bid< 3, $3 <$ave_bid< 5 and ave_bid≥ 5.

Table 9.3 presents a summary of the results. Considering the agent utility hypothesis first. H5 is rejected meaning that the performance of the algorithms does indeed have a significant effect on the AU obtained. To understand this result, a post-analysis of the AU values obtained by each algorithm was necessary. Here, the conclusion is that the performance of RL is better by a statistically significant amount ($\mathsf{AU}_{\mathsf{RL}} = 8,822.36$) than NL ($\mathsf{AU}_{\mathsf{NL}} = 7,151.12$). This leads to the conclusion that the RL agents take more profitable decisions about coordination than the NL agents. However to understand the origin of the agents' reward it is necessary to analyse H6 in more detail.

Hypothesis to evaluate	**p**	**Outcome**
H5: $\mathsf{AU}_{\mathsf{RL}}$=$\mathsf{AU}_{\mathsf{NL}}$	0.000	Rejected
H6: $\mathsf{TCT}_{\mathsf{RL}}$=$\mathsf{TCT}_{\mathsf{NL}}$	0.000	Rejected

Table 9.3: Contrasting NL versus RL agents: result of ANOVA.

H6 evaluates the effectiveness of achieving CTs. Again, this hypothesis is rejected; meaning that the total number of CTs achieved does depend on the algorithm executed. There is a significant impact on the TCT achieved by performing RL and NL, where the results are 84.0 to NL and 71.76 to RL (Figure 9.6 shows on its Y axis the total CTs accomplished by agent type). Contrary to what was expected, agents that achieve higher numbers of CTs are not always those with a better AU. To explain this result Figure 9.6 shows the total reward obtained by agent role (X axis). Here notice that the reward gained by achieving ST tasks is the biggest part of the total reward and NL accomplishes fewer ST tasks than RL. Another point to note, is that in this scenario it is expensive (due to the set-up cost of the CMs) to invest in a CT when there is some uncertainty about achieving it. With NL, it seems that the AiCs cannot make good enough predictions of ave_bid. Therefore they attempt coordination (or the AiC might even fail after the evaluation phase) even though the profit obtained after achieving the CT was not as high as the reward that was being gained by RL-AiSs (RL-AiSs obtained in total approximately 84% of the total reward by accomplishing STs and NL-AiSs achieved 77%). By not predicting ave_bid accurately enough, the NL-AiC is faced with the situation where the AiCoops request higher bids than it expected meaning its profit is reduced. It is important to observe that the solution is not to avoid the CTs tasks and only persue STs. Rather, the answer is to find the right balance between the two because in this scenario CTs always provide better rewards than STs.

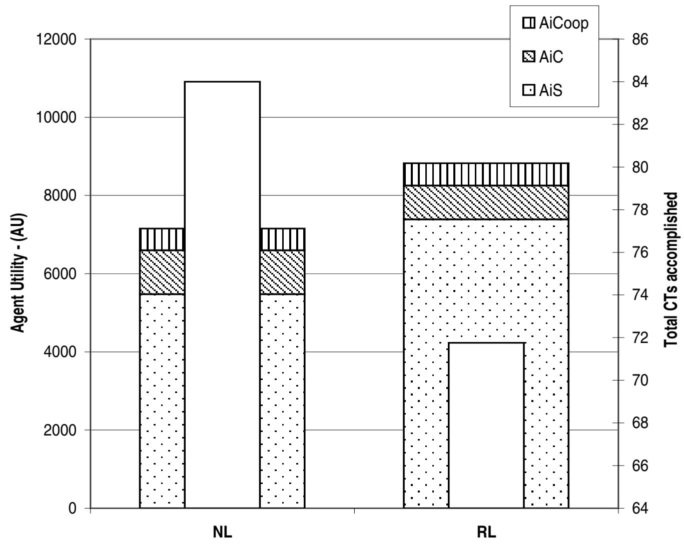


Figure 9.6: NL versus RL agents. Reward obtained by agent role.

In summary, in dynamic and unpredictable environments RL agents perform better than NL agents because they are more certain about when to invest time in a CT and, more importantly, when not to do it (because it is not worth it). RL agents then use this time to take advantage of persuing STs. Learning about ave_bid helps RL agents to have a more precise model of this value and, consequently, their predictions are closer to actuality. This, in turn, means the agents are more effective at maximising their profits.

9.4 Discussion

This chapter showed that reinforcement learning techniques, and in particular Q-learning, are useful when agents cannot correctly predict the behaviour of others (as occurs in open, unpredictable and dynamic settings). Specifically, the experimental evaluation showed that agents improve their decision making about when and how to coordinate. However, the experiments also highlighted the fact that learning is not a panacea. In particular, it was ineffective when agents operate in more static environments in which they can make reasonable predictions about their environment and other agents.

There are, however, a number of aspects that still need to be explored regarding the introduction of learning in this model. Firstly, learning could be introduced in areas other than learning to make decisions and learning about the constituent factors of the decisions themselves. For example, agents could learn the meta-data parameters of the CM (i.e. they could refine the CM's parameters given the efficiency of its actual execution). However, this aspect was not addressed in this chapter because it is directly related with the specifics of how the different CMs operate to achieve coordination, which is outside the scope of this thesis. Secondly, the comparison between algorithms in this chapter followed the experimental evaluation methodology outlined in Chapter 6 rather than the more classical way of reporting a learning algorithm's performance. In the ML literature, the performance of a learning algorithm typically focuses on its converge time, and the parameter settings associated with the various levels of performance (Claus & Boutilier, 1998; Tan, 1993; Nagayuki *et al.*, 2000). Here, however, the purpose was not to focus on the learning details, nor to reduce the convergence time but rather to fix all the parameters and then contrast the various agents' performances. This is because the main aim was to allow fair comparisons to be undertaken. However, this decision has certain consequences. For example, intuitively, it was believed that learning algorithms RL1 and RL2 in Section 9.2 should obtain a better performance than NL in scenario1. However, recall that the results were taken considering the exploration and exploitation phase, not once the learning converges to the optimal action. Thus, it is now believed that by not taking into account the time invested in the exploration and making the comparison with the results once the algorithms reach the optimal policy, RL agents might indeed obtain a better performance than the NL. However, this claim needs corroboration (see Section 10.3).

Chapter 10

Conclusions and Future Work

This chapter summarises the findings of this thesis towards sustaining the claim stated in Chapter 1 *that autonomous agents which have the flexibility to select at run-time the most appropriate mechanism to coordinate their activities exhibit better performance than those which do not possess such flexibility*. In order to support such a claim, a novel line of research was identified in which agents reason at a high level of abstraction about the coordination problem and take decisions at the time when the selection of coordination mechanisms is required. Although several researchers have attempted to introduce flexibility into coordination solutions, their attention has generally focused on different aspects of the problem and thus this thesis can be seen as a opening up a new direction of study.

More specifically, this thesis presents and evaluates a decision making framework that agents can use to make choices about coordination options according to their prevailing context. By using this framework, agents are able to deal with coordination decisions in a more effective and efficient manner. This is possible because the framework possesses several important characteristics. First, it clearly distinguishes the concepts of coordination and cooperation in the sense that agents deliberate about whether to coordinate over cooperative actions and with which coordination mechanism to do so. Second, it incorporates meta-data about coordinating mechanisms into the agent's decision procedures. This meta-data identifies the fundamental issues for characterising coordination techniques; *what* is needed to apply the coordination technique and *what* is obtained as a result of its application but it is not concerned with *how* the interactions between agents are managed. Such a representation enables agents to discriminate between coordination mechanisms and to recognise the particular situations in which each technique can be applied without being concerned with the details of the resulting interactions. Third, it incorporates information into its run-time reasoning about the possible collaborators an agent might encounter, the environmental features and the coordination mechanisms. Such information has been shown to be the key elements of the coordination problem. Finally, it integrates a model of commitments into the agent's decision making procedures. In this extended framework, agents have the possibility of reconsidering agreements based on the ongoing coordination activity if more productive cooperative situations appear. This model of commitment takes into account three levels of commitments (total, partial and loose) and three kinds of sanctions (fixed, partially sanctioned and sunk cost).

When taken together, these features of the framework allow agents to accomplish flexibility with respect to coordination decisions in scenarios that are open and dynamic. Particularly, in the more dynamic settings it was shown that learning was an important requirement in order to achieve the necessary degree of flexibility. In particular, learning agents were designed that could tailor the process of selecting the CM to their experiences and that could refine the components of the decision procedure according to previous interaction. However, it was also shown that learning agents are not effective in all situations. The results demonstrated that when an agent's prediction about the other agents in the environment is approximately correct, learning does not provide any real advantage. The introduction of learning in the way described in this research is a contribution to the state of the art in itself because this work has demonstrated that agents can learn to cooperate using a perspective that has not been addressed in the current learning literature.

In what follows, the remainder of this chapter discusses in greater detail the results of the investigation carried out focusing on the following aspects:

- the decision making framework introduced in Chapter 4 (Section 10.1)
- the extension to the framework to deal with flexible commitments and penalties introduced in Chapter 8 (Section 10.2)
- the inclusion of learning abilities to the agents as detailed in Chapter 9 (Section 10.3)

Each of these sections draws out the main conclusions and then details the work that still needs to be investigated in the future. Now, each section is dealt with in turn.

10.1 About the Decision Making framework

This thesis argued that autonomous agents need to be given the flexibility to dynamically select the mechanism they use for coordinating their actions during problem solving. Thus, a decision making framework to make decisions about when to coordinate (and when not to do it), which coordination mechanism to use and with which agents to cooperate was introduced in Chapter 4. This framework enables agents to make informed choices about their coordination actions because it abstractly characterises coordination mechanisms in terms of their cost and their expected benefits. This decision is clearly separated from the enactment of the mechanism which is how the task is actually achieved and how the actions of the various agents are actually coordinated. Moreover, it was shown that in the grid world scenario the agents are more successful by having the ability to select the coordination mechanism dynamically.

Although the specifics of the decision procedures are clearly related to the particular grid world scenario, the basic processes and structures developed are suitable for reasoning about coordination mechanisms in more general domains. For example, several of the agent's decisions relate to the proximity of potential collaborators. This conception of distance in the grid can easily and naturally be mapped into a range of analogous concepts that have more general application. The first of these is the notion of trust in social relationships (as represented, for example, by the degree of connectivity in social network theory (Burt, 1982)). Thus, cases in which agents are more certain of receiving help, because there is a high degree of trust between them, are similar to the cases in the scenario where potential collaborators are close to hand. In such situations, the results indicate that agents are more likely to attempt coordination using mechanisms that have relatively low times to set up. On the other hand, when collaboration is more difficult to establish, because there is a low degree of trust (equivalent to the agent being far away), the agents are more likely to opt for mechanisms that are more likely to succeed. The second relates to the dynamism in the environment. In more static environments, there is a greater chance of more accurately predicting the behaviour of the various agents present in the system. This corresponds to the case of the agent being near the centre of the grid as this situation involves much less uncertainty. In more dynamic environments, on the other hand, predicting the behaviour of others is more difficult and so corresponds to decision making around the edges of the grid.

This research work also demonstrated that a successful agent is one that takes the right decision about the benefits it will accrue for the time invested in the coordination mechanism versus what the agent would gain by achieving its individual tasks. The importance of this result is that it corroborates the fact that in some environments an agent that can dynamically select its CM according to its circumstances is able to take more profitable coordination decisions. However, this thesis does not claim that the agents always select the best CM. For example, when agents are at the edges of the grid, they assume it will take a long time for the AiCoops to arrive to the CT cell and consequently they select the CM that has the highest probability of success regardless of the high cost of setting up. This is true even though this is not always the case in practice. Thus, an agent that makes a good decision needs to balance the coordination mechanism set up cost and its likelihood of success with the information predicted. However, in the experimentation reported here good performance is measured simply by the total reward obtained at the end and not by how well this trade-off was performed. Thus, a potential conclusion from this could be that the constituent factors of the framework need to be refined to more precisely model the environment. However, the process of environment modelling has the drawback that unless the problem deals with static environments (which is not the case), this task can imply a great effort and cost (Durfee, 1999b). Thus, the research position taken in this work is that the effort must be directed to, on one hand, having a reasonable model of those aspects on which agents base their coordinating decisions and, on the other hand, exploring the "right" level of approximation in modelling in order to ensure the agent can coordinate effectively in practice (Durfee, 1999b). The framework as outlined here can be viewed as a mature point of departure because it provides a good approximation about how the key factors that need to be taken into consideration should be combined. However, more work is needed to refine the modelling problem and to systematically evaluate the alternatives.

Another area of future work involves systematically classifying coordination mechanisms according to the meta-data dimension. Although some work was performed toward this direction in Chapter 2 more comparative analysis is needed with respect to related classifications. This is necessary in order to use in the context of practical applications and may result in a more refined set of characteristics of the coordination mechanisms themselves. In the same order of ideas, it is believed that the factors taken into consideration in the CM abstraction are those that are necessary as a starting point for modelling CMs. However, it is also believed that this abstraction may need to incorporate more aspects into the CM themselves and, consequently, in the decision making procedures. Examples of such aspects might be quality of coordination, robustness, overhead limitations and so on.

Additional work is also necessary to incorporate greater heterogeneity both in the agent population and in the coordination mechanisms available. In particular, to model other basic coordination mechanisms, which will allow the assumptions related to the Contract-Net-like protocol that has been implemented to be relaxed. This strand of work will also enable a clear separation to be made between the decision-making procedure and the coordination protocols. Moreover, although the grid world scenario incorporates a reasonable degree of dynamism and uncertainty, it is important to take the model into real applications. Focusing on such applications could also raise new problems that have not been addressed in the present scenario. Although comparatively little work addresses the run-time selection of particular coordination protocols (as detailed in Chapter 2), the research undertaken in the area of dynamic selection of problem-solving techniques deals with the run-time selection of algorithms and, under this perspective, the research of this thesis is somewhat related. Generally speaking, the work on dynamically selecting problem-solving techniques has developed several solutions that operate under different assumptions; these include design-to-time algorithms (Garvey & Lesser, 1993) and anytime algorithms (Zilberstein, 1996). The former assume that algorithms use all the available time in a given situation to generate the best solution possible. The latter improve their output quality as more time is invested and can be interrupted at any time. In contrast, the decision making framework of this thesis has no such mechanism for controlling its reasoning with varying time and resources availabilities (it simply consumes a constant amount of time). However, in complex and dynamic environments it might be appropriate to give this reasoner a time and resource related flavour and so it would be interesting to determine how these techniques could be applied to this end.

10.2 About Flexible Commitments and Penalties

In order to incorporate greater flexibility in the agents' decision making, the basic framework was extended to focus on the issues of variable commitment levels between agents and of different penalty sanctions for reneging on contracts. The empirical results highlighted the fact that flexibility with respect to commitment levels can indeed improve the effectiveness of coordination. It was shown that a certain degree of loyalty to existing contracts leads to better overall performance than continually jumping to new opportunities as they arise. For penalty sanctions, it was shown that setting them based on the prevailing context also improves an agent's performance.

In general terms, the importance of introducing variable levels of commitment and penalties rests on the fact that it was possible to construct an integrated decision making framework that involves the following decisions:

- which CM to select to coordinate (or not to coordinate)
- how much to bid to participate in coordination
- which bids to select
- when to decommit
- how to set the penalty fee

For the future, it is necessary to incorporate the empirical findings into an agent's decision procedures so it can select the level of commitment and penalty sanction for itself according to its prevailing circumstances. Further work is also needed in order to investigate additional sanctions mechanisms. Finally, to account for more heterogeneous agent populations, it is also necessary to evaluate communities of agents with different levels of commitments in the same environment. To this end, the idea would be to conduct the same exploration as that of Appendix A but considering levels of commitments and variable penalties. In addition, it would be also important to compare the performance of agents in communities in which some agents use the basic framework and some employ the framework with the enhanced commitment properties.

10.3 About Learning Extensions

This thesis showed that agents benefit from being more flexible when taking decisions about coordinating problems. One way of achieving such flexibility it through learning and adaptation. Thus, this thesis analysed the use and the efficacy of agents learning about making decisions about when and how to coordinate. In particular, it was shown that learning improves the decision making when agents are uncertain about the other agents' actions. This improvement occurs because agents learn to recognise the situations where the most profitable actions must be selected. However, it was also shown that learning was ineffective when agents operate in more static environments in which they can make reasonable predictions about their environment and other agents.

Against this background, a broader investigation is needed with respect to the issues of when and how to exploit learning techniques in allowing agents to take decisions based on their experience. To this end, the results presented here can be viewed as a first step in that direction. For the future, the aim is to extend the use of learning to cover other aspects of the agent's decision framework. In particular, agents might learn to take the decision about how much to bid in a request for coordination, when to become an AiCoop (equation 4.6) and which bids to accept (equation 4.8). It is also intended to allow agents to construct more sophisticated models of one another and to have the ability to vary the details of this modelling according to the agent's coordination context.

Despite the effectiveness achieved by introducing learning techniques into the model, to incorporate additional aspects requires a further investigation because the literature in this area highlights the complexity of introducing learning aspects in MAS problems. Specifically, formal studies of Q-learning (Kaelbling *et al.*, 1996; Littmann, 1994; Mitchel, 1997b; Sutton & Barto, 1998) in single agent environments indicate the effective way this algorithm converges. However, convergence in MAS environments has not yet been demonstrated. This is because the premises of Q-learning in environments where all agents learn concurrently violate the basic assumptions of the simple case. Consequently, developing agents that learn how to improve their behaviour in a MAS is a very difficult task. The work presented here showed that the learning process is highly influenced by how agents in the environment model one another. Though this thesis explored a comparatively simple case, it is believed that in order to accomplish more effective learning objectives, agents should model the others as *1-level agents* by explicitly representing knowledge about others. This is because most of the agent's decisions take into consideration predictions about the other agents and to refine these predictions an agent needs to represent in a more precise way the behaviour of the others in the scenario.

In a broader context, a final aspect to discuss is that learning can also be employed to refine the values associated to the CM meta-data based on an agent's individual experiences in a given context. In order to do this, the main aspect to work on is in developing several real CMs whose corresponding performance can be evaluated. In the model presented here, the likelihood of the outcome being achieved was represented simply as a percentage. However this is clearly not the only way of doing it. Thus, if agents measure and update the consequences of the selected CM's execution, they could update the meta-data information. For example, if an agent at run-time selects planning as a CM (it did so because it has an associated x as cost to set-up and y as the probability of success), the idea is to execute the planning coordinating algorithm and then update the meta-data based on this experience. Thus, over time the meta-data could be refined to reflect the agent's individual experiences.

Appendix A

Coordination in Heterogeneous Settings

In order to make meaningful comparisons, most the experiments reported in Chapters 7, 8 and 9 take place in broadly the same coordination environment (here termed a homogeneous setting) [1]. Thus, agents assign the same values to the simulation variables in the same simulation run and, as a consequence, agents possess analogous reasoning in their decision making processes. For example, all agents have the same set of CMs at their disposal, the same willingness factor (ω), and so on. However, it is also important to understand the impact on the performance of the decision making framework in more heterogeneous settings. To this end, the simplest level of heterogeneity is when agents associate different values to the simulation variables at the same execution time. Another, perhaps stronger, form of heterogeneity is when agents possess alternative techniques to take similar decisions.

Against this background, to ensure that the results of the main empirical chapters are not specific to the chosen configuration, this appendix explores a variety of heterogeneous settings. In particular, here the focus is on communities of agents that show alternative decision making behaviour. In more detail, the specific aspects of heterogeneity considered are:

- Agents with different dispositions to cooperate (Section A.2),
- Agents with alternative values for the constituent factors of their decision procedures (Section A.3), and

[1]The only case in which this was not the case was in the evaluation of the decision making framework in Section 7.5.

- Agents with different reasoning capabilities with respect to taking coordinating decisions (Section A.4).

These types of heterogeneity were chosen because they best represent the main aspects discussed in this thesis. More precisely, one important element discussed in Section 7.4 was the different social attitudes agents possessed regarding their willingness to cooperate (ω). Thus Section A.2 explores whether the conclusions of Section 7.4 still hold when agents possess varying ω settings in the same simulation run. Turning to the second point, it has been strongly suggested throughout this thesis that the constituent factors of the agent's decision making process are a determinant factor of its behaviour. However, the main body of this research [2] did not explore to what extent an agent's behaviour is effected by varying one fundamental and common component of the decision procedures. Thus, Section A.3 explores this by varying the agent's reward rate (r). Finally, perhaps strongest form of heterogeneity is having agents that have totally different mechanisms to that outlined in Chapter 4 in order to take decisions about coordination problems. To this end, Section A.4 explores environments that are populated by agents that follow the learning abilities specified in Chapter 9 (in particular Section 9.2) and those that follow the basic decision making process of Chapter 4. In other words, the environment is composed of agents that learn about which CM to select and agents that do not have such abilities.

Naturally, these experiments do not deal with all aspects of heterogeneity that could arise in the context of this thesis. Nevertheless, it is believed that some minimal evaluation of the heterogeneous cases is important for at least the following reasons. First, to start to be able to generalise the applicability of this framework to more realistic scenarios in which heterogeneity is found. Second, *"heterogeneity is desirable because it increases the systemwide capabilities, allowing agents with complementary attributes to combine their objectives beyond what they they can achieve individually."* (Durfee, 2001, pp. 42). Third, and more specifically, it is important to fully analyse to what extent an agent's performance is the result of its interactions with others and whether the conclusions drawn in previous chapters can be extended to these new settings.

[2] The experimental evaluation of Section 9.3 explores a related attempt, namely, agents learn the ave_bid that is employed in the decision procedure about which CM to select.

A.1 The Experimental Setting

The experimental evaluation in this chapter follows the methodology of Chapter 6 and the experimental and simulation variables are as defined in Tables 6.1 and 6.2. Being consistent with the previous experiments, the main hypotheses seek to evaluate whether a particular aspect of heterogeneity produces any benefit to the agents (meaning the main focus is on AU and TCT).

To explore these issues, it should be noted that the environment might be inhabited by different agents (called agent types) depending on the specific class of heterogeneity being evaluated. And, consequently, a number of communities (groups of agents (*group*) [3]) might be formed and need to be explored separately. Thus, as a result of having a number of groups composed of different agent types the hypotheses are evaluated at three levels of detail [4]:

Level i. Here the analysis is concerned with determining if some groups of agents exhibit behaviour that dominates that of the others groups. For example, if groups whose agents have more social attitudes obtain a better performance than those whose agents are less social. Thus, the hypothesis are of the form:

$$\text{H1} : \mathsf{AU}_{group\ 1} = \ldots = \mathsf{AU}_{group\ n}$$

$$\text{H2} : \mathsf{TCT}_{group\ 1} = \ldots = \mathsf{TCT}_{group\ n}$$

where n indicates the total number of groups composed.

Level ii. Here the analysis is concerned with the performance of individual agents within particular groups. For example, how does an altruistic agent perform in a group composed of greedy and neutral agents?. These hypotheses are evaluated with ANOVA and have the following form:

$$\text{H1.1} : \mathsf{AU}_{A0} = \mathsf{AU}_{A1} = \mathsf{AU}_{A2} = \mathsf{AU}_{A3} = \mathsf{AU}_{A4}$$

$$\text{H1.2} : \mathsf{AU}_{A0} = \mathsf{AU}_{A1} = \mathsf{AU}_{A2} = \mathsf{AU}_{A3} = \mathsf{AU}_{A4}$$

...

$$\text{H1.n} : \mathsf{AU}_{A0} = \mathsf{AU}_{A1} = \mathsf{AU}_{A2} = \mathsf{AU}_{A0=3} = \mathsf{AU}_{A4}$$

[3]Even in the case of 5 agents in the environment, as indicated in Table 6.1, the number of groups will depend on the number of types of agent being considered.

[4]Recall from Chapter 6 that H0 is formulated with the equality of means of the variables to be analysed.

where the subindex in the hypothesis number is associated with the group being tested. For example: H1.2 tests the AU (because H1 tests AU) in group 2 and H1.n tests the agent types in group n.

Level iii. Here the analysis is concerned with the overall performance of the agent types in all groups. For example, there will be tests to determine whether altruistic agents perform better than greedy ones regardless of the composition of the groups.

$$\text{H3} \;:\; \mathsf{AU}_{Agent\ type\ 1}{=}...{=}\mathsf{AU}_{Agent\ type\ m}$$

where m refers to the total number of agent types being considered.

A.2 Different dispositions to cooperate

Section 7.3 (and in particular Figure 7.6) showed the relevance of the willingness to cooperate factor (ω) as a key element in the agents' decision making processes. However, it is clearly unrealistic in most settings to assume that all agents have the same degree of willingness to cooperate. Thus, this section explores the consequences of having communities of agents with different attitudes to cooperation (i.e. the agents have different values for ω).

Recalling the relevant definitions, a Greedy agent is one with $\omega > 1.0$, a Neutral agent is one with $\omega = 1.0$, and an Altruistic agent is one with $\omega < 1.0$. From this, a number of groups of agents can be modelled; those in which all members are of a specific type (as per Section 7.4), those in which the majority are of a particular type and those where no one type dominates. Against this background, a number of different agents communities were established (see Table A.1). Here the columns of the table indicate: the group reference number, the group label (or group description) and the details of the individual agents in the group [5].

To begin with, the following hypotheses of **level i** test the performance of the different groups:

H1: A mixed group of agents obtain the same AU as Altruistic, majority Altruistic, Greedy, majority Greedy, Neutral and majority Neutral groups. In shorter terms: H1 test if groups 1, 2, 3, 4, 5, 6 and 7 group obtain the same AU.

[5] The results are not affected by the ordering of the group. Thus, for example, in a given experiment it does not matter whether A_0 or A_4 is the Greedy one.

Group	Group label	Agent distribution
1	Neutral	All agents Neutral
2	Altruistic	All agents Altruistic
3	Greedy	All agents Greedy
4	Majority Neutral	A_0,A_1,A_3 Neutral, A_2 Altruistic, A_4 Greedy
5	Majority Altruistic	A_0,A_2,A_3 Altruistic, A_1 Greedy, A_4 Neutral
6	Majority Greedy	A_0 Altruistic, A_1 Neutral, A_2,A_3,A_4 Greedy
7	Mixed	A_0,A_3 Greedy, A_1, A_2 Altruistic, A_4 Neutral

Table A.1: Constituent agent groups (given ω).

Hypothesis to evaluate (level i)	p	Outcome
H1: $AU_1=AU_2=AU_3=AU_4=AU_5=AU_6=AU_7$	0.000	Rejected
H2: $TCT_1=TCT_2=TCT_3=TCT_4=TCT_5=TCT_6=TCT_7$	0.000	Rejected
Note: The subindex associated with AU and TCT refers to the group's number as described in Table A.1.		

Table A.2: Agents' performance per group (given ω): result of ANOVA.

H2: The number of CTs accomplished by a mixed group of agents is the same as Altruistic, majority Altruistic, Greedy, majority Greedy, Neutral and majority Neutral groups. In other words, this hypothesis tests whether groups 1, 2, 3, 4, 5, 6 and 7 accomplish the same number of CTs.

The result of ANOVA (Table A.2) shows a significant effect on the AU and TCT given the type of group (p value is 0.000 and H1 and H2 are rejected). A detailed analysis of this result is presented in Figure A.1. From this, it is possible to observe a correspondence between TCT and AU; the more TCTs, the better the AU. Focusing on the total AU obtained by group, the best performance is obtained with the Altruistic groups (2 and 5) and the worst with the Greedy ones (3 and 6). This result can be explained by the fact that agents' bids are highly affected by their ω factor (equation 4.6). Thus, the greedier an agent is, the higher the bids it submits, and the more difficult it is for them to be accepted. In contrast, the more altruistic an agent is, the greater its opportunity to get bids accepted. So, the Greedy group gets the lowest AU and the Altruistic one gets the highest. This observation is consistent with the findings of Section 7.4. However, these results do not show anything about the performance of the individual agents in the specific communities. For example, one important question to answer is whether Altruistic agents always perform better than Greedy ones?. To answer this, it is necessary to test ANOVA for each kind of agent in each group (**level ii** of detail). Thus, Table A.3 describes the level of willingness associated to each agent by group and Table A.4 presents the hypothesis tested for each group regarding an individual agent's AU.

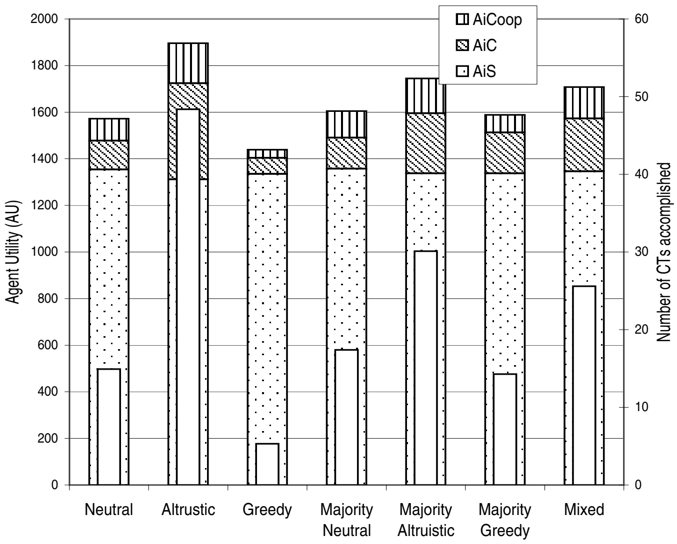


Figure A.1: Agents' role performance per group (given ω).

Group	Agent distribution
1	All agents $\omega = 1.00$
2	A_0 $\omega = 0.90$, A_1 $\omega = 0.40$, A_2 $\omega = 0.10$, A_3 $\omega = 0.60$, A_4 $\omega = 0.20$
3	A_0 $\omega = 2.50$, A_1 $\omega = 2.75$, A_2 $\omega = 1.25$, A_3 $\omega = 3.25$, A_4 $\omega = 1.50$
4	A_0 $\omega = 1.00$, A_1 $\omega = 1.00$, A_2 $\omega = 0.80$, A_3 $\omega = 1.00$, A_4 $\omega = 1.25$
5	A_0 $\omega = 0.75$, A_1 $\omega = 2.25$, A_2 $\omega = 0.25$, A_3 $\omega = 0.50$, A_4 $\omega = 1.00$
6	A_0 $\omega = 0.25$, A_1 $\omega = 1.00$, A_2 $\omega = 1.75$, A_3 $\omega = 2.50$, A_4 $\omega = 3.25$
7	A_0 $\omega = 1.75$, A_1 $\omega = 0.50$, A_2 $\omega = 0.30$, A_3 $\omega = 2.25$, A_4 $\omega = 1.00$

Table A.3: ω factor per agent and group.

Not surprisingly, most of the hypotheses of Table A.4 show that an agent's willingness factor has a significant effect on the AU obtained. The hypotheses accepted by ANOVA are for groups 1, 3 and 4 because agents in these groups obtain (statistically speaking) the same AU. In other words, it does not matter what type of members are in the community, the reward obtained by agents in groups Neutral, Majority Neutral and Greedy are broadly the same. In contrast, in groups 2, 5, 6 and 7 the hypotheses were rejected because some agents perform significantly better than others.

Form this, it is important to verify whether certain types of agent have real advantages over others regardless of the group. To this end, the fourth column in Table A.4 indicates the winner agent in each community; that is, the agent that

Hypothesis to evaluate (level ii)	**p**	**Outcome**	**Winner**
H1.1: AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}	0.535	Accepted	
H1.2: AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}	0.000	Rejected	A1 (Altruistic)
H1.3: AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}	0.188	Accepted	
H1.4: AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}	0.064	Accepted	
H1.5: AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}	0.000	Rejected	A0 (Altruistic)
H1.6: AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}	0.000	Rejected	A_2 (Greedy), A_1 (Neutral)
H1.7: AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}	0.000	Rejected	A4 (Neutral)
Note. Recall that the subindex in the hypothesis number is associated with the group tested, for example: H1.5 tests the AU in the group with majority Altruistic members (group 5).			

Table A.4: Agents' type performance within groups (given ω): result of ANOVA.

gained the most AU (i.e. a statistically superior AU than the others). This result highlights several interesting points. First, contrary to expectation, groups with the same type of agents (or with a majority of the same type) have dominant behaviour. For instance, hypotheses H1.2 and H1.5 were rejected by ANOVA. One reason for this result is that agents have associated ranges of ω values to be a particular type of agent and this factor has a direct effect on their decision making. Thus, though agents belong to the same type, their bids can have a significant difference in their values and, consequently, this can affect their performance (see the ω value per agent and group in Table A.3). So, this shows that ω does indeed affect the agents' cooperative attitudes. Second, Altruistic agents do not always perform the best. This is because their behaviour is highly dependent on the other members of the community. For example, in group 6 (with mainly Greedy agents) the Altruistic agent (A_0) performs the worst (table A.5). Once again, the justification for this is because agents are highly dependent on the others' cooperative attitudes. Thus, an Altruistic agent has a good performance when the other agents in the environment are Altruistic as well. Third, Greedy agents perform the worst unless the community is composed mainly of other Greedy agents (in which case the total gain is similarly distributed between the members (H1.3 was accepted)). Thus, Greedy A_2 obtained the best performance in group 6. The explanation, which in addition justifies the results discussed in previous points, is that the winner agents in the groups with rejected hypothesis (groups 2, 5, 6 and 7) are those with an ω value in the mid-point of the limits (i.e. those that balance the social attitude in the community). A_1, for example, has exactly the ω factor in the middle of the limits constituted by A2's and A0's ω in group 1. The same pattern is followed by groups 5, 6 and 7 where the winners are also in the middle of the limits. Thus, independently of the type of agents that make up the communities, the agent that balances its group and individual tendencies is the one with the best performance.

Agent	**AU**		
	1	2	3
A_0	1493.98		
A_4		1584.97	
A_3		1591.30	1591.30
A_2		1631.18	1631.18
A_1			1641.27
p	1.000	0.151	0.102

Table A.5: Group 6 (Majority Greedy): post-analysis.

The above analysis explained how the ω factor affected individual agent types in different types of groups. However, this next experiment seeks to evaluate if a more general dominant behaviour can be established between the agents' attitudes (**level iii** of detail). Being specific, hypothesis H3 evaluates whether there is a dominant behaviour with respect to an agent's ω factor:

H3: The AU gained by Altruistic agents is the same as that obtained by Greedy or Neutral agents in all groups.

Hypothesis to evaluate (level iii)	**p**	**Outcome**
H3: $AU_{Altruistic}=AU_{Greedy}=AU_{Neutral}$	0.000	Rejected

Table A.6: Agents' type performance in all groups (given ω): result of ANOVA.

ANOVA in Table A.6 rejects the hypothesis and the post-analysis (Table A.7) built three groups. Table A.7 indicates that Altruistic agents perform, in general, better than Neutral and Greedy ones. H3 does not contradict H1 and H2, it rather claims different things. The previous discussion (related to H1 and H2) outlined the agents' performance given the particular group they belonged to. This analysis was important because it is the only way of finding any relation between the AU obtained given the community. H3, in contrast, takes into consideration the whole sample (the 7 groups) and this is the basis on which the evaluation is performed. Thus, H3 demonstrates that because Altruistic agents perform better (recall that in group 6 this agent type was the worst) most of the time, when agents participate in several communities, they have a greater probability of obtaining a better performance on average than the other types. However, this result must be taken with caution because it takes into account the 7 groups used to evaluate the hypothesis. An alternative subset of groups may well lead to other results. This thesis has claimed that dynamism and openness are key determining factors in a multiagent setting. Thus, the number of groups to be taken into account cannot be determined in advance, neither can the type of members of each group be predetermined. But, when analysing particular groups with particular members, the probability of reproducing the results and obtaining the same agents' dominant behaviour presented by specific groups (for example those obtained with Altruistic agent in H1.5) is more likely to be obtained.

Agent type	**AU**		
	1	2	3
Greedy	1556.36		
Neutral		1624.89	
Altruistic			1776.26
p	1.000	1.000	1.000

Table A.7: Agents' type performance in all groups (given ω): post-analysis.

To summarise the analysis of heterogeneity regarding willingness to cooperate, the experimentation performed here explains the effect of ω on the agent's individual behaviour and on that of the group as a whole. It is clear that the agents' bids are effected by their social attitude (set by this factor) and this determines whether they expect to receive more, less or the same average reward for social activities as they do for non-social ones. Some types of agents perform better than others, but this is very much determined by the constituency of the remaining agents. Thus, there is no universally best willingness to cooperate factor. The same conclusion was also reached by the analysis performed in Section 7.4, however the results of this subsection extend this claim to more heterogeneous settings.

In more general terms, this analysis also shows one of the key problems in this MAS setting, the difficulty of obtaining optimal agent behaviour without having a mechanism to precisely model the degree of cooperativeness of the other agents. In this framework, the performance of the agents are highly based on their capacity to take decisions based on the proposed decision making framework, which, in part, is based on their ability to predict the others' attitudes. This analysis also shows that regarding ω, the better an agent can predict the others' attitudes, the more it can adapt its own behaviour in order to obtain a better performance.

A.3 Alternative values for decision procedures factor

One of the key elements in the decision making framework is the reward rate (r) that agents seeks to improve (recall that this considers the ST reward they would obtain assuming all agents have to move an average distance with equation (4.1)). Now, this is not the only way of taking this decision. Therefore, this subsection explores a range of alternatives. Thus, instead of using a fixed reward rate in all decision making procedures, the idea is to update this rate each time agents face a decision problem. In particular, the following means of calculating this reward rate were explored:

- Short reward rate. This is approximated based on the current goal and the expected time to achieve the goal. Note that this rate is calculated differently depending on the specific role the agent is playing. For AiC it is based on the particular CM it has selected, for AiCoop it is based on the contract it has agreed and for the AiS's it is based on the ST goal it is pursuing. For example, assume that AiS_2 received a request for cooperation in the particular situation illustrated in Figure 4.1. The Average reward rate used in that example is 0.625 (from equation (4.1)), while, the Short reward is calculated with $S = 2$ and the time it expects to reach ST_2 (which in this case is 4). Thus, the Short reward in this example is $2/4 = 0.5$.

- Long reward rate. This is the accumulated reward at the moment of the decision. In other words, this rate is calculated with the total reward achieved by STs and CTs and the timed invested on them. For example, assume that an agent has accomplished an AU of 100 in 120 times steps, the Long reward has a value of: $100/120 = 0.833$.

- Average reward rate. This is the case used in all the previous experiments (equation (4.1)). For example, in Figure 4.1 the Average reward value is 0.625.

Once again, the hypotheses are evaluated at the three levels of detail. The group reference, group label, and the agent's characteristics in each group are detailed in Table A.8. Starting with **level i**, the following hypotheses compare the agents' AU and TCT using each reward rate:

Group	Group label	Agent distribution
1	Short group	All agents with Short rate
2	Long group	All agents with Long rate
3	Average group	All agents with Average rate
4	Majority Short	A_0,A_1,A_2 Short, A_3 Average, A_4
5	Majority Long	A_0 Average, A_1,A_3,A_4 Long, A_2 Short
6	Majority Average	A_0,A_1,A_3 Average, A_2 Long, A_4 Short
7	Mixed	A_0,A_3 Average, A_1 Short, A_2,A_4 Long

Table A.8: Constituent agent groups (given r).

H1: Groups of agents using Short and majority Short reward rate obtain the same AU as groups using Long and majority Long, Average and majority Average and the same as groups with mixed agents.

H2: The number of CTs accomplished by groups of agents employing Short and majority Short reward rate is the same as those obtained by groups of agents using Long and majority Long, Average and majority Average and the same as groups with mixed reward rates.

Hypothesis to evaluate (level i)	p	Outcome
H1: AU_1=AU_2=AU_3=AU_4=AU_5=AU_6=AU_7	0.000	Rejected
H2: TCT_1=TCT_2=TCT_3=TCT_4=TCT_5=TCT_6=TCT_7	0.000	Rejected
Note: The subindex associated with AU and TCT refers to the group's number as described in Table A.8.		

Table A.9: Agents' performance per group (given r): result of ANOVA.

As shown in Table A.9, the reward rate used by agents does indeed have implications on the AU and the number of CTs accomplished (H4 and H5 are rejected with ANOVA). This result can be explained by considering the nature of the agents' decision making processes. The reward rate is used by agents when they submit bids (equation (4.6)), when they evaluate the CM by measuring the cost of setting it up (equation (4.2)), and when they estimate how much the other agents are likely to ask for work in cooperation (equation (4.3)). In the case of submitting bids, when agents use Short reward they will propose higher or smaller bids than when they use Average or Long rewards depending on whether they are closer or further away from their goals (respectively). In the former situation, their bids are more difficult to get accepted because with Short reward agents seek to improve upon the current goal they are pursuing. However if the action succeeds, AiCoops gain a very good reward. On the other hand, when they are closer (below the average distance) their bids often get accepted. Thus, with Short reward agents are not consistent in getting their bids accepted and, consequently, they accomplish fewer CTs than the other rates. To this end, Average and Long agents achieve more CTs because it is more likely they will have their bids accepted (at least most of the time) and hence they obtain more reward in total. The difference between the use of Long and Average rates is mainly that the former tends to be more precise about what is occurring because it is updated as time passes. This contrasts with Average which never updates its approximation no matter whether this joint action is successful or not (because it is based on a fixed calculation). Consistent with the previous explanation, Table A.10 presents two conclusions: the Long rate group (group 2) perform the best and the Short rate ones (groups 1 and 4) perform the worst. With respect to the other communities, it is not possible to conclude anything because the results are highly contingent on the composition of the communities. For example, although the groups of majority Long (group 5) perform the second best, its dominance is not statistically significant over groups of mixed agents and Average.

Group	**AU**			
	1	2	3	4
1	1469.80			
4	1479.40			
6		1519.48		
7		1529.16	1529.16	
3		1530.04	1530.04	
5			1545.58	
2				1579.28
p	0.923	0.883	0 472	1.000

Table A.10: Agents' performance per group (given r): post-analysis.

To support the conclusion that, in general, Long agents dominate over the other two types, H3 tests the same hypothesis but, this time, consideres the AU obtained by an agent regardless of the groups to which it belongs (**level iii**):

H3: The AU gained by agents which use Short reward is the same as that obtained by agents which use Long or Average rewards in all groups.

Tables A.11 and A.12 confirm the above discussion; agents that take Long term rewards perform better than the others. However, the H3 result has the same

Hypothesis to evaluate (level iii)	**p**	**Outcome**
H3: $\mathsf{AU}_{\mathsf{Short}}=\mathsf{AU}_{\mathsf{Long}}=\mathsf{AU}_{\mathsf{Average}}$	0.000	Rejected

Table A.11: Agents' type performance in all groups (given r): result of ANOVA.

Agent type	**AU**		
	1	2	3
Short	1482.97		
Average		1524.64	
Long			1554.60
p	1.000	1.000	1.000

Table A.12: Agents' type performance in all groups (given r): post-analysis.

drawback as hypothesis H3 of Section A.2; namely, it is partial. ANOVA does not take the whole population into account, rather it considers only a sample of 7 groups (though this sample represents a good range of the possible combinations of groups it is not the whole population). Thus, the ANOVA result regarding H3 cannot be generalised.

Despite the dominance achieved by Long agents in Table A.10, it is necessary to investigate whether they show the same winner performance by focusing in on how the groups are constituted. To deal with this issue, Table A.13 shows the experimental evaluation for each type of agent in each type of group of agent (**level ii**). The ANOVA results indicates that no matter what the agent's community is, there is no truly dominant behaviour. This is somewhat surprising. It was expected that Long reward agents would obtain a better AU than Short and Average agents because it had shown a dominant behaviour in the previous analysis at the various different levels of detail. The explanation is that the reward rates are used in all the decision making procedures and some of these decisions benefit specific roles and some benefit others. For example, with Short reward, AiC performs poorly because it assumes that others (equation (4.3)) will require a significant reward and hence it does not go for CTs very often. This shows that some roles perform better than others based on the particular reward rate. In the end, the use of different ways of calculating this reward means that different member types in the various communities do not really have a distinctive performance pattern. In other words, no matter how each group is integrated and which reward rate is used, there is not a significant difference in the individual performance. Thus, for example, in group 7 all type of agents obtain (statistical) the same amount of reward although the members use different reward rates.

Hypothesis to evaluate (level ii)	p	Outcome
H1.1: $AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}$	0.816	Accepted
H1.2: $AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}$	0.630	Accepted
H1.3: $AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}$	0.655	Accepted
H1.4: $AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}$	0.276	Accepted
H1.5: $AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}$	0.122	Accepted
H1.6: $AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}$	0.378	Accepted
H1.7: $AU_{A0}=AU_{A1}=AU_{A2}=AU_{A3}=AU_{A4}$	0.080	Accepted
Note. Recall that the subindex in the hypothesis number is associated with the group tested. For example: H1.3 tests the AU in the group with only Average agents (group 3).		

Table A.13: Agents' type performance within groups (given r): result of ANOVA.

The above discussion demonstrates that regardless of the fact that Long dominantes in some experiments, agents using the Long reward rate do not obtain significantly better long term rewards. Although Long reward agents represent more precisely the real reward they are gaining, this does not make a difference in the final AU obtained. Thus, the benefits of using the reward rate types explored here are only significant from the group's points of view, not from the individual's perspective.

One potential problem with the experimentation performed here is that the same reward rate is used in all the agent's decision procedures; namely how to bid, which CM to select and so on. However, it does not need to be this way. Thus, based on the nature of each type of decision, agents could use different reward rates. For example, when deciding which CM to select, agents could use Long reward, but when bidding they might use Short reward. However, this is left as future work.

To sum up, the main conclusion related to the matter of reward rate is that the decision of using Average reward rate as defined in equation (4.1) in all the experiments in the main body of the thesis was correct. Thus, unless the aim is to evaluate the particulars of agent behaviours given the whole range of communities, the reward rate does not affect the conclusions already drawn.

A.4 Different reasoning capabilities

The last scenario to consider is what happens when learning agents share the environment with non learning ones. While Sections A.2 and A.3 compare and contrast the agents' behaviour by only changing some aspects of the decision making process, here the comparison tests heterogeneity in more fundamental way. The question is whether agents that learn about the CM to select have a dominant behaviour over those that do not.

As before, the experiments in this section were conducted as per Section A.1. However, the main distinction is that it only describes the dynamic setting (environment scenario2 in Chapter 9) because it is in this context that more interesting conclusions are likely. In particular, here some agents posses the learning ability (as detailed in Section 9.2). These are called RL agents. The agents that do not learn are called NL agents (they use the decision making framework as indicated in Chapter 4). Thus, Table A.14 introduces the groups' reference number, the group label and the members of each group used in this experiment.

Group	**Group label**	**Agent distribution**
1	NLs	All members are non learning agents
2	RLs	All members are learning agents
3	Majority NLs	A_0 RL, A_1 NL, A_2 NL, A_3 NL, A_4 NL
4	Majority RLs	A_0 NL, A_1 RL, A_2 RL, A_3 RL, A_4 RL
5	Mixed RLs	A_0 RL, A_1 NL, A_2 RL, A_3 NL, A_4 RL
6	Mixed NLs	A_0 NL, A_1 RL, A_2 NL, A_3 RL, A_4 NL

Table A.14: Constituent agent groups (given RL and NL agents).

Following the experimental setting of Section A.1, the first hypotheses to test are whether the groups of Table A.14 perform the same (**level i** of detail):

H1: Groups of RLs and majority RLs agents obtain the same AU as groups of NLs and majority NLs agents and the same as groups of mixed RLs and NLs agents.

H2: The number of CTs accomplished by groups of RLs and majority RLs agents is the same as groups of NLs, majority NLs, and as the mixed RL and NL agents.

Tables A.15 and A.16 present the ANOVA results and the post analysis performed on the data collected in these experiments. As expected, H1 and H2 were rejected with ANOVA. Group 2 (with all learning agents) performs the best and group 1 (with non learning agents) performs the worst. This result corroborates the fact that agents that learn the situation in which the CM should be applied

Hypothesis to evaluate (level i)	**p**	**Outcome**
H1: $\mathsf{AU}_1=\mathsf{AU}_2=\mathsf{AU}_3=\mathsf{AU}_4=\mathsf{AU}_5=\mathsf{AU}_6$	0.000	Rejected
H2: $\mathsf{TCT}_1=\mathsf{TCT}_2=\mathsf{TCT}_3=\mathsf{TCT}_4=\mathsf{TCT}_5=\mathsf{TCT}_6$	0.000	Rejected
Note: The subindex associated with AU and TCT refers to the group's number as described in Table A.14.		

Table A.15: Agents' performance per group (given RL and NL agents): result of ANOVA.

are more effective (Section 9.2). However, note that in both groups the members are similar and it is necessary to check the group's performance when there are a mixed set of members. Here the interesting result is that the more learning agents are in a group, the better the overall performance (Table A.16).

Group	**AU**		
	1	2	3
1	7151.12		
3	7151.28		
6	7285.08	7285.08	
5	7436.36	7436.36	7436.36
4		7562.44	7562.44
2			7696.78
p	0.170	0.196	0.259

Table A.16: Agents' performance per group (given RL and NL agents): post-analysis.

Hypothesis to evaluate (level ii)	**p**	**Outcome**	**Winner**
H1.1: $\mathsf{AU}_{A0}=\mathsf{AU}_{A1}=\mathsf{AU}_{A2}=\mathsf{AU}_{A3}=\mathsf{AU}_{A4}$	0.476	Accepted	
H1.2: $\mathsf{AU}_{A0}=\mathsf{AU}_{A1}=\mathsf{AU}_{A2}=\mathsf{AU}_{A3}=\mathsf{AU}_{A4}$	0.209	Accepted	
H1.3: $\mathsf{AU}_{A0}=\mathsf{AU}_{A1}=\mathsf{AU}_{A2}=\mathsf{AU}_{A3}=\mathsf{AU}_{A4}$	0.000	Rejected	A_0 (RL)
H1.4: $\mathsf{AU}_{A0}=\mathsf{AU}_{A1}=\mathsf{AU}_{A2}=\mathsf{AU}_{A3}=\mathsf{AU}_{A4}$	0.000	Rejected	A_2, A_4, A_3, A_1 (RLs)
H1.5: $\mathsf{AU}_{A0}=\mathsf{AU}_{A1}=\mathsf{AU}_{A2}=\mathsf{AU}_{A3}=\mathsf{AU}_{A4}$	0.000	Rejected	A_0,A_2,A_4 (RLs)
H1.6: $\mathsf{AU}_{A0}=\mathsf{AU}_{A1}=\mathsf{AU}_{A2}=\mathsf{AU}_{A3}=\mathsf{AU}_{A4}$	0.000	Rejected	A_1, A_3 (RLs)
Note. Recall that the subindex in the hypothesis number is associated with the group tested, for example: H1.2 tests the AU in the group with RLs members (group 2).			

Table A.17: Agents' type performance within groups: result of ANOVA (given RL and NL agents).

Turning to the agents' type performance within groups (**level ii**), the hypothesis now validates RL agents' performance per group. To this end, Table A.17 clearly indicates that those groups that are composed of the same agent type (groups 1 and 2) do not have a particular distinctive performance (ANOVA was accepted in H1.1 and H1.2). However, the interesting point is that ANOVA rejects the AU significance in groups 3, 4, 5 and 6. This is because in these groups there is a mix of member types. The results indicate that the groups composed of the RL agents are those that perform best. However, now it is interesting to verify if the hypothesis is rejected because RL agents do have a significantly better performance than NL ones. For this purpose, the post-analysis of hypotheses H1.3, H1.4, H1.5 and H1.6 are shown in Tables A.18 and A.19. The results are conclusive; regardless of the group, RL agents invariably obtain more AU than NL agents.

Post analysis H1.3		
Agent	**AU**	
	1	2
A_4	6678.81	
A_3	6936.20	
A_2	7034.06	
A_1	7139.00	
A_0		7968.31
p	0.264	1.000

Post analysis H1.4		
Agent	**AU**	
	1	2
A_0	6691.27	
A_2		7750.99
A_4		7752.50
A_3		7764.69
A_1		7852.73
p	1.000	0.950

Table A.18: Agents' type performance within groups 3 and 4: post-analysis.

Post analysis H1.5		
Agent	**AU**	
	1	2
A_3	6769.96	
A_1	6955.69	
A_2		7805.74
A_4		7812.48
A_0		7837.90
p	0.909	1.000

Post analysis H1.6		
Agent	**AU**	
	1	2
A_2	6761.54	
A_0	6776.75	
A_4	7002.84	
A_3		7917.54
A_1		7966.70
p	0.757	0.999

Table A.19: Agents' type performance within groups 5 and 6: post-analysis.

Given these results, it is expected that the **level iii** of analysis would corroborate the significantly better performance of RL agents with the following hypothesis:

H3 : The AU gained by RL agents is the same as that obtained by NL agent in all communities.

Hypothesis to evaluate (level iii)	p	Outcome
H3: AU_{RL}=AU_{NL}	0.001	Rejected

Table A.20: Agents' performance in all groups (given RL and NL agents): result of ANOVA.

Table A.20 shows that RL agent do have a (statistically significant) better AU than agents that do not learn at all. The same conclusion can be drawn in Figure A.2 in which the AU obtained by the RL agent was approximately 7260.82 whereas the value of the NL agent was 7500.20.

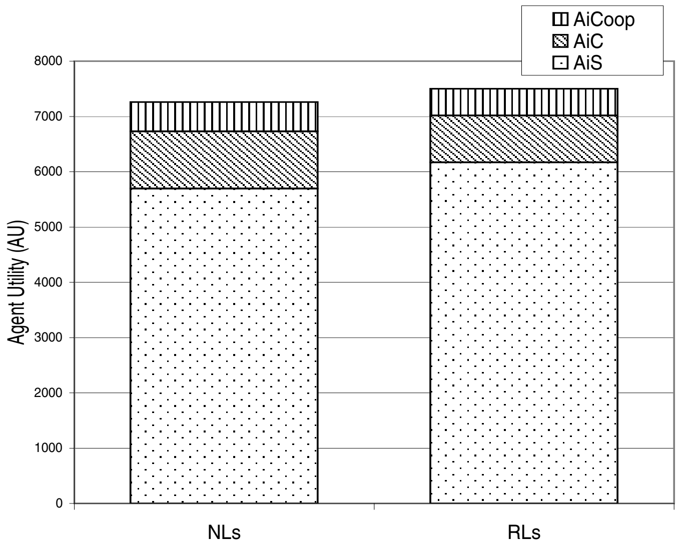


Figure A.2: Agents' type performance in all groups (given RL and NL agents).

A.5 Discussion

The experimental evaluation of this Appendix has further evaluated the performance of the agents when various forms of heterogeneity are introduced. In general terms, a fundamental exploration at three levels of study was carried out focusing on i) different dispositions to cooperate, ii) alternative values for calculating the reward rate (r) factor and iii) different reasoning capabilities (namely; learning and non learning agents). To this end, the results obtained here support, in general, the conclusions drawn in the main body of this thesis. In particular, the willingness to cooperate experiments show that the ω factor does indeed effect the agent's individual disposition to cooperate and consequently on the group as a whole. Moreover, it was also shown that there is no universal ω factor that is effective in all settings. How well or badly an agent performs is highly based on the attitudes of the other agents. Regarding the different methods for calculating r, it was demonstrated that no matter how this factor is calculated, there is no significant difference in the long term performance of the agents. Finally, the most clear out results related to the agent's effectiveness was when comparing learning and non learning agents. Undoubtedly, agents that take advantage of learning by experience perform better than those that do not do so. Once again, this result corroborates the conclusions of Chapter 9.

References

Axelrod, R. M. (1985). *The Evolution of Cooperation*. Basic Books: New York, NY.

Baccala, B. e. (1997). Connected: An internet encyclopedia (third edition). http://www.freesoft.org/CIE/.

Barber, K. S., Han, D. C., & Liu, T. H. (2000). Coordinating distributed decision making using reusable interaction specifications. In *Design and Applications of Intelligent Agents: Third Pacific Rim International Workshop on Multi-Agents (PRIMA 2000)*, pp. 1–15 Melbourne, Australia.

Barbuceanu, M., Gray, T., & Mankovski, S. (1998). Coordinating with obligations. In Sycara, K. P., & Wooldridge, M. (Eds.), *Proceedings of the Second International Conference on Autonomous Agents (AGENTS'98)*, pp. 62–69 Minneapolis, MN, USA. ACM Press, New York, NY, USA.

Bond, A. H., & Gasser, L. (1988a). An analysis of problems and research in DAI. In *Readings in Distributed Artificial Intelligence* (Bond & Gasser, 1988b), Chapter 1, pp. 3–35.

Bond, A. H., & Gasser, L. (Eds.). (1988b). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers: San Mateo, CA.

Bourne, R. A., Excelente-Toledo, C. B., & Jennings, N. R. (2000). Run-time selection of coordination mechanisms in multi-agent systems. In Horn, W. (Ed.), *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000)*, Vol. 54, pp. 348–352 Berlin, Germany. IOS Press, Amsterdam, The Netherlands.

Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conference on*

Artificial Intelligence (IJCAI-99), pp. 478–485 Stockholm, Sweden. Morgan Kaufmann publishers Inc.: San Mateo, CA.

Briggs, W., & Cook, D. (1995). Flexible social laws. In Pollack, M. E. (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pp. 688–693 Montreal, Canada. Morgan Kaufmann publishers Inc.: San Mateo, CA.

Burt, R. S. (1982). Network structure: The social context. In *Toward a Structural Theory of Action. Network models of Social Structure, Perception, and Action*, Chapter 2, pp. 19–93. Academic Press: New York, NY.

Carley, K. M., & Gasser, L. (1999). Computational organization theory. In Weiss, G. (Ed.), *Multiagent Systems: A Modern Approach To Distributed Artificial Intelligence*, Chapter 7, pp. 299–330. The MIT Press, Cambridge, MA.

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In Rich, C., & Mostow, J. (Eds.), *Proceedings of Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pp. 746–752 Madison, MI. Menlo Park, CA: AAAI Press.

Clement, B. J., & Durfee, E. H. (1999a). Theory for coordinating concurrent hierarchical planning agents using summary information. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pp. 495–502 Orlando, Florida. Menlo Park, CA: AAAI Press.

Clement, B. J., & Durfee, E. H. (1999b). Top-down search for coordinating the hierarchical plans of multiple agents. In Etzioni, O., Müller, J. P., & Bradshaw, J. M. (Eds.), *Proceedings of the Third International Conference on Autonomous Agents (AGENTS'99)*, pp. 252–259 Seattle, Washington. ACM Press.

Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, *42(2-3)*, 213–261.

Cohen, P. R. (1995). *Empirical Methods for Artificial Intelligence*. The MIT Press: Cambridge, MA.

Corkill, D. D. (1979). Hierarchical planning in a distributed problem solving environment. In *Proceedings of the Sixth International Joint Conference on Artificial ntelligence (IJCAI-79)*, pp. 168–175 Tokio, Japan. Morgan Kaufmann publishers Inc.: San Mateo, CA.

Decker, K., Pannu, A., Sycara, K., & Williamson, M. (1997). Designing behaviors for information agents. In Johnson, W. L., & Hayes-Roth, B. (Eds.), *Proceedings of the First International Conference on Autonomous Agents (AGENTS'97)*, pp. 404–412 Marina del Rey, CA, USA. ACM Press, New York, NY.

Decker, K. S. (1995). *Environment Centered Analysis and Design of Coordination Mechanisms*. Ph.D. thesis, Department of Computer Science, University of Massachusetts, Amherst.

Decker, K. S., & Lesser, V. R. (1997). Designing a family of coordination algorithms. In Huhns, & Singh (Huhns & Singh, 1997), Chapter 4. Models of Agency.

Decker, K. S., Sycara, K., & Williamson, M. (1997). Middle-agents for the internet. In Pollack, M. E. (Ed.), *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pp. 578–583 Nagoya, Japan. Morgan Kaufmann Publishers.: San Mateo, CA.

deRoure, D., Baker, M. A., Jennings, N. R., & Shadbolt, N. (2003). The evolution of the grid. *Concurrency and Computation: Practice and Experience*, *15*(To appear).

Durfee, E. H. (1999a). Distributed problem solving and planning. In Weiss (Weiss, 1999), Chapter 3, pp. 121–164.

Durfee, E. H. (1999b). Practically coordinating. *AI Magazine*, *20*(1), 99–116.

Durfee, E. H. (2001). Scaling up agent coordination strategies. *IEEE Computer*, *34*(7), 39–46.

Durfee, E. H., & Lesser, V. R. (1991). Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(5), 1167–1183.

Durfee, E. H., Lesser, V. R., & Corkill, D. D. (1989). Trends in cooperative distributed problem solvers. *IEEE Transactions on Knowledge and Data Engineering*, *1*, 63–83.

Durfee, E. H., & Montgomery, T. A. (1991). Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(6), 1363–1378.

Ephrati, E., & Rosenschein, J. S. (1994). Divide and conquer in multi-agent planning. In Hayes-Roth, B., & Korf, R. E. (Eds.), *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pp. 375–380 Seattle, Washington. Menlo Park, CA: AAAI Press.

Excelente-Toledo, C. B., Bourne, R. A., & Jennings, N. R. (2001). Reasoning about commitments and penalties for coordination between autonomous agents. In Müller, J. P., Andre, E., Sen, S., & Frasson, C. (Eds.), *Proceedings of the Fifth International Conference on Autonomous Agents (AGENTS'01)*, pp. 131–138 Montreal, Canada. ACM Press.

Excelente-Toledo, C. B., & Jennings, N. R. (2002). Learning to select a coordination mechanism. In Castelfranchi, C., & Johnson, W. L. (Eds.), *Proceedings of First International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS'02)*, pp. 1106–1113 Bologna, Italy. ACM Press.

Excelente-Toledo, C. B., & Jennings, N. R. (2003a). The dynamic selection of coordination mechanisms. *Journal of Autonomous Agents and Multi-Agent Systems*, *Submitted*.

Excelente-Toledo, C. B., & Jennings, N. R. (2003b). Learning when to coordinate. In *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03)* Tlaxcala, Mexico. IEEE Computer Society Press.

Filar, J. A. (1997). *Competitive Markov decision processes*. Springer-Verlag: New York, NY.

Fisher, K., Müller, J. P., & Pischel, M. (1996). AGenDA-A general testbed for DAI applications. In O'Hare, & Jennings (O'Hare & Jennings, 1996), Chapter 15, pp. 401–427.

Fox, M. S. (1981). An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, *11*(1), 70–80.

Galbraith, J. (1973). *Designing Complex Organizations*. Addison-Wesley Publishing Company, Inc. Reading, MA.

Garvey, A., & Lesser, V. (1993). Design-to-time real-time scheduling. *IEEE Transactions on Systems, Man and Cybernetics, Special Issue on Planning, Scheduling and Control*, *23*(6), 1491–1502.

Hanks, S., Pollack, M. E., & Cohen, P. R. (1993). Benchmarks, test beds, controlled experimentation, and the design of agent architectures. *AI Magazine*, *14*(4), 17–26.

He, M., Jennings, N. R., & Leung, H.-f. (2003). On agent-mediated electronic commerce. *IEEE Transactions on Knowledge and Data Engineering*, *16*. To appear.

Hendler, J. (2003). Science and the semantic web. *Science Magazine: Policy Forum Communication*, *299*(5606), 520–521.

Hogg, L. M. J., & Jennings, N. R. (2001). Socially intelligent reasoning for autonomous agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part A*, *31*(5), 381–393.

Hu, J., & Wellman, M. P. (1998). Online learning about other agents in a dynamic multiagent system. In Sycara, K. P., & Wooldridge, M. (Eds.), *Proceedings of Second International Conference on Autonomous Agents (AGENTS'98)*, pp. 239–246 Minneapolis, MN. ACM Press, New York, NY.

Huberman, B. A., & Hogg, T. (1995). Distributed computation as an economic system. *The Journal of Economic Perspectives*, *9*(1), 141–152.

Huhns, M., & Singh, M. P. (Eds.). (1997). *Readings in Agents*. Morgan Kaufmann Publishers: San Mateo, CA.

Huhns, M. N., & Stephens, L. M. (1999). Multiagent systems and society of agents. In Weiss, G. (Ed.), *Multiagent Systems: A Modern Approach To Distributed Artificial Intelligence*, Chapter 2, pp. 79–120. The MIT Press, Cambridge, MA.

Jennings, N. R. (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, *8*(3), 223–250.

Jennings, N. R. (1996). Coordination techniques for distributed artificial intelligence. In O'Hare, & Jennings (O'Hare & Jennings, 1996), Chapter 6, pp. 187–210.

Jennings, N. R. (1998). A roadmap of agent research and developement. *Autonomous Agents and Multi-Agent Systems*, *1*(1), 7–38.

Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, *44*(4), 35–41.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., & Wooldridge, M. (2001). Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, *10*(2), 199–215.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*(4), 237–285.

Kinny, D. N., & Georgeff, M. P. (1991). Commitments and effectiveness of situated agents. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 82–88 Sydney, Australia. Morgan Kaufmann publishers Inc.: San Mateo, CA.

Kraus, S. (1993). Agents contracting tasks in non-collaborative environments. In Fikes, R., & Lehnert, W. (Eds.), *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pp. 243–248 Washington, D.C. Menlo Park, CA: AAAI Press.

Kraus, S., Nirkhe, M., & Sycara, K. (1993). Reaching agreements through argumentation: a logical model (preliminary report). In *Proceedings of the 12th International Workshop on Distributed Artificial Intelligence*, pp. 233–247 Hidden Valley, Pennsylvania.

Labrou, Y., Finin, T., & Peng, Y. (1999). Agent communication languages: The current landscape. *IEEE Intelligent Systems*, *14*(2), 45–52.

Lane, D. M. (2001). Hyperstat online textbook. http://davidmlane.com/ hyperstat/. January/2003.

Lesser, R. V. (1999). Cooperative multiagent systems: a personal view of the state of the art. *IEEE Transactions On Knowledge and Data Engineering*, *11*(1), 133–142.

Lesser, V., Decker, K. S., Wagner, T., Carver, N. F., Garvey, A., Horling, B., Neiman, D. E., Podorozhny, R., NagendraPrasad, M., Raja, A., Vincent, R., Xuan, P., & Zhang, X. (2002). Evolution of the GPGP/TAEMS domain-independent coordination framework. Tech. rep. 02-03, University of Massachusetts/Amherst Computer Science Technical Report.

Lesser, V. R. (1998). Reflections of the nature of multi-agent coordination and its implications for an agent architecture. *Autonomous Agents and Multi-Agent Systems*, *1*(1), 89–111.

Lesser, V. R., & Corkill, D. D. (1983). The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, *4*(3), 15–33.

Littmann, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning (ICML-94)*, pp. 157–163 San Francisco, CA. Morgan Kaufmann.

Luck, M., McBurney, P., & Preist, C. (Eds.). (2003). *Agent Technology: Enabling Next Generation Computing. A Roadmap for Agent Based Computing. (Version 1.0).* http://www.agentlink.org/.

Malone, T. W. (1987). Modeling coordination in organizations and markets. *Management Science*, *33*(10), 1317–1332.

McGuire, J., Pelavin, R., Kuokka, D. R., Weber, J. C., M., T. J., Gruber, T. R., & Olsen, G. R. (1993). Shade: A medium for sharing design knowledge among engineering tools. *Concurrent Engineering: Research and Applications*, *3*(1).

Miles, S., Joy, M., & Luck, M. (2002). Towards a methodology for coordination mechanisms selection in open systems. In *Third International Workshop on Engineering Societies in the Agents World (ESAW-2002).*

Mitchel, T. (1997a). *Machine Learning*. McGraw-Hill.

Mitchel, T. (1997b). Reinforcement learning. In *Machine Learning* (Mitchel, 1997a), Chapter 13, pp. 366–390.

Moulin, B., & Chaib-Draa, B. (1996). An overview of distributed artificial intelligence. In O'Hare, & Jennings (O'Hare & Jennings, 1996), Chapter 1, pp. 3–55.

Müller, J. H. (1996). Negotiation principles. In O'Hare, & Jennings (O'Hare & Jennings, 1996), Chapter 7.

Nagayuki, Y., Ishii, S., & Doya, K. (2000). Multi-agent reinforcement learning: An approach based on the other agent's internal model. In Durfee, E. H.

(Ed.), *Proceedings on the Fourth International Conference on Multi-Agent Systems (ICMAS-00)*, pp. 215–221 Boston, MA. IEEE Computer Society.

Norman, T. J., Sierra, C., & Jennings, N. R. (1998). Rights and commitments in multi-agent agreements. In Demazeau, Y. (Ed.), *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98)*, pp. 222–229 Paris, France. IEEE Computer Society Press.

Norman, T. J., & Jennings, N. R. (1998). Generating states of joint commitment between autonomous agents. *Agents and Multi-Agent Systems: Formalisms, Methodologies, and Applications. Lecture Notes in Artificial Intelligence*, *1441*, 123–133.

O'Hare, G. M. P., & Jennings, N. R. (Eds.). (1996). *Foundations of Distributed Artificial Intelligence*. John Wiley & Son, Inc. New York, NY.

Ott, R. L., & Mendenhall, W. (1995). *Undertanding statistics*. Duxbury Press: Melmont, CA.

Pappachan, P. M., & Durfee, E. H. (2000). A multiagent plan evaluation heuristic for real-time coordination. Personal communication.

Parunak, H. V. D. (1999). Industrial and practical applications of dai. In Weiss (Weiss, 1999), Chapter 9, pp. 377–421.

Parunak, H. V. D. (2000). A practitioners' review of industrial agent applications. *Autonomous Agents and Multi-Agent Systems*, *3*(4), 389–407.

Prasad, M. V. N., & Lesser, V. R. (1996). Off-line learning of coordination in functionally structured agents for distributed data processing. In *Workshop on Learning, Interaction and Organizations in Multiagent Environments, ICMAS96*. Menlo Park, CA: AAAI Press.

Prasad, M. V. N., & Lesser, V. R. (1999). Learning situation-specific coordination in cooperative multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, *2*(2), 173–207.

Raiffa, H. (1982). *The Art and Science of Negotiation*. Harvard University Press: Cambridge, MA.

Ramamritham, K., Stankovic, J. A., & Zhao, W. (1989). Distributed sheduling of tasks with deadlines and resource requirements. *IEEE Transactions on Computers*, *38*(8), 1110–1123.

Rosenschein, J. S., & Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. The MIT Press: Cambridge, MA.

Russell, S. J., & Norvig, P. (1995a). *Artificial Intelligence: A Modern Approach*. Prentice Hall: Upple Saddle River, NJ.

Russell, S. J., & Norvig, P. (1995b). Intelligent agents. In *Artificial Intelligence: A Modern Approach* (Russell & Norvig, 1995a), Chapter 2, pp. 31–52.

Russell, S. J., & Norvig, P. (1995c). Learning. In *Artificial Intelligence: A Modern Approach* (Russell & Norvig, 1995a), Chapter 18,19,20,21, pp. 524–648.

Russell, S. J., & Norvig, P. (1995d). Making simple and complex decisions. In *Artificial Intelligence: A Modern Approach* (Russell & Norvig, 1995a), Chapter 16-17, pp. 471–522.

Russell, S. J., & Norvig, P. (1995e). Reinforcement learning. In *Artificial Intelligence: A Modern Approach* (Russell & Norvig, 1995a), Chapter 20, pp. 598–624.

Sandholm, T. W. (2001). Leveled commitment constracts and strategic breach. *Games and Economic Behaviuor*, *35*, 212–270.

Sandholm, T. W. (1999). Distributed rational decision making. In Weiss (Weiss, 1999), Chapter 5, pp. 201–258.

Sandholm, T. W., & Lesser, V. R. (1995). Issues in automated negotiation and electronic commerce: Extending the contract net protocol. In Lesser, V., & Gasser, L. (Eds.), *Proceedings of the First International Conference in Multiagent Systems (ICMAS-95)*, pp. 328–335 San Francisco, CA. The MIT Press.

Sandholm, T. W., & Lesser, V. R. (1996). Advantages of a leveled commitment contracting protocol. In Clancey, W. J., & Weld, D. (Eds.), *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 126–133 Portland, Oregon. Menlo Park, CA: AAAI Press.

Sandholm, T. W., Sikka, S., & Norden, S. (1999). Algorithms for optimizing leveled commitment contracts. In Dean, T. (Ed.), *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pp. 535–540 Stockholm, Sweden. Morgan Kaufmann publishers Inc.: San Mateo, CA.

Sen, S., & Durfee, E. H. (1994). The role of commitment in cooperative negotiation. *International Journal on Intelligent & Cooperative Information Systems*, *3*(1), 67–81.

Sen, S., Sekaran, M., & Hale, J. (1994). Learning to cooperate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pp. 426–431 Amherst, MA. Menlo Park, CA: AAAI Press.

Sen, S., & Weiss, G. (1999). Learning in multiagent systems. In Weiss (Weiss, 1999), Chapter 6, pp. 259–298.

Shen, W., & Barthes, J.-P. (1995). Dide: A multi-agent environment for engineering design. In Lesser, V., & Gasser, L. (Eds.), *Proceedings on the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 344–351 San Francisco, CA, USA. The MIT Press.

Shoham, Y., & Tennenholtz, M. (1992). On the synthesis of useful social laws for artificial agent societies. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pp. 276–281 San Jose, California. Menlo Park, CA: AAAI Press.

Shoham, Y., & Tennenholtz, M. (1995). On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, *73*(1-2), 231–252.

Smith, R. G., & Davis, R. (1981). Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, *11*(1), 61–70.

Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, *3*(8), 345–383.

Sugawara, T., & Lesser, T. (1998). Learning to improve coordinated actions in cooperative distributed problem-solving environments. *Machine Learning*, *33*(2/3), 129–153.

Sutton, R. S., & Barto, G. A. (1998). *Reinforcement Learning: an introduction.* The MIT Press: Cambridge, MA.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pp. 330–337 Amherst, MA.

Vidal, J. M., & Durfee, E. H. (1997). Agents learning about agents: A framework and analysis. In *Collected papers from the AAAI-97 workshop on Multiagent Learning* Providence, Rhode Island.

Watkins, C. J. C. H., & Dayan, P. (1992). Technical note: Q-learning. *Machine Learning*, *8*, 279–292.

Weiss, G. (Ed.). (1999). *Multiagent Systems: A Modern Approach To Distributed Artificial Intelligence.* The MIT Press: Cambridge, MA.

Weiss, G., & Dillenbourg, P. (1999). What is multi in multiagent learning. In Dillenbourg, P. (Ed.), *Collaborative learning: Cognitive and Computational approaches*, Chapter 4, pp. 64–80. Pergamon Press.

Wellman, M. P. (1993). A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, *1*, 1–23.

Wooldridge, M. (2001). *An Introduction to Multiagent Systems.* John Wiley & Sons Ltd: Chichester, England.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, *10*(2), 114–152.

Wurman, P. R., Wellman, M. P., & Walsh, W. E. (1998). The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In Sycara, K. P., & Wooldridge, M. (Eds.), *Proceedings of the Second International Conference on Autonomous Agents (AGENTS'98)*, pp. 301–308 Minneapolis, MN. ACM Press, New York, NY.

Wurman, P. R., Wellman, M. P., & Walsh, W. E. (2001). A parametrization of the auction design space. *Games and Economic Behavior*, *35*(1-2), 304–338.

Zilberstein, S. (1996). Using anytime algorithms in intelligent systems. *AI Magazine*, *17*(3), 73–83.