UNIVERSITY OF SOUTHAMPTON
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

Coordination in multi-agent systems

A progress report submitted for continuation towards a PhD

Supervisor: Professor Nicholas R. Jennings
Examiner: Professor Paul Lewis

by Mair Allen-Williams

August 6, 2006

Contents

Chapter 1

Introduction

> *"Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed."*
>
> —[Maes, 1995]

Multi-agent systems are systems of interacting intelligent actors, or *agents*, existing in some environment. This environment provides stimulation to the agents' senses, and reacts to the agents' actions. There is no global view, rather, each individual is able to sense part of the system. Such systems are becoming increasingly important as they draw together a number of important trends in modern technology [Wooldridge, 2002]:

Ubiquity: As computing chips become smaller and cheaper it is possible to add computational power and intelligence to many kinds of devices in almost any location. Multi-agent systems made of networks of these ubiquitous devices have much greater possibilities than individual devices, as well as the potential for mobility.

Decentralisation: With the advent of the world wide web and computing networks, systems that distribute data and tasks among a network of machines are increasingly common.

Openness and dynamism: Open systems are those in which agents may enter or leave the system at any time, while dynamic systems have a constantly changing environment. Many real-world systems are modelled using open, dynamic environments. In particular, there is a trend in computing to move away from entering a static problem and waiting for a solution, towards interactive systems which are able to respond to a changing environment.

Uncertainty: Uncertainty plays a large part in systems which respond to environmental or sensor inputs. Moreover, a trend towards increasingly large and complex systems means that frequently systems are effectively uncertain, even if they are technically deterministic.

The combination of these features describes the kinds of decentralised data and information systems which are increasingly required by many commercial and industrial organisations. Multi-agent systems can be used to implement or to model all or part of these systems. Example application areas are as diverse as modelling eBay auctions [Rogers et al., 2006], modelling social structures [Sun and Naveh, 2004], or creating fight scenes in films (agent systems were used in *The Lord of the Rings*[1]). Consequently, multi-agent research is a lively and growing area facing many challenges.

In particular, the creation of large-scale systems with predictable behaviour is a formidable task. Current research (such as [Xu et al., 2005], [Shen et al., 2004]) focuses on the use of local models to generate desirable global properties. The idea behind this technique is to enforce the desired properties within small sub-networks of a large networked system. If this is done correctly, then the links between the sub-networks propagate the properties through the network. However, systems made up of many small components are liable to experience unexpected emergent behaviour in the large scale [Scerri et al., 2005]. This is a potential disadvantage of relying entirely on local, small-scale models. To combat this, researchers have looked at a variety of mechanisms for sharing information between disparate agents ([Mataric, 1998]). This means that models of flexible communication for open and dynamic systems form another area of active research. One final important area of research is that of dynamically evolving or learning solutions to problems, either to handle complex problems in large state spaces which cannot be programmed manually, or to keep up with systems which are changing over time.

Against this background, we seek to bring these distinct areas together in order to tackle the problem of coordination in complex, dunamic multi-agent systems. Thus, extant models together with our solutions, will bring together work on

[1] http://www.massivesoftware.com/what_massive.html

the use of local models, flexible communication, and online learning. To provide a specific grounding for this research, we will consider the domain of disaster response. As we outline in the next section, this has all of the aforementioned characteristics.

1.1 Coordination in the Disaster Response Domain

Consider disaster scenarios such as terrorist attacks, floods or earthquakes. In such scenarios, many different teams from a number of organisations must cooperate to attempt to recover the situation. Their work may be interrupted by self-interested actors such as journalists or (in the case of terrorist attacks) terrorist organisations. Some of the cooperating organisations may have conflicting goals. For example, suppose during an aeroplane crash an injured person is trapped in the wreckage very close to the "black box". The police will wish to keep the black box intact for the purposes of determining what caused the crash, while ambulance teams are concerned only with removing the injured person, perhaps necessitating the destruction of the black box unless they are very careful.

Scenarios of this nature provide rich grounds for the implementation of agent systems. In such applications, there is a scale determining the extent to which the system itself is autonomous, and the extent to which the agents rely upon human input or instruction. At one end of the scale, we may use multi-agent systems to model every aspect of the disaster response, simulating the disaster, the affected humans, and the response agents. Robocup Rescue[2] (described in detail in section 3) is an example of such a system. At the other end of the scale, agent systems can be used alongside the human response teams, processing data and interactively suggesting courses of action [Dorais et al., 1998]. In the middle of the scale can be found human-robot teams [Schurr et al., 2005] or agents who defer to humans in scenarios they are uncertain about [Scerri et al., 2004b]. The focus in this work is the broadest possible: the use of multi-agent systems for modelling the complete disaster response. The results of such models could be practically used in human controlled interactive systems, in which the automatic system's function is to propose courses of action which may be explored by the human user.

[2] `http://www.rescuesystem.org/robocuprescue/`

Taking disaster response as our illustrative domain for exploring multi-agent systems motivates a number of important requirements for the domain. We propose these as requirements which any coordination algorithm of interest ought to be able to handle:

Large: Disaster recovery scenarios may involve hundreds or thousands of distinct actors, organisations or teams, operating over a wide area.

Dynamic: It is unreasonable to assume that a realistic system will be static. Environmental conditions are subject to constant change and agents must be able to adapt to these changes. In disaster recovery scenarios agents must react to changing weather, unexpected events such as building collapse or fires and constantly moving traffic, among many other changing conditions.

Open: Systems of this nature will have agents moving in and out of the system constantly. In the worst case, in disaster scenarios agents are liable to die, hence vanishing suddenly. On the other hand, as volunteers and taskforces from elsewhere rush to contribute help, new agents will enter the response system.

Decentralized: [Panait and Luke, 2005] argue that providing a central server is equivalent to reducing the system to a single-agent system. Furthermore, in large and dynamic systems of the kind we are investigating, providing a central controller is likely to be infeasible: there are unlikely to be the resources to allow communications between one central controller and every other node, one central controller is almost certainly not going to be able to obtain a complete view of the system, and the potentially rapid changes as agents enter and leave the system would be difficult to track.

Uncertain: Large, dynamic, open systems typically have inherent uncertainty. Even if the system is technically deterministic, the complexity in the system is likely to make it effectively uncertain. For example, in disaster recovery scenarios taking place over broad areas, it is unlikely that any agent will have a complete view of the situation. Moreover, information which reaches the agent may be error-prone, increasing the uncertainty. At a different level of granularity, environmental conditions such as the expected weather or the height of a tide can be equally uncertain.

Heterogeneous: There are many different types of agents involved in a disaster response scenario, with a variety of capabilities and (potentially conflicting) goals. At a minimum there will be the rescue teams, each with distinct tasks: ambulances, police, helicopter teams, and there will be the people affected by the disaster. Involved may also be journalists, crime teams, environmental agencies, to name but a few.

Bandwidth-limited: One of the characteristics seen time and time again in disaster scenarios is limited communication [Committee on using IT to Enhance Disaster Management, 2005]. For example, mobile phone networks are jammed and only a fraction of the messages initiated are able to reach their destination.

Competitive: In domains such as disaster response, agents may have conflicting goals, as in the example above. Furthermore, self-interested agents have no reason to attempt to resolve the conflicts cooperatively.

There are many challenges when working in such domains, focusing on disaster response scenarios. The essential task of an agent in a multi-agent system is to process the inputs it receives, and plan to how to act. Actions will take place in the context of other agents, and so planning must take the other agents into account. Therefore, the two central challenges for such agents are information processing and coordination (we consider coordinated planning to be a coordination task).

Information processing is a vital task as scattered, incomplete, potentially error-prone, conflicting, time-delayed messages reach nodes in the system from many heterogeneous sources. We briefly discuss information fusion techniques in section 2.1; however, they are focused on a single agent rather than being specific to multi-agent systems. By contrast, acting in the context of other agents (*coordination*) is central to the notion of a multi-agent system, and it is this which forms the focus of the report.

1.2 Multi-agent Coordination

Coordination is central to multi-agent systems. In the broadest sense, "coordination" refers to an agent being aware of other agents in its environment [Durfee, 2001]. This may be in the context of resource allocation and consumption, task allocation, communication, or movement. Most approaches to coordination fall into one of three broad categories, or are built from a combination of approaches from these categories [Boutilier, 1996]:

1. **Conventions and roles** are the simplest way of coordinating between agents, and the least flexible. A convention is a commonly-known rule to which agents adhere. There are many real-word precedents for coordination by convention. For example, traffic control is frequently based around conventions such as stopping at red lights, or travelling faster in the right-hand lane of a motorway than the left-hand lane. Such coordination has the advantages of being simple and requiring no setup time [Fitoussi and Tennenholtz, 2000]. However, it is inflexible, and relies on all participants knowing the conventions and cooperating with them.

   An extension to the notion of convention for coordination is the use of roles within organisational structures. An agent's current role determines which conventions are appropriate, and which protocols are available to it. Hence, organisational structures based around agent roles have more flexibility than simple convention-based techniques, selecting between conventions based on the current role. Role-based structures have been successfully implemented for teams such as Robocup soccer teams [Tambe et al., 1999]. However, this work is not applicable to very open domains as it considers teams which are fixed over the relevant time period. It might be possible to implement more flexible role-based structures which could adapt to agents entering or leaving the system using appropriate conventions, although doing so without causing confusing complexity would require care.

2. **Communication** is another common human coordination technique. Coordination through communication has a small setup time and some bandwidth costs. It requires a common language, and the flexibility of this language determines the flexibility of the resulting coordination. There is potential for probabilistic models of language [Fischer et al., 2005], permitting (for example) adaptation to changing environments. Alongside a language for coordination, agents must have some means of reasoning internally about the outcomes. The nature of the coordination will thus depend considerably on the agents' internal coordination models. In any large system, such as our focus domain, there must be some form of communication in order to share information between agents; it will be impossible for any one agent to sense all the information it needs to function effectively in context [Dutta et al., 2004]. We expect to make limited use of communication beyond this information-sharing, as the bandwidth restrictions will preclude it in most cases.

3. **Learning** is both the most complex and the most flexible means of coordination. It may be combined with conventions or communication: conventions may be learned, for example, through communication [Kazakov and Bartlett, 2004]. There are many different ways of applying learning to coordination, from learning to choose between coordination protocols [Excelente-Toledo and Jennings, 2005], through learning to use simple communication techniques for coordination as in [Kazakov and Bartlett, 2004], to evolving social rules from scratch [Boutilier, 1996]. Learning techniques provide the potential for evolving detailed policies within large and complex state spaces, and for adapting to dynamic systems. However, they have a high setup cost, as learning good policies may be time-consuming and is potentially computationally intensive.

In a complex scenario, such as disaster response, an agent will typically be involved in multiple coordination activities at any one time. There may be multiple levels of coordination activity with a particular agent, such as a fire agent coordinating on a choice of fire site, and then a particular building on a site, and then the particular area of the building to target. There may be multiple types of coordination activity with a particular agent, for example an agent may be concurrently negotiating over a resource with one group of agents, while coordinating its position with another group of agents. Finally, coordination activities may take place with many agents in the system simultaneously. Moreover, there may be several dependencies between these coordinations. For example, a fire agent F may need to refuel, but it will not care whether it refuels from location A or location B. It negotiates simultaneously with agents at A and B in order to decide on a refuelling location. At the same time, other fire agents will be negotiating refuelling locations and so F should try and be aware of potential traffic jams if all agents head for the same fuelling point. Furthermore, F may be trying to minimise bandwidth during the negotiation in order to leave sufficient bandwidth to receive instructions from the central fire station.

In section 2, we discuss approaches to coordination in the light of our example domain and these interaction issues. The extent to which current coordination models take the possibility of interactions into account varies. We show that there is very little work on explicitly relating coordination interactions and using this information to improve all parts of the coordination. We suggest some ways in which future work could build on current models to use such explicit relationships when coordinating.

1.3 Report Structure

The aim of this report is to shed some light on the issues associated with coordination in challenging domains as defined above, and to introduce the use of the a disaster response testbed, Robocup Rescue, for investigating such issues. The rest of this report is structured as follows:

- In section 2, we introduce the kinds of agents with which we will work and discuss the state of the art with regard to coordinating such agents, focussing on the three coordination models described above.
- In section 3, we describe the Robocup Rescue testbed and its application to the coordination problems we have introduced. Some initial work within the Robocup testbed is explained and preliminary results are presented.
- In section 4, we conclude the report, proposing a number of directions for future study and a timeline for immediate work.

Chapter 2

Literature Review

This section begins with an overview of the kinds of intelligent agents which will carry out coordination in domains such as our example scenarios (section 2.1). We then introduce some of the issues relevant to coordination. The main body of the chapter describes in more detail the three primary coordination techniques introduced in chapter 1: conventions (section 2.2.1), communication (section 2.2.2) and learning (section 2.2.3). Some of the issues related to learning in large domains are then discussed (section 2.2.3). We conclude in section 2.3 with a summary of the coordination mechanisms and the ways in which they will be used in our example domains.

2.1 Autonomous Agents

A single agent functions by in some way responding to its environment. The kinds of agent which interest us have a notion of a goal or goals, and the ability to logically reason about their actions. Inconsistencies and conflicts in agent goals and beliefs may crop up and the agent's reasoning mechanism must be capable of handling these in some way. We believe that probabilistic methods are the most realistic of several possible techniques for doing this for a number of reasons. First, such techniques are effective for simulating the way humans may reason. Second, probabilistic representations are more straightforward than logical for both the input data and the agent models.

An agent will also need to have some way of modelling its environment, including any surrounding agent. It may do this explicitly, as in [Rovatsos, 2005] reasoning about the environment and agents, or it may only maintain a mapping of state signals to behaviours, leaving the models implicit. Explicit models have more potential for reasoning about states and behaviours as they store more information explicitly. However, maintaining explicit models may be computationally and memory intensive [Excelente-Toledo and Jennings, 2005]. We expect to work with both kinds of agents, modelling states explicitly where practical, and maintaining implicit mappings in computationally limited situations. In practice, if the world is large and detailed, agents will only be able to model small parts of it accurately. Determining which parts are of particular interest to any agent algorithm, including our coordination work, will form a part of that algorithm.

In a multi-agent system, agents will interact with each other, as well as their environment. Wooldridge's model [Wooldridge, 2002] of this interaction is to define each agent as having some sphere of influence within its environment. Overlapping spheres of influence indicate some form of interaction between agents. The agent's model of these influences contributes to its coordination decisions. In the next section, we discuss coordination in more detail.

2.2 Coordination in Multi-agent Systems

We take a broad view of coordination; agents acting in the context of other agents [Durfee, 1999]. As described in chapter 1, at any one time, not only may agents be performing a number of different coordination actions with different agents or groups of agents, but they may be performing a number of different types of coordination actions. The different forms of coordination may be controlled as part of the same algorithm, or they may use completely separate structures. For example, agents may use a known set of traffic conventions to manage their movements, while negotiating for one set of resources, and allocating another resource according to some organisational structure.

Regardless of the specific coordination method or level of abstraction, there are are several issues pertaining to the process of coordination. One key issue is the extent to which agents aim to perform cooperatively. [Shen et al., 2004] demonstrated that agents whose aims consider only global welfare perform less well than selfish agents, if their beliefs about other agents are false. Models should therefore include not only the goals and capabilities of the other agents, but an evaluation of the certainties in their model. In systems where agents communicate with each other about environmental conditions or the capabilities of other agents, it is necessary to judge the accuracy of the communication—the system may be error-prone, or agents may have self-interested reasons for offering false information [Kar et al., 2002]. An alternative view of this interaction between an agent's own goals and cooperative or group goals is to describe agent schemas for roles, norms and sanctions. These specify social rules which control the extent to which an agent behaves cooperatively [Weiss et al., 2005], building partial cooperation into the agent. Such schemas are more complex than a Bayesian model of certainty. However, they could provide an elegant way of defining boundaries on cooperation in a medium-sized or large agent society.

Another important issue is that in order to coordinate practically, agents may need to coordinate at a number of levels of abstraction [Durfee, 1999]. For example, a team of agents attempting to extinguish a fire must first form into a number of clusters who will each tackle one part of the fire. Within these clusters, more detailed coordination will take place determining the exact positioning of each agent.

Alongside this hierarchical decomposition of coordination, there is a lateral interaction between coordination actions [Wagner et al., 2000]. Suppose, for example, that an agent, Bertie, is attempting to find a route from A to Z, where the route involves crossing a river. Bertie may travel via the tunnel, which has a toll whose price may be negotiated. Alternatively, he may take the bridge. However, there is a weight limit on the bridge, and a lorry is parked on it. Bertie will initiate coordination protocols with both the toll booth and the lorry. Clearly these are not independent; when one negotiation comes to a successful conclusion, the other may be dropped. We can consider there to be coordination between the toll booth and the lorry. Both hierarchical and lateral interactions are very much in evidence in the kinds of large-scale heterogeneous scenario under consideration, and we will investigate techniques for handling both, based on the structure definid in [Wagner et al., 2000].

Finally, we must consider the way in which a coordination algorithm may be evaluated. Coordination may be evaluated in a number of ways, depending partly upon the particular test scenario. Algorithms may be compared according to how quickly or how efficiently they enable agents to carry out a series of missions, what resources agents consume over a period of time, what communication costs the algorithms occur, or what computational costs they occur. The choice of evaluation technique will depend on the particular requirements which we have of an algorithm or scenario. For example, the computation requirements of a coordination technique become relevant if the method may be used in agents which have very limited resources, such as the sensors in sensor networks. In other situations, the computation resources may not be known in advance, necessitating the use of an *anytime* technique.

Given some evaluation criteria, algorithms must be evaluated not only in the light of a particular scenario or set of scenarios, but in the way they behave as the setting becomes increasingly challenging. When we consider the way in which an algorithm scales to more difficult cases, there are a number of dimensions which can be considered [Durfee, 2001]:

1. Number of agents
2. Heterogeneity of agents
3. Agent complexity
4. Extent of interaction between agents
5. Degree of dynamism
6. Distributivity

We can match these challenges to the domain challenges listed in section 1.1. In particular, we seek coordination algorithms that are able to scale up to large numbers of other agents, and that permit agents to coordinate in the presence of perhaps many heterogeneous agents. We also aim to meet the challenge of coordinating within a dynamic environment. Scaling up through increasing degrees of interaction or agent complexity are of less relevance to the disaster domain, where these characteristics are generally consistent. The degrees of agent interaction are determined by the locations of agents and the structure of the communication networks (web-based, radio, 'phone...), while agent complexity refers to the (possibly human) disaster teams and other involved agents. On the other hand, several of the algorithms discussed below rely on small-world networks for their scaling properties—that is, an agent will be related to only a few other agents. However, in the kinds of domain we consider, such properties cannot be assumed. This is especially the case if we wish to consider interactions between coordinations which may have completely different network structures. Finally, our challenge domains may be highly distributed, with information sources and agents of all types scattered throughout the scenario region.

One final issue when evaluating an algorithm for scalability is the notion of robustness. This refers not just to graceful degradation, but to algorithms which are robust against the possibility of emergent phenomena [Huhns et al., 2005]. This is particularly the case in competitive domains, where malicious agents can exploit any emergent phenomena [Scerri et al., 2005] or predictable behaviour. In a disaster rescue situation, we envisage the possibility of self-interested journalist agents who would be prepared to exploit rescue agents.

Bearing these key issues in mind, we consider in more detail the three approaches to coordination introduced in section 1.2 and their potential use within our example domain.

2.2.1 Conventions and Roles

Social conventions or protocols are a low-bandwidth, commonly understood form of coordinating. Human society relies on some adherence to conventions for effective functioning. For example, traffic in the UK keeps to the left-hand side of the road; traffic on motorways travels faster on the rightmost lanes. In agent systems, it is possible to use shared conventions for some kinds of coordination. Even in dynamic, open systems of the kinds we investigate, we may assume some shared conventions based on background knowledge: in a disaster scenario, for example, there will be traffic rules known to the agents of the kinds described above. Other conventions might include rescuing children first, or demarcation of boundaries with red-and-white tape.

However, both rational and irrational agents in some situations may find themselves in positions where failure to adhere to the conventions appears to be an appropriate course of action. For example, in a disaster situation in which all the agents are attempting to travel in the same direction—away from a flood, say—there is no need to keep to the left-hand side (or right-hand side) of the road. This may result in a less well coordinated system. Consequently, intelligent agents which choose to adhere to the conventions should be able to adapt to other agents ignoring them.

In realistic scenarios, simple protocols or signals are likely to be insufficient to obtain effective coordination: a simple set of rules is rarely enough to handle complex scenarios, nor can it deal with coordinating distant agents who may not be aware of each other. Furthermore, no one set of protocols will be able to cover all possible developments in an open and uncertain dynamic environment. Therefore, large and complex situations, agent coordination will usually take place within some organisational structure, or collection of organisational structures [Horling and Lesser, 2005]. Individual agent roles within this structure may be associated with some set of conventions. For example, the role of goalkeeper in a soccer game may have a set of conventions determining where the goalkeeper is able to travel to. Such conventions are clearly different for the different team members. Associating conventions with roles permits a greater complexity of interaction than a uniform set of conventions, and has been found to be sufficient for coordinating teams such as Robocup soccer [Tambe et al., 1999].

In an arbitrary setting, or in open systems such as our example domain, it is preferable not to specify a fixed organisational structure, but for the system to be able to dynamically self-organise, and re-organise as the situation changes. Agents may move about; the scenario may demand different expertises from the "leader", and in our open systems agents enter and leave the system and the structure must adapt to this. Adaptive agents may use reasoning about the system, knowledge of simple organisational conventions, or some form of online learning in order to achieve this organisation [Horling and Lesser, 2005].

Other models separate out the coordination aspects of a structure or hierarchy from the application specific aspects [Sims et al., 2004]. In so doing, agents may be supplied with distinct coordination roles and goals which may or may not be associated with their current task. This can provide a useful abstraction for implementing coordination models separately from agent tasks. It may be possible to apply such a model to open and dynamic domains where the specific agents or tasks are changing, but the overall coordination structure can stay the same. The separation of coordination and task-specific aspects of a structure can be further extended to designate some agents as coordination proxies, delegating all coordination management to these agents [Scerri et al., 2005]. Such a model involves the use of communication, for the proxies to instruct the agents under their control. In the disaster domain, one particular manifestation of this proxy system could use humans to coordinate local agent teams, adjusting the model in [Schurr et al., 2005].

Organisation and role-based models of coordination must be very flexible in order to be effective in highly dynamic, open domains. Although common social conventions may enable agents to adapt to changing situations, or agents entering a system to quickly find a place in an organisation, relying on social conventions is potentially subject to exploitation by malicious agents. For example, a woman stealing a baby from another at a disaster scene would be saved quickly, following the convention that children are saved first (and the “mother” must go with her baby).

One final problem with the use of such models in large, complex domains is that there is little scope for managing the interrelations between different kinds of coordination, each of which may correspond to a different organisational structure. For example, a coordinated search of a disaster area by any available rescue teams must be interlinked with a communication protocol. There may be at least two separate communication structures; one through a mobile ’phone network and one through a radio network. Interlinked with this search will be traffic conventions. At the same time, agents may be negotiating for resources such as vehicles, water pumps, or bulldozers. The results of these negotiations may affect their willingness to contribute to a search. Role-based models lack the expressive power to handle these relationships well.

However, there may be some place for the use of social rules within our domain. In particular, as modelling disaster response scenarios involves some modelling of human society, rules such as traffic rules may be applied. It may also be the case that some organisational structures can be used as part of a coordination algorithm. Nonetheless, neither conventions nor roles seem sufficient for the formation of a complete coordination solution in such a domain.

2.2.2 Communication

In its simplest form, communicating may be a form of signalling information, an agent effectively announcing “I am going to do X”, or “Y has occurred”. The primary coordination mechanism may still be based around social convention, or some learnt mechanism; the purpose of the announcement is to enable the listeners to update their world view. As I bicycle in the UK, when I indicate right on the main road, I communicate the message “I am likely to turn right”. The cars behind me respond to this message by slowing down.

This simple signalling can be extended to share all kinds of information about the system. Information may be shared about the capabilities of other agents [Dutta et al., 2004], beliefs about their intentions [Seo and Sycara, 2005], or about other parts of the environment [Becker et al., 2005]. When I pull up at the traffic lights, the driver beside me who has been listening to his car radio may lean over to tell me about a crash on a nearby road, encouraging me to change my route plan to avoid that road. He might also tell me that the driver of the lorry three places back appears to be somewhat drunk, so I should leave plenty of room when the lorry tries to pass. I must make my own judgments about how much to trust this information, but if it is accurate, both the items enable me to coordinate better with the rest of the road traffic.

Suppose, however, that when I pull up beside the driver at the lights, they are just about to change. Before driving away, he will have time to tell me just one of the two pieces of information above. He must choose which one has the greatest value to me. In agent systems, Bayesian techniques are often used to make decisions about when and what to communicate in such bandwidth-limited scenarios [Goldman and Zilberstein, 2003], [Shen et al., 2003]. The driver's decision may take into account whether he will have another opportunity to communicate with me in the future. If so, he may choose to save his energy at this point [Becker et al., 2005], or to mention the lorry driver now with the intention of describing the crash at the next lights.

One of the complications of using communication for coordination is that the bandwidth itself is likely to be a coordination subject. A potential solution is to use one technique to manage use of bandwidth, while communication actions coordinate some other parts of the system. One example of this would be the coordination of a Mother's Union jumble sale with a predefined telephone tree for communication between members.

Another complication, touched on above, is that of communications which may be false, either because self-interested agents have some motivation for transmitting false information (if Bertie the lorry driver tells all the other drivers that Road A is blocked, they will avoid Road A, leaving it clear for Bertie to drive down), or because the communication is lossy or subject to transmission errors. Probabilistic techniques and maintaining detailed models of the other agents can be used to manage these difficulties (e.g. [Patel et al., 2005]) We will use both of these methods when tracking communication.

Information sharing of the kind we have described, and in particular the sharing of non-local information, is likely to be very useful within the kinds of large and dynamic domains under our consideration, where it is impossible for an agent to determine the full picture independently. Indeed, it should be unnecessary for any agent to have the full picture, but some communication of important information will be necessary to enable agents to coordinate effectively [Scerri et al., 2004a]. This is powerfully demonstrated in the Robocup Rescue testbed discussed in section 3.

Beyond information sharing, communication may be used within an organisational structure to transmit explicit instructions to agents. If agents are reduced solely to responding to explicit instructions, then they are no longer autonomous agents. However, the agents with which we will be working are likely to be maintaining complex goal structures and may be performing several actions at a time (for example: moving, communicating, giving or accepting some resource) towards the accomplishment of their goals. Accepting instructions *from time to time* does not prevent them from being autonomous.

Contrasting with a hierarchical (uni-directional) approach, another traditional use of communication in coordination is coordination via negotiations [Ramchurn, 2004]. In an agent negotiation there is some shared language which agents can use to “discuss” a task or resource until they come to a shared conclusion. There are many examples of human coordination in this fashion. In a human negotiation, the full power of human language and world knowledge is available, and for example haggling over a price may involve many diverse references intended to persuade the opponent to give way. Agents will have a much more limited language and usually some straightforward reasoning rules to apply to the situation.

Agent negotiations may be useful in situations where there is no clearly applicable set of conventions and sufficient bandwidth to carry out the negotiation. They apply equally well to coordination with agents disposed to be cooperative and those disposed to be non-cooperative. Furthermore, they guarantee that should they reach a conclusion, it is satisfactory to all communicating parties—although not necessarily optimal. However, in large domains and when communicating with unknown opponents, there is a risk of reaching deadlock. Furthermore, negotiating may be both time and bandwidth consuming. In a problem such as disaster response, it is reasonable to use simple negotiations for some small-scale problems in local sub-scenarios. This may be integrated with a coordination model at a higher level of abstraction, which could determine the reasoning agents use in deciding what solutions are satisfactory.

Specifically, we propose to use communication primarily for information sharing at all levels of a multi-level coordination model. Bayesian techniques such as those mentioned above will be used to determine when and what to communicate, and individual agents will maintain probabilistic models of the world based on integrating received communications. We may also use negotiations or instructions for some parts of the coordination, where appropriate. These will be within coordination models which are being used on some subsystem involving small numbers of agents and perhaps having a short-term coordination requirement.

2.2.3 Learning

There are several reasons for incorporating learning into multi-agent systems. Generally speaking, in large and complex systems it is impossible to suggest policies for every eventuality. Instead, agents may be initialised with some simple policies and left to learn appropriate behaviour based on the situations which actually occur. This is particularly relevant in dynamic systems where agents need to be able to adapt to changes over time. Finally, coordination in large open systems relies on agents learning about the agents around them, who may have entered the system at any point. For example, in a disaster scenario, human agents may volunteer to help as the situation progresses, or computing power may be donated to build software agents.

There are, therefore, a number of ways we can use learning in a system which is ultimately trying to learn to coordinate effectively:

- Learning conventions (this may be from scratch, or it may be that conventions are gradually changing over the course of the play and we need to keep up)
- Learning to choose between conventions
- Learning about parts of the environment not visible to us.
- Learning about the capabilities of other agents.
- Learning about the intentions of other agents.
- Learning languages and ontologies for communication.
- Learning what information may be useful to communicate.
- Learning when to communicate.

Approaches applicable to learning when to communicate may also be applicable to learning about other coordination actions. For example, we can consider communication traffic in a limited bandwidth network in an equivalent fashion to motor traffic on a transport network. The value of communication actions is parallel to the value to an agent of being in a particular location (where there are either resources it needs, or agents it needs to cooperate with). This means that there may be scope for re-using learned models in varied situations.

Maintaining models of the other agents is important in the cases where we are learning from other agents (from their actions [Price and Boutilier, 2001] or their communications [Goldman and Rosenschein, 1995]), or about other agents [Chalkiadakis and Boutilier, 2004], [Dutta and Sen, 2003]. For example, when learning about conventions, we will give less weight to signals from agents whom we do not trust to conform to conventions [Boutilier, 1996]. Similarly, when learning about the environment, if we suspect an agent is motivated to be dishonest, we will treat its messages with caution.

In the context of the domains and challenges introduced in section 1, we consider the following properties particularly relevant to learning:

Openness: Learning may be used to discover the characteristics of new agents joining the domain.

Dynamism: Online learning potentially enables the agents to track changes to the environment over time and adapt to new scenarios.

Levels of abstraction: Agents can learn to determine which level of abstraction is appropriate in a situation and flexibly adjust between these levels as the situation changes.

Relations between coordination actions: Agents can learn about relationships between coordination actions in situations where they may not be easy to specify explicitly.

Of the many ways in which learning can be incorporated into coordinating multi-agent systems, we do not envisage agents learning to coordinate from scratch, which is a large and time-consuming problem. Rather, we believe that agents should use online learning techniques in order to track the dynamics of the domain and to model interactions between the coordination processes which are occurring. Using learning in this way is a means of allowing additional flexibility and dynamism on top of the existing coordination mechanisms. By contrast, learning to coordinate from scratch would be a daunting and time-consuming task, more appropriate as an offline function than in time-critical scenarios.

In large and complex domains, online learning faces the challenge of state spaces which are equally large and complex. In the following section we discuss ways in which we might manage such state spaces when learning in our example domain.

State spaces in complex domains

Machine learning is a way of mapping a state signal to some kind of "action" or decision. The state signal comes from the agent's environment in some way. Agents maintain a *policy*: a mapping of states to actions. A very simple policy can be stored as a table, but in realistic domains, more innovative ways of mapping states to actions must be considered, especially as the state signal itself may not be discrete.

There are a number of different methods of managing the state space. The simplest is to partition by discretizing state signals [Sutton and Barto, 1998]. The number of partitions for each signal is determined by practical computational limits, and by an estimate of the accuracy needed from the signal. It may well be that the partitions will not be of the same size. For example, an obstacle-avoiding robot with a sensor range of 1m could have the "distance from object" input broken into three sections: *far*, *near*, *touching*. The "near" signal might refer to 0–5cm, while the "far" signal refers to 5–95cm. A more sophisticated method of partitioning is to dynamically determine or adjust the partitions, based on the areas of the state space most frequently visited [Sutton and Barto, 1998].

An alternative to explicitly discretizing the state space is to use some form of function to map from states to actions. One such mapping is the neural network. Other functional approximations include tiling of the state space, or radial basis functions [Sutton and Barto, 1998]. A more sophisticated form of approximation is the use of Kohonen maps, which attempt to model the topology of the state space, using clusters based on this topology to generate the inputs to a neural network [Smith, 2002].

While such techniques may be effective, they can be computationally intensive, and the outputs of something like a neural network are often hard to understand. Relational reinforcement learning [van Otterlo and Kersting, 2004] is a principled approach which attempts to model the state space using a set of logical predicates and constants. It is then possible to abstract the policy by replacing state constants with appropriate variables. Approaches to this abstraction may be inductive or deductive.

In a similar vein is a hierarchical approach to reinforcement learning [Fischer et al., 2004] : a small set of high-level actions is mapped to a small set of high-level states. The high-level action can then be broken down into a specific policy. This framework has been successfully tested on a model of agent communication, but could be equally appropriate in the granular coordination problems we investigate. In addition, such a framework would permit the use of different coordination models at different levels of abstraction, such as we propose to investigate.

There are advantages to both the relational and the hierarchical models. It may be possible to combine them: high-level states in the hierarchical model would correspond to very abstract states in a relational representation. Concretising some parts of the abstraction then moves the state down the hierarchy. This has not yet been tried.

In our work, it is not immediately clear how large the effective state space will be. The learning models we are proposing may apply only to small parts of the environment. This would be true, for example, if we were learning about the abilities of new agents which join the domain, in which case the "state signal" is solely the visible actions of the agent of interest. While the rest of the domain may be affected by the actions of the agent, we assume that it will be too complex to use this information to infer the agent's behaviour. It may, therefore, be sufficient initially to use a simple partitioning of the state space when learning. In other cases, such as to track (and adjust) conventions, it may be necessary to consider more complex state signals. For such complex states, we propose to use the combined relational-hierarchical model.

2.3 Summary

There are a variety of methods which have been used to coordinate effectively in a diverse range of domains. Techniques based around common conventions are very simple and well-suited to problems where agents need to share a common understanding quickly. However, they are generally restricted to local coordination, tend to be inflexible and may not be robust to deliberately destructive agents. They are also often insufficient in complex environments. In environments where agents have limited knowledge, for example because the domain is too large to investigate it all, or in an open domain with complex heterogeneous agents where the other agents are unknown factors, some form of communication may be necessary to aid coordination. This may take the form of agents sharing information about the environment with one another, or it may be that agents negotiate with each other to come to a common agreement. Some kind of communication is likely to be absolutely necessary in large and dynamic domains. However, bandwidth limitations and potential trust issues with agents mean that it should be used with care. Finally, agents can adapt to dynamic situations or learn about the important parts of the environment using learning techniques in combination with some other coordination mechanism.

In the kinds of large and complex domain we are exploring, we believe that a combination of coordination techniques should be used, with different methods appropriate at different levels of abstraction. This may be because understood conventions break down at certain levels of granularity, or because the environmental conditions are different—for example, communication can often take place freely on a local scale, but not on a wider scale. Another example of combining techniques would be to combine learning with some understood conventions, allowing agents to adapt the conventions as the situation changes over time. In such a system, each agent will dynamically select both the level of abstraction at which it wishes to coordinate with another agent, and the technique it uses for this coordination.

One important issue in such complex domains is the way in which different coordination processes emanating from the same agent may interact with other. An explicit representation for these interactions will be useful for effective coordination and permit better integration of the different coordination techniques which may be used for the separate processes. This is an avenue worthy of further investigation. A possible starting point for such an investigation is Sycara's coordination mechanism based on token passing [Xu et al., 2005], as this has scope for defining relationships between tokens. A model can be defined based on these relationships. We propose to investigate this possibility, and other ways of explicitly representing relationships between interactions, further. Coordination models based on flexibly combining different coordination techniques will be also explored. In the next chapter, we describe the use of the Robocup Rescue system for investigating coordination.

Chapter 3

Coordinating in the Robocup Rescue Domain

This section describes the use of a testbed, Robocup Rescue, for exploring coordination algorithms in realistic situations where there is more than one level of granularity. The work done in this domain, to date, has focused on simple convention-based coordination. The purposes of this work were to:

- Familiarise ourselves with the Robocup testbed and its use for evaluating coordination algorithms.
- Allow us to observe simple coordination in action in a realistic scenario, thereby obtaining some intuitive insights into which techniques might be appropriate to such domains.
- Provide a baseline for working with more elaborate coordination techniques in the Robocup testbed.

In the following section we introduce the Robocup testbed and its relevance to our problem. We then describe the Robocup scenario in more detail (section 3.1) and discuss some of the approaches to solving the Robocup problem (section 3.2). A simple approach is then given (section 3.3) and the results discussed (section 3.3.1). We conclude in section 3.4 with some insights about the effectiveness of our simple approach.

In more detail, the Robocup Rescue simulation[1] models a medium-scale disaster response scenario. It is a non-homogeneous, decentralized, uncertain scenario

[1] `http://www.rescuesystem.org/robocuprescue/`

which relies on coordination between agent strategies if agents are to function well, and has limited communication. It is therefore an interesting testbed for exploration of multi-level, decentralized, bandwidth-limited coordination strategies, of the type discussed in this report.

We chose to use the Robocup Rescue platform for our work for several reasons:

- Robocup Rescue is used throughout the international research community as a platform for testing aspects of integrated information fusion and agent systems. This means that there is a body of existing work within the Robocup domain which we can draw on, and against which our work can be evaluated.
- The Robocup Rescue scenario is based on real-world scenarios, with detailed simulators modelling different parts of the system. This provides a more thorough and useful testbed for coordinating agents than, for example, a simple gridworld model such as that used by [Tan, 1998].
- The Robocup Rescue base is open-source and the base is extensible in many ways. For example, it is possible to add new simulators to model different kinds of disaster scenario. This gives us the flexibility to test scenarios not encompassed by the base system and to develop new scenarios following evaluation of the initial work.
- Robocup Rescue is particularly pertinent to exploring coordination at different levels of granularity, and coordination processes which interact with each other. The scenario it models is well suited to a combination of local and global coordination, and there are a number of separate coordination processes (traffic management, global map search, communication decisions) which should all be integrated.

In the next section we describe the Robocup Rescue scenario and the coordination challenges which are found in a Robocup Rescue simulation.

3.1 Scenario

A robocup rescue scenario is based around a map of a city (or of a virtual or imagined city). The basic unit of a map is a node. Roads are connected by nodes, and buildings can be found opening off nodes. There may also be rivers

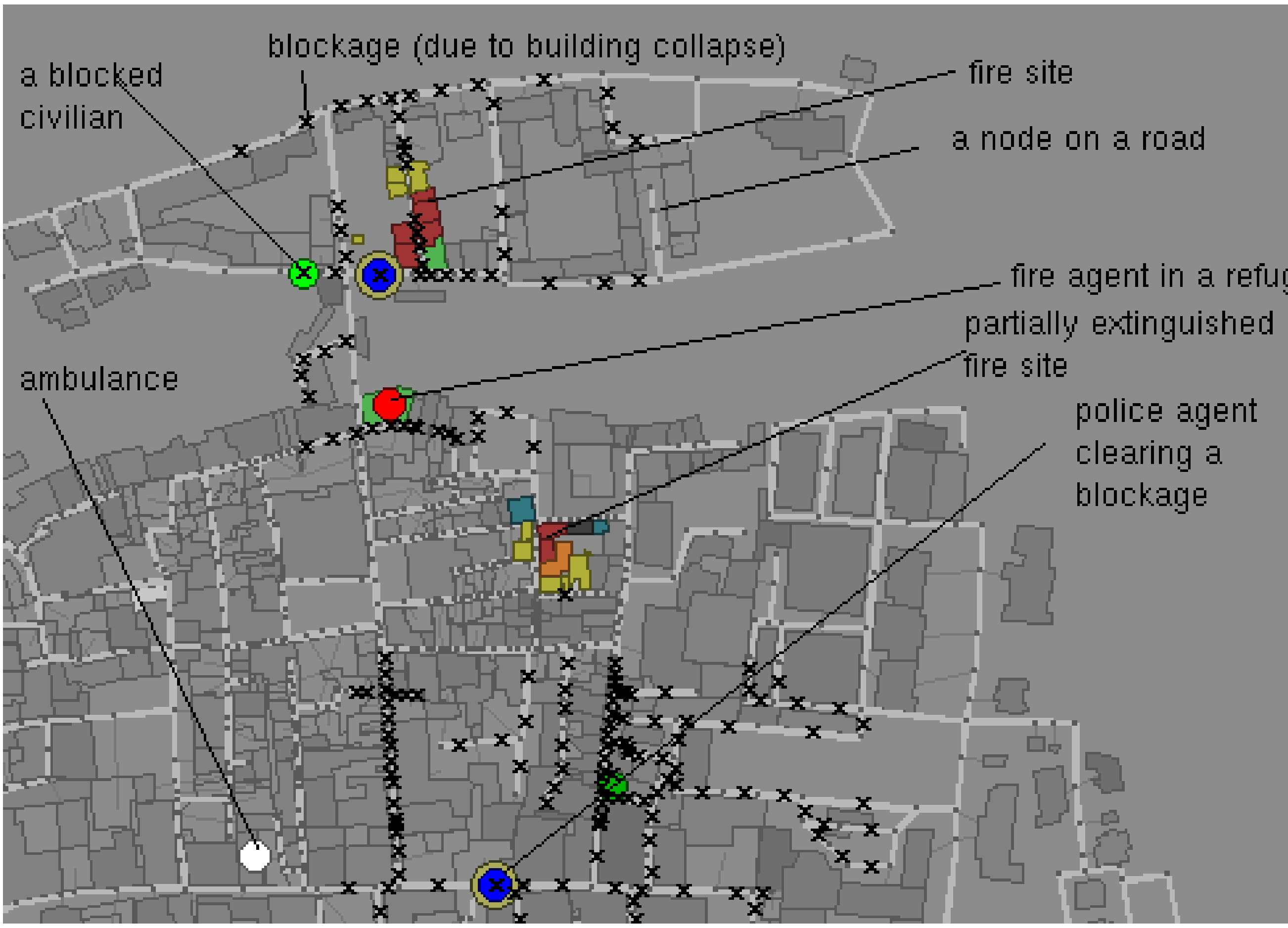


Figure 3.1: Robocup visualisation

Agent Type	Task	Target type
Ambulance	Rescue buried civilians	Civilians
Fire	Extinguish fires	Burning Buildings
Police	Remove blockages	Blockades

Table 3.1: Robocup Rescue agent tasks and abilities

on the map. Certain buildings are marked as *refuges*. Figure 3.1 shows the 2-d visualisation of a robocup scenario a short way into a simulation.

The scenario begins by assuming that there has been an earthquake in the city. At the beginning of a simulation, a number of the buildings may have collapsed, possibly with humans buried inside. Building collapse may cause road blockage. Finally, some of the buildings may have ignited.

All robocup rescue agents are able to see 10 metres around them, and the agents have x-ray vision; this distance is fixed even through building walls! In order to obtain a wider view of the map, they must communicate with one another. However, communication bandwidth is limited. During each minute, ground ("platoon") agents may receive no more than four messages, each of no more than 256 bytes. Their central offices (or "centers"), if they have such, may receive 2*n messages (of no more than 256 bytes), where n is the number of platoon agents. Agents may also communicate with other nearby agents—local communication is unlimited and has a range of 30 metres.

Robocup rescue agents have specific capabilities: ambulance teams are able to recover buried civilians, and transfer them to refuges (where they can be tended to); fire teams are able to extinguish fires, and police force teams are able to clear blocked roads. Table 3.1 summarises these capabilities. In a particular Robocup Rescue scenario, each type of rescue agent may have a team center. Centers have no action capabilities—their function is solely to pass messages to members of their platoon or other centers (message content is decided by the strategy). Consequently, they have considerably more communication bandwidth than the platoon agents.

The challenge for a Robocup Rescue team is to save the lives of as many humans as possible, and to minimise the area of the city which is burnt, during a simulation run of 300 virtual minutes. This is evaluated using a formula which takes into account the percentage of live citizens (including the rescue agents), the state of health of live citizens, and the average building damage (both fire and water). Scaling factors are used to adjust the relative importance of each of those.

To meet this challenge, each platoon must have a strategy related to its specific task (as defined in table 3.1), as well as strategies determining its part in the global tasks of searching and monitoring. These strategies should be coordinated with each other and among the agents. We describe these coordination challenges in more detail.

3.1.1 Coordination Challenges in Robocup Rescue

The first challenge facing a Robocup Rescue team is to provide a strategy for each platoon—police, fire and ambulance. Each platoon must have some means of prioritising its targets and a coordinated protocol for dispatching agents to targets. Platoons should be able to coordinate whether or not they have centers. Each platoon needs a slightly different coordination strategy. Only one police agent is able to work on a particular blockage at any point. By contrast, ambulance and fire agents may carry out a task faster if there are several agents at a site. Fire agents, however, must coordinate to distribute themselves around a site as well as to decide on a target site.

The second challenge is to coordinate between platoon types. Primarily, this involves a coordinated exploration of the map from the beginning of the simulation, and a common communication protocol for sharing discoveries. Agents must also cooperate to avoid traffic jams at hotspots on the map. Finally, agents may cooperate with each other in task-specific ways. For example, police agents who have no blocked targets may monitor civilian health to aid ambulance teams, while fire agents may concentrate their efforts on extinguishing buildings which are close to civilian targets.

3.2 Handling a Robocup Disaster Scenario

A complete Robocup Rescue strategy consists of: deciding on an organisational framework (within the existing structure), determining the communication protocols within this framework, creating a target prioritisation strategy for the platoon agents, and deciding how to coordinate agents as discussed above. These issues are interconnected: for example, the communication protocol will depend on the organisational structure and will restrict the possible agent strategies. Each of these points is discussed further below.

Organisational framework: In scenarios where a platoon has a center present, it is possible to use the center to collect information from all the platoon agents and then to determine the coordination within that platoon, sending out instructions to platoon members. This form of centralized coordination is most effective if there are centers for every platoon, as centers may only receive messages from platoon agents of their own kind. Any strategy which uses centralized coordination must also be capable of functioning efficiently in the cases where there are no centers. This could be by having distinct strategies for the different cases, and selecting one at run-time based on the scenario.

It is also possible to design a kind of centralized coordination by appointing platoon agents as leaders. This has the advantage of flexibility—the leader may change over the course of the run, and that more than one leader per platoon may be appointed if appropriate. For example, there may be a fire agent coordinating the group at each burning site. However, such coordination must be carefully negotiated. Bandwidth is very limited, and many simple coordination protcols rely on the assumption that an agent has an up-to-date world world view. Therefore, using bandwidth on coordination protocols in this way may not be effective (although there is, of course, room for experimentation).

Communication: Clear and well-coordinated communication is vital to the functioning of successful agents. It is tempting to clutter the communication protocol with “special” messages requesting a blockade clearance or a monitoring target. However, researchers have found it to be more effective to use communication purely for transmitting information about what has been sensed, leaving agents to decide their own targets [Habibi et al., 2006]. The same target-prioritisation algorithms may be used either way; transmitting information rather than requests makes it likely that the prioritisation algorithms will have more information to make use of (information fused from different sources), and may enable the agent carrying out tasks to balance them better as it can prioritise several targets together, making use of (for example) proximity information about targets of different types.

Prioritisation: Current approaches vary from hand-writing strategies [Skinner et al., 2004], to making use of sophisticated genetic sequencing techniques to determine targets [Kleiner et al., 2004]. Successful techniques use learning methods for making priority decisions [Eker and Akin, 2004]—the details of the interactions in the system are too complex for simple models to handle. We do not go into details of specific approaches for the agent types, as prioritisation techniques can be considered separately from the coordination techniques which interest us.

Coordination: As discussed, each platoon type will use a coordination protocol suited to its type and strategy, while using some global protocol for information sharing, contributing to the global search, and monitoring civilians where possible. The means of coordination must be entangled with the choice of communication protocol. In particular, if communication is intended solely to distribute information, then agents will not be able to negotiate with one another to coordinate, limiting coordination to being based upon shared conventions within a known organisational structure.

Evaluation: There is considerable interaction between agent strategies, so quantitatively evaluating one agent type alone is unrealistic. It is important in complex scenarios such as Robocup to evaluate strategies by observing a simulation and looking for ways in which agent behaviour appears to be strange or suboptimal, as well as by qualititatively scoring different strategies.

3.3 A Simple Strategy

Our main focus in this report is on the police agents, for whom coordination with other agents is inherent. The police agents should prioritise targets based entirely on their perception of the needs of the other agents, freeing stuck agents and ensuring there is access to refuges and fire sites before clearing the other routes on the map. Once the map has been searched and cleared, police agents can monitor other target types (civilians and fire sites), notifying the appropriate agents if there is a change in status which might require action.

Below, we describe our simple strategy with respect to the key issues identified in section 3.2. The strategy described here was used in the Robocup Rescue competition in Bremen in 2006, where it performed well but not brilliantly. We discuss the competition performance further in section 3.3.1.

Organisational framework: Initially, only scenarios where there was guaranteed to be a center agent for each platoon were considered. The extension to the full decentralized architecture is left for future work. We used the centers only for message passing, preferring to aim to give each agent as much information as possible with which to decide its own targets.

Communication: The communication protocol is a key part of agent strategies, and provides the backbone structure for agent coordination. We implemented a communication protocol which mostly transmitted information about what agents had sensed around them, but incorporated a small number of dedicated requests. In particular, agents which determine that they are stuck send a STUCK_REQUEST which is transmitted to the police agents (it need not be transmitted to agents of the other platoon types). It would be possible to eliminate this message if the stuck agents sent their location and a list of known blockages (pure information). A police agent could then run the *is_stuck* inference algorithm for all known agents on the map to determine which were stuck. However, this would be a large efficiency hit (each run of the *is_stuck* function requires a call to the route planner) in return for a small bandwidth saving.

While there is potential for applying compression algorithms to the communicated information, our initial work does not go this far. In order to function more effectively within the limited bandwidth, messages were prioritised, with the stuck requests receiving the highest priority (always sent); messages about sick civilians being prioritised above messages about fires (which can be seen from larger distance and hence will be reported by more agents), and messages about searched buildings given a low priority among fire and ambulance agents (since the building search is primarily carried out by the police agents—the reasons are explained in the "coordination" section).

Prioritisation: The ambulance teams prioritise targets by estimating how imminent death of the target is if it is not rescued. They use a scheduling algorithm to allocate agents to targets, possibly assigning more than one agent to each target to quicken the rescue. Each agent computes the full allocation of agents to targets and then moves to its own target—that is, coordination by convention. The convention is the commonly known scheduling algorithm which every agent uses.

The fire teams prioritise targets using a combination of features based around models of how fires spread and in what situations they can be successfully controlled. There is essentially no coordination among the fire teams. However, distance from a target is incorporated into the prioritisation, so that agents may distribute themselves among fire sites. Random movements during the search phase should cause fire agents to spread out even if they are initially at the same point on the map.

This provides the background for the police strategies. Their aim is to keep the roads clear for the ambulance and fire teams. Their targets are therefore prioritised according to their beliefs about the needs of the other teams. The highest priority is to free agents which have been completely blocked in. Other high priorities are clearing roads close to refuges (so that the ambulance teams can take civilians there) and clearing roads around fire sites, allowing the fire teams access. The priority ordering was decided empirically by observation of many Robocup Rescue simulations.

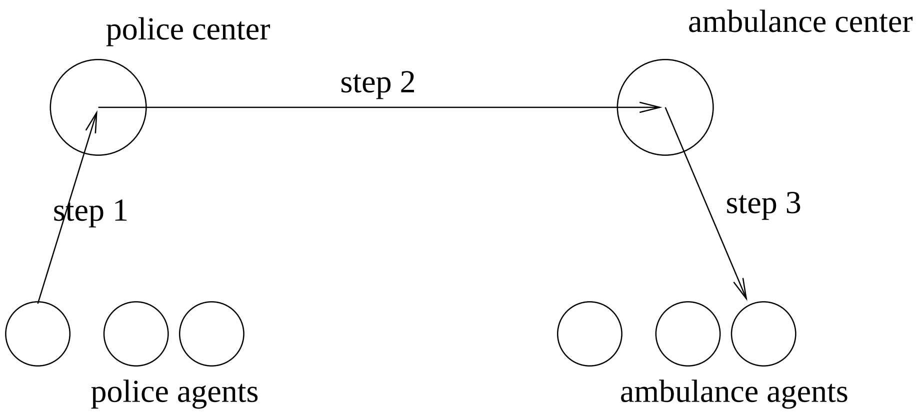


Figure 3.2: Passing a message between platoon agents in Robocup Rescue

By contrast with the ambulance teams, only one police agent may be clearing a blockage at any one time. It is therefore reasonable to supply only an ordering on target priorities without caring about relative importance. Agents are allocated to the highest priority targets first.

Coordination—global search: In the initial stages of the simulation, or at any point when they have no targets, all kinds of agents contribute to a coordinated search, travelling across the map and entering buildings to seek buried civilians. All agents communicate what they have searched so that they all share a world view.

All communication-based protocols for coordination will be lossy (since agents may have to ignore some of the messages they receive). Furthermore, there can be a delay of several cycles in transmitting information, since to get a message from one platoon agent to a platoon agent of a different type the message must go via the two centers, taking a minimum of three messages (see figure 3.2). Finally, although the agents aim to communicate all their knowledge to all other agents, inevitably there will be some differences between the agents' views at any one time. In particular, rescue agents are able to move very quickly across the map, meaning that their perceptions of each other's locations are liable to be out of date. We therefore introduced a coordination protocol for the search using a convention which agents would be able to compute independently. Agents will assume that they have similar world views to other agents, but that they do not know anything about the location of the other agents.

The search protocol is based around the allocation of agents to fixed sectors. The sectors are determined at agent initialisation using the k-means data clustering algorithm to create clusters of buildings. This algorithm is a simple way of clustering the region so that buildings that are within the same block are likely to be within the same cluster. It works as follows:

1. *(Initialisation)* Define k n-dimensional points as centres (in this case $n = 2$, as the points are x-y coordinates). We use the agent locations, which are known at the point of agent initialisation, as the initial centres.
2. Data points, here the building midpoints, are then allocated to the nearest centre, forming k clusters.
3. The centres are recomputed as the cluster centres.
4. Unless the clusters have stabilised (i.e. the centres have not changed), repeat from 2.

Figure 3.3 shows a viewer depicting the world view of one police agent (the blue dot towards the bottom left-hand corner). The buildings shaded yellow are those in the sector allocated to that agent. Those shaded white are the ones which the agent believes to have been searched at this stage in the simulation (a few cycles in). Two civilians have been discovered so far (the green dots at the top and towards the bottom on the right of the map). It is clear from the disparity of the searched locations that different agents have carried out the searches, communicating their discoveries to the agent whose world view is being shown.

One set of sectors is generated for each platoon, so each platoon could potentially search the whole map. Typically, however, after a short search, civilians will be found who must be rescued promptly if they are to be rescued at all. Similarly, fires should be extinguished promptly if they are to be controlled effectively. A good strategy will therefore take fire and ambulance agents out of the search fairly early on, as they go to deal with their own targets. This means that the majority of the search is likely to be carried out by the police.

Once an agent has been allocated a sector, it searches buildings which it believes to be unsearched, selecting targets based on to their proximity to the agent. There is nothing to prevent agents of different types carrying out overlapping searches. However, providing two agents of different types do not start from the same place, this should not occur.

This approach is somewhat ad-hoc approach at present. Essentially, it is a small collection of manually designed conventions which have been gathered together.

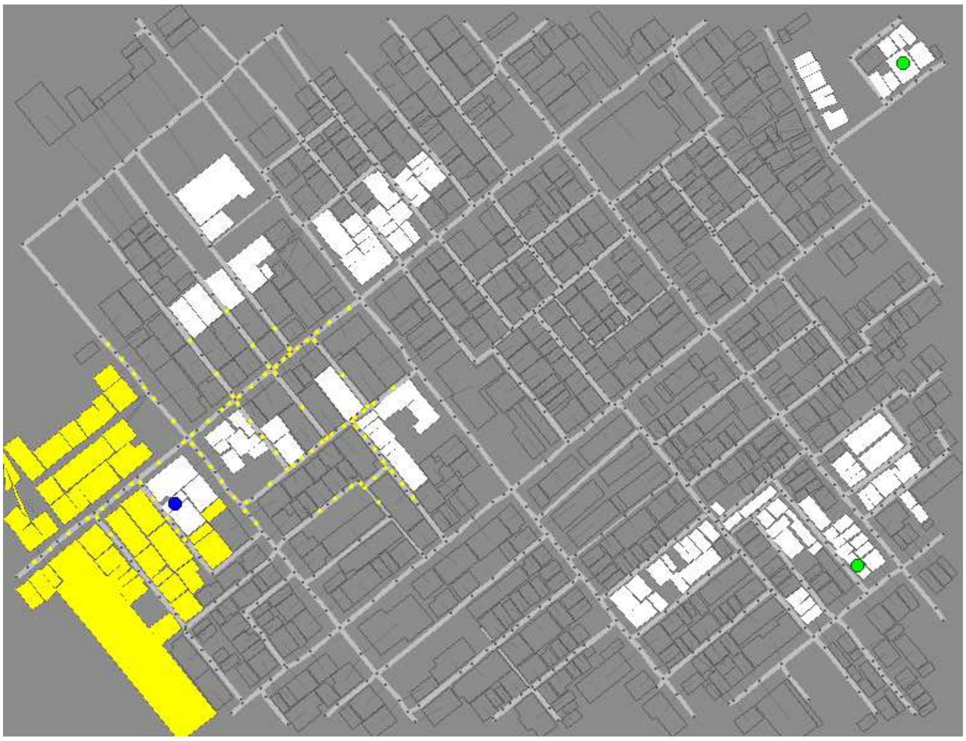


Figure 3.3: A police viewer, showing the search sector

However, it is both a simple and apparently effective approach which does not require any communication. One minor improvement might be to enable agents to detect when there are rescue agents close by (using the local communication protocols) and use some convention to separate the agents. This would still be fairly ad-hoc; an agent might end up bouncing around its entire sector running into other agents and moving away from them. Testing would determine whether this is a practical problem.

Coordination—police teams: As in the search, police are coordinated among targets using a sector-based convention. High-priority targets are considered important enough for agents to leave their sector. Targets are allocated in order of priority and each agent is allocated to the nearest unallocated target, where "near" is a measure of the distance between sector centres. This simplistic method is straightforward for each agent to compute without knowledge of the other agents' locations. It assumes that agents will be close to their own sectors—the initialisation of the k-means algorithm based on agent locations attempts to ensure this, although it may not always be possible (imagine, for example, the case where all the agents begin at the same point).

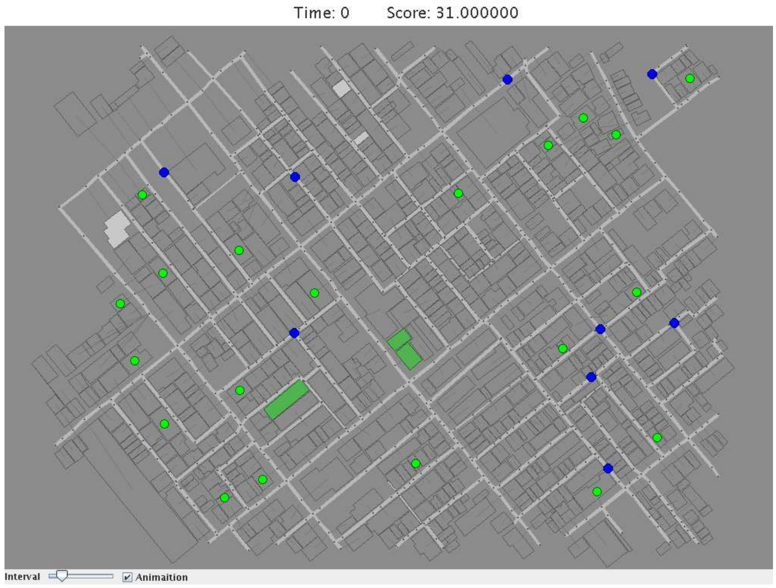
Several minor variations on the police strategy were tested. For example, the search strategy was crudely modified to try and ensure high-level coverage of the entire sector (that is, to have agents who had passed within sensing distance of each building) before the detailed building search. Another variation combined high-priority blockage targets into clusters, and assigned one agent per cluster. This reduced traffic jams in some cases, but sometimes resulted in high-priority targets not being cleared as soon as necessary.

Evaluation: Although our interest is primarily in the police agents, individual strategies can only realistically be evaluated in the context of the complete strategy. However, it is possible to get some insight into agent behaviour by testing some subparts of the strategy. As described, the police behaviour consists of three phases: searching, clearing blockages, and monitoring targets. Although in a real scenario agents will move back and forth between phases, it is possible to generate simplified scenarios which test some of these phases separately. A scenario with no buried civilians and blockades allows us to test the search phase exclusively. De-prioritising the search and initialising all agents with knowledge of the blockage locations provides a way of testing the clearance phase.

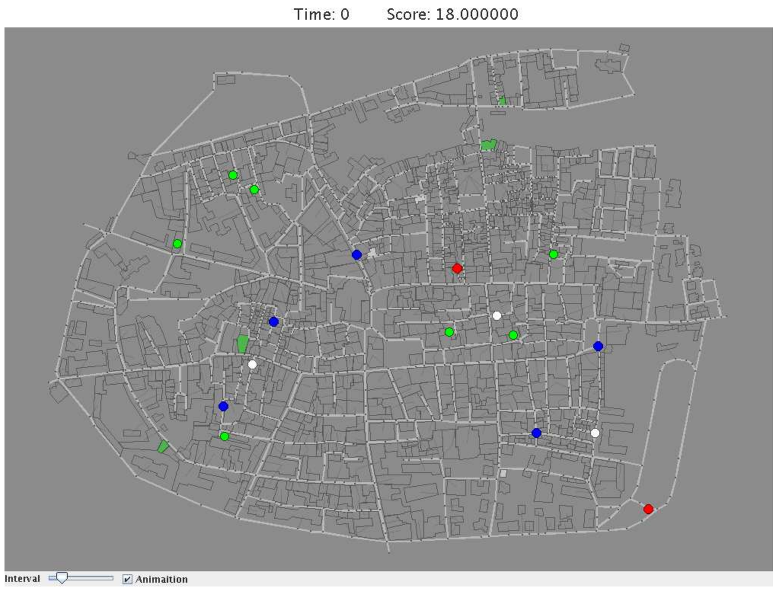
3.3.1 Results

The initial work done in this domain is somewhat limited, and we do not present a detailed set of comparative results here. Rather, we try to give a flavour of the way in which the algorithms behave for the police teams. We present some results for the speed at which the agents are able to search simple maps, discussing the results and the insights we can obtain from these results. We then discuss the behaviour of the full strategy in more general terms.

The four maps used are shown in figure 3.4. Table 3.2 shows the numbers of roads, nodes and buildings in these maps. The simplest (smallest) of the maps is Kobe (3.4(a))). Of medium complexity, but with quite different structures, are Foligno (an Italian town) and VC ("Virtual City") (3.4(b) and 3.4(c)). Foligno has narrow curved roads, with blocks of buildings tightly packed between them. Traffic jams occur easily on the many single-lane roads. Routing around the Foligno map with its many roads and nodes is less straightforward than it is in the structured VC. The most complex of the maps is the Random Large (3.4(d)). Another virtual



(a) Kobe

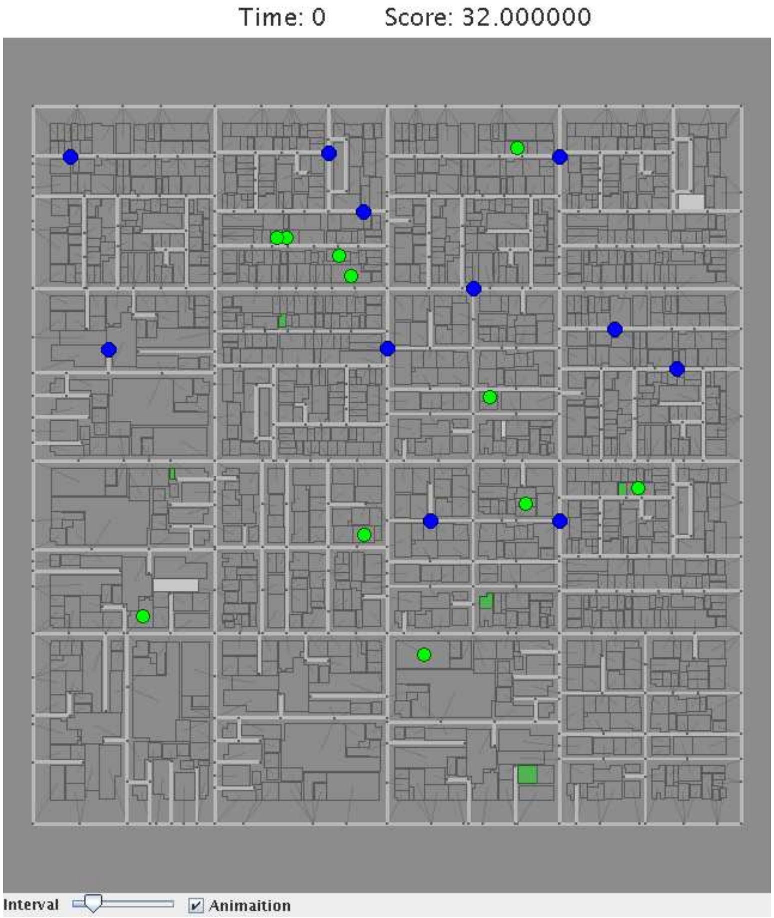


(b) Foligno

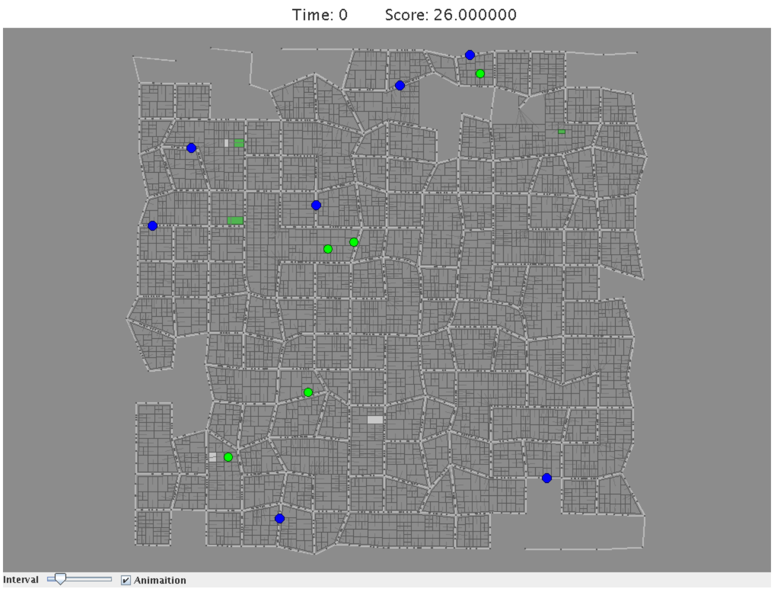
Figure 3.4: Robocup Rescue maps

city, less structured than VC, the main source of difficulty in this map is its sheer size.

Search: Table 3.3 shows the time taken for a team of police agents to search the buildings on a map on which there are a small number of civilians, no fires, no blockages, and no other rescue agents. This is not a realistic scenario, but gives us some insights into the behaviour of the coordinated search strategy. Two results are missing from the table: with only five agents the largest map, RandomLarge,



(c) VC



(d) RandomLarge

Figure 3.4, continued: Robocup Rescue maps

Property	Kobe	VC	Foligno	RandomLarge
Number of roads	820	621	1480	3002
Number of nodes	765	530	1369	2872
Number of buildings	734	1263	1078	2727

Table 3.2: Map properties for the four maps used

NumAgents	Kobe	VC	Foligno	RandomLarge
5	98	145	200	*80% completed*
10	55	80	92	212
15	41	53	66	148
20	38	46	63	-

Table 3.3: Time taken to search a blank map

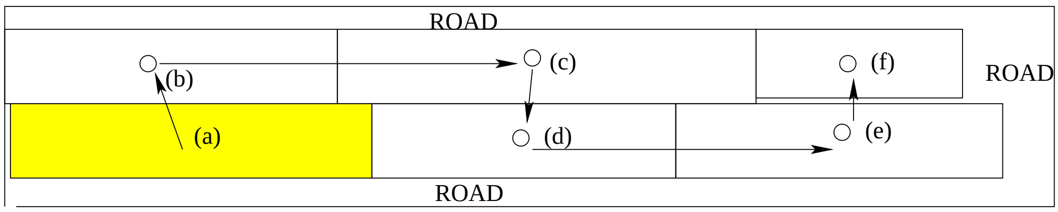


Figure 3.5: Poor target planning in a Robocup Rescue map. The agent will travel among the buildings in the order shown, causing it to move back and forth along the road several times

was only 80% searched within the 300 timesteps available; with twenty agents the machine did not have sufficient memory to run the simulation, resulting in the agents failing to move at all.

We examine the way in which the search strategy scales across larger maps, particularly as the number of buildings increases, and the way in which it improves as the number of agents is increased. Although the trend indicates that the search time is roughly proportional to the number of buildings, we can see from the results for VC and Foligno that this is not always the case. The additional complexity of the Foligno map results in it taking longer to search than VC, although there are fewer buildings to enter. Part of the reason for this is a deficiency in the search strategy: an agent targets the nearest building as the next building. In some cases this may be the building backing onto the current one, while the next door one is unsearched necessitating unnecessary travelling (see figure 3.5). This occurs more frequently in the Foligno map where there are densely packed small buildings.

As the number of agents increases, the rate of improvement in the search completion time decreases. This is partly because agents complete their own sector quickly, but are not then required to help out other agents, so that the total time corresponds to the time taken for the last agent to travel from its initialisation point on the map to its sector, and then complete the search. If agents were to move to incomplete sectors after finishing their own, this would only mitigate the scaling problem slightly because of the time taken to travel.

A second problem that occurs as the number of agents increases is that although the search may in fact be complete, many agents will believe it to be incomplete, because too many updates are being transmitted between agents for all them to be received. This highlights the importance of careful prioritisation of communication messages. During the search, for example, it is not actually necessary for agents to know which buildings have been searched in an area unless they are close to that area; they need only know pertinent information such as whether there are injured civilians in a building or blockages nearby. An improved search strategy might therefore prioritise these messages.

Complete strategy: Analysing a complete strategy is as much as matter of watching the agents' behaviour in a situation as creating a series of graphs. The Robocup Rescue competition provides a good opportunity for observing and comparing a number of agent strategies, as well as for testing our own simple strategy in challenging scenarios.

During the Robocup Rescue competition in Bremen this year, our agents performed respectably, demonstrating that they were capable of coming within the top eight agent teams of the twenty qualified entries. The police search strategy was competent, although there is room for tuning—civilians towards the edges of sectors were not always found on the large maps, for example. Two of the simulations were badly affected by failure of the police to clear important blockades, rendering some of the agents impotent. Police monitoring of civilians rarely had the opportunity to take place, and had little effect on the overall results. The ambulance team strategy performed reliably throughout; again, with some room for improvement. By contrast, the fire teams with their more complex strategy performed admirably in some scenarios and poorly in others.

Some of the lessons learned from this year's competition are:

- An otherwise effective strategy can be utterly ruined by failure to clear important blockages.

- A strategy for quickly homing in on civilians during the search is more important than searching and clearing blockages from the entire map.
- Rescue agents which are running on the same machine need to cooperate not just for resources within the Robocup scenario, but for computational resources. Anytime algorithms are particularly important when there may be many agents competing for CPU power.
- Although saving civilians is usually the most important way of gaining points, a strategy which allows the entire city to burn will drop the points to zero. Strategies should therefore try and integrate both tasks where possible.
- It is important to test for and be able to respond to pathological edge cases (that is, agents should be robust to difficult or unexpected scenarios)!

3.4 Summary

We have described an initial attempt at a complete strategy for Robocup Rescue, focussing on the techniques used for coordination. The current approach is often ad-hoc and based on intuition or observations combined with simple communication-free conventions. By improving the model for information sharing, and integrating it with other parts of the agent behaviour such as the global search, it will be possible to use some of the communication bandwidth for coordination messages, improving the overall quality of the strategy. As a result of this initial work we identify the following key challenges for coordination in the robocup rescue scenario:

- Flexible coordination with limited communication.
- Making use of local messages for local coordination.
- Taking a global view when coordinating.
- Testing coordination protocols on much larger robocup rescue scenarios.
- Explicitly identifying interactions between different coordination processes such as the global search and platoon coordination, and using this information for improving agent behaviour.

Most of these challenges are familiar to us as being related to those we have discussed in the introductory section and literature review, along with ways of approaching these problems: using flexible coordination models with online learning incorporated, initially perhaps using a token-based model to identify relationships between coordination interactions. In the next section we conclude the report with a number of specific directions for future work using these ideas.

Chapter 4

Conclusions and Future Work

In this report we have discussed issues relating to coordination in large multi-agent systems, focusing on the disaster response problem as an example domain. We have highlighted the problem of explicitly representing coordination interactions as one worthy of further investigation, with the intention of creating a holistic coordination strategy for a multi-agent system, such as could be deployed in a disaster response scenario. In section 3 we introduced Robocup Rescue as a testbed for coordination related work, and demonstrated a simple strategy within that testbed.

We now identify a number of directions for future work (see figure 4.1):

- Extending the Robocup kernel (**T3**):
  - Increasing the flexibility of the kernel to connect different types of agents
  - Adding the ability to have agents enter and leave the scenario throughout the duration of the run
  - Modelling lossy or error-prone communication

  These additions will enable us to test competitive, open scenarios, and to test scenarios with inexact information. These two types of framework are important cases within the proposed example domain.

- By running the agents on the Southampton Beowulf cluster, large problems (with perhaps hundreds of agents) can be tested in the Robocup domain (**T4**). This is a useful way of testing coordination algorithms in large and challenging domains.

- We will explore methods of combining coordination mechanisms (**T1**). From the work done to date on Robocup, we have seen that a good strategy should use different mechanisms for local coordination among nearby agents, for coordination among platoons, and for the global coordination. We will experiment with different ways of combining coordination models and compare the results.

- Additionally, we will aim to develop a model for automatically adjusting the coordination models where appropriate (**T1**), as the system is altered or scaled up. The novelty of this proposal lies particularly in the scope of the domain: the combination of large, heterogeneous, and open domains will form a demanding test environment.

- Another important research direction we intend to focus upon is the interactions between different forms of coordination (**T2**). For example, in Robocup we might consider communication, traffic management, search, and agent-specific tasks all to be separate coordination tasks which interact with each other. The relationships between them should be explicitly represented and used to improve the overall coordination—this is the kind of problem motivated in the introduction. Development of a coordination model which takes into account these interactions will be a useful extension to the current state of the art. This could be based on a variation of the token-passing model in [Xu et al., 2005]

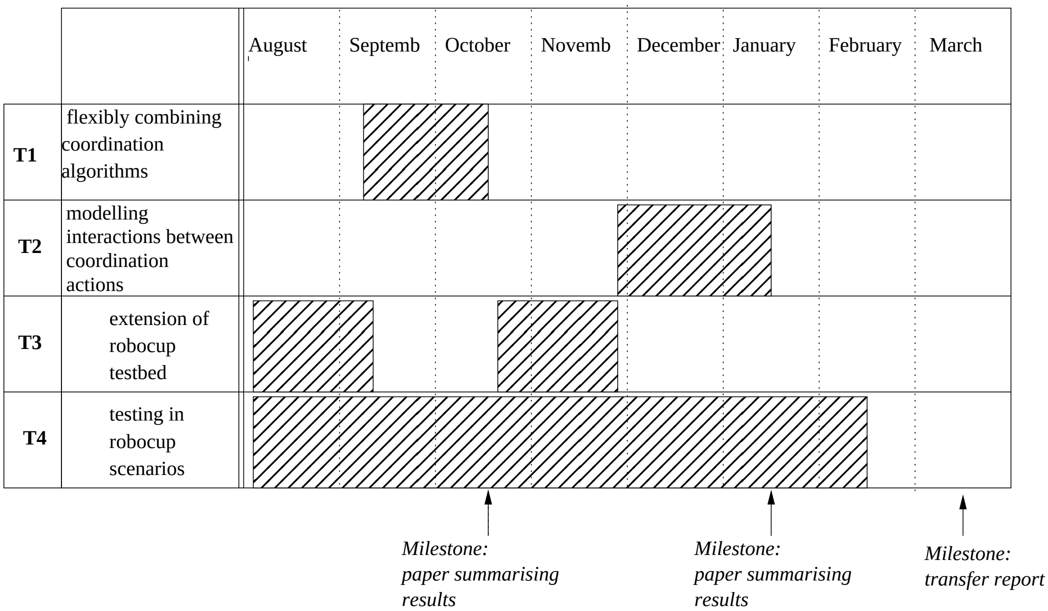


Figure 4.1: Timetable of work

To a large extent, the results of the work will drive later progress. However, we believe that firstly, future work should include testing the models in other challenging testbeds and other example domains with similar properties to the disaster response domain. This will provide more thorough testing for the algorithms and enable exploration of more of the scaling dimensions identified by [Durfee, 2001]. Secondly, we expect there to be potential for incorporating online learning into the last item above. Agents would be able to learn about the interactions between coordination actions, and the strengths of these interactions. They could then flexibly adjust their coordination appropriately.

Bibliography

Raphen Becker, Victor R. Lesser, and Shlomo Zilberstein. Analyzing myopic approaches for multi-agent communication. In *International Conference on Intelligent Agent Technology*, pages 550–557, 2005.

Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pages 195–210, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc. ISBN 1-55860-417-9.

Georgios Chalkiadakis and Craig Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1090–1097, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 1-58113-864-4.

National Research Council Committee on using IT to Enhance Disaster Management. Summary of a workshop on using information technology to enhance disaster management, 2005. URL `http://www.nap.edu/catalog/11458.html`.

G. Dorais, R. Bonasso, D. Kortenkamp, P. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems on Mars, 1998. Presented at the Mars Society Conference.

Edmund H. Durfee. Scaling up agent coordination strategies. *Computer*, 34(7): 39–46, 2001. ISSN 0018-9162.

Edmund H. Durfee. Practically coordinating. *AI Magazine*, 20(1):99–116, 1999. URL `http://jmvidal.cse.sc.edu/library/durfee99b.pdf`.

Partha S. Dutta, Srinandan Dasmahapatra, Steve R. Gunn, Nicholas Jennings, and Luc Moreau. Cooperative information sharing to improve distributed learning. In *Proceedings of The AAMAS 2004 workshop on Learning and Evolution in Agent-Based Systems*, pages 18–23, 2004.

Partha Sarathi Dutta and Sandip Sen. Forming stable partnerships. *Cognitive Systems Research*, 4(3):211–221, 2003.

Baris Eker and H Levent Akin. Roboakut 2004 rescue team description, 2004. URL http://www.science.uva.nl/~arnoud/research/roboresc/robocup2004/tdps-Rescue-Simulation-2004/.

Cora B Excelente-Toledo and Nicholas R Jennings. Using reinforcement learning to coordinate better. *Computational Intelligence*, 21(3):217–245, August 2005. URL http://www.ingentaconnect.com/content/bpl/coin/2005/00000021/00000003/art00001.

Felix Fischer, Michael Rovatsos, and Gerhard Weiss. Hierarchical reinforcement learning in communication-mediated multiagent coordination. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1334–1335, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 1-58113-864-4.

Felix Fischer, Michael Rovatsos, and Gerhard Weiss. Acquiring and adapting probabilistic models of agent conversation. In S. Koenig and M. Wooldridge, editors, *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 106–113. ACM Press, 2005.

David Fitoussi and Moshe Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119(1-2):61–101, 2000.

Claudia V Goldman and Jeffrey S Rosenschein. Mutually supervised learning in multiagent systems. In *Adaption and Learning in Multi-Agent Systems*, pages 85–96, 1995.

Claudia V. Goldman and Shlomo Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 137–144, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-683-8.

Jafar Habibi, Alireza Fathi, Saeed Hassanpour, Mohammadreza Ghodsi, Behzad Sadjadi, Hamidreza Vaezi, and Majid Valipour. Cooperation in a multi-agent environment, 2006. http://ce.sharif.edu/~impossibles/rescue_2006/impossibles_2006.pdf.

B. Horling and V. Lesser. a survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19:4:281–316, 2005.

Michael N. Huhns, Munindar P. Singh, Mark Burstein, Keith Decker, Ed Durfee, Tim Finin, Les Gasser, Hrishikesh Goradia, Nick Jennings, Kiran Lakkaraju, Hideyuki Nakashima, Van Parunak, Jeffrey S. Rosenschein, Alicia Ruvinsky, Gita Sukthankar, Samarth Swarup, Katia Sycara, Milind Tambe, Tom Wagner, and Laura Zavala. Research directions for service-oriented multiagent systems. *IEEE Internet Computing*, 9(6):65–70, 2005. ISSN 1089-7801.

Parijat Prosun Kar, Sandip Sen, and Partha Sarathi Dutta. Effect of individual opinions on group interactions. *Connection Science journal*, 14(4):335–344, 2002.

Dimitar Kazakov and Mark Bartlett. Cooperative navigation and the faculty of language. *Applied Artificial Intelligence*, 18(9-10):885–901, 2004.

Alexander Kleiner, Michael Brenner, Tobia Brauer, Christian Dornhege, Moritz Gobelbecker, Mathias Luber, Johann Prediger, and Jorg Stuckler. ResQ Freiburg: Team description and evaluation, 2004. URL `http://www.science.uva.nl/~arnoud/research/roboresc/robocup2004/tdps-Rescue-Simulation-2004/`.

Pattie Maes. Artificial life meets entertainment: Life like autonomous agents. *Communications of the ACM*, 38(11):108–114, 1995.

Maja J. Mataric. Using communication to reduce locality in distributed multiagent learning. *ournal of Experimental and Theoretical Artificial Intelligence*, 10(3):357–369, 1998.

Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005.

Jigar Patel, W. T. Luke Teacy, Nicholas R. Jennings, and Michael Luck. A probabilistic trust model for handling inaccurate reputation sources. In *iTrust*, pages 193–209, 2005.

Bob Price and Craig Boutilier. Imitation and reinforcement learning in agents with heterogeneous actions. In *AI '01: Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pages 111–120, London, UK, 2001. Springer-Verlag. ISBN 3-540-42144-0.

S. D. Ramchurn. Multi-agent negotiation using trust and persuasion, 2004. PhD thesis. ECS, University of Southampton.

A. Rogers, E. David, J. Schiff, and N. R. Jennings. The effects of proxy bidding and minimum bid increments within ebay auctions. *ACM Transactions on the Web.*, 2006.

Michael Rovatsos. Practical social reasoning systems, 2005. URL `http://homepages.inf.ed.ac.uk/mrovatso/papers/rovatsos-esb.pdf`.

P. Scerri, Y. Xu, E. Liao, J. Lai, and K. Sycara. Scaling teamwork to very large teams. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 888–895. IEEE Computer Society, 2004a.

Paul Scerri, Katia Sycara, and M. Tambe. Adjustable autonomy in the context of coordination. In *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, 2004b. Invited Paper.

Paul Scerri, Elizabeth Liao, Y. Xu, Michael Lewis, G. Lai, and Katia Sycara. Coordinating very large groups of wide area search munitions. *Theory and Algorithms for Cooperative Systems*, 2005.

Nathan Schurr, Janusz Marecki, John P. Lewis, Milind Tambe, and Paul Scerri. The defacto system: Coordinating human-agent teams for the future of disaster response. In *Multi-Agent Programming*, pages 197–215. Springer, 2005.

Young-Woo Seo and Katia Sycara. Exploiting multi-agent interactions for identifying the best-payoff information source. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-2005)*. IEEE Computer Society Press, September 2005.

Jiaying Shen, Victor Lesser, and Norman Carver. Minimizing communication cost in a distributed Bayesian network using a decentralized Markov Decision Process. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 678–685, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-683-8.

Jiaying Shen, Xiaoqin Zhang, and Victor Lesser. Degree of Local Cooperation and its Implication on Global Utility. In *Proceedings of Third International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2004)*, volume 2, pages 546–553, New York, New York, July 2004. IEEE Computer Society. URL `http://mas.cs.umass.edu/paper/359`.

Mark Sims, Daniel Corkill, and Victor Lesser. Separating Application-Specific and Organizational Coordination Issues during Multi-Agent Organizational Design and Instantiation. Computer Science Technical Report 04-98, University of Massachusetts, November 2004. URL `http://mas.cs.umass.edu/paper/382`.

Cameron Skinner, Jonathan Teutenberg, Gary Cleveland, Mike Barley, Hans Guesgen, Pat Riddle, and Ute Loerch. The Black Sheep team description, 2004. URL `http://www.science.uva.nl/~arnoud/research/roboresc/robocup2004/tdps-Rescue-Simulation-2004/`.

A. J. Smith. Dynamic generalisation of continuous action spaces in reinforcement learning: A neurally inspired approach, 2002. Ph.D. thesis, Division of Informatics, Edinburgh University, UK.

R. Sun and I. Naveh. Simulating organizational decision-making using a cognitively realistic agent model. *Journal of Artificial Societies and Social Simulation*, 7(3), 2004.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Milind Tambe, Jafar Adibi, Y. Alonaizon, Ali Erdem, Gal A. Kaminka, Stacy Marsella, and Ion Muslea. Building agent teams using an explicit teamwork model and learning. *Artificial Intelligence*, 110(2):215–239, 1999.

Ming Tan. Multi-agent reinforcement learning: independent vs. cooperative agents. In *Readings in agents*, pages 487–494. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. ISBN 1-55860-495-2.

M. van Otterlo and K. Kersting. Challenges for relational reinforcement learning. In P. Tadepalli, R. Givan, and K. Driessens, editors, *Proceedings of the Workshop on Relational Reinforcement Learning at ICML'04*, pages 74–80, December 2004.

Thomas Wagner, Brett Benyo, Victor Lesser, and Ping Xuan. Investigating interactions between agent conversations and agent control components. In Frank Dignum and Mark Greaves, editors, *Issues in Agent Communication*, pages 314–330. Springer-Verlag: Heidelberg, Germany, 2000.

Gerhard Weiss, Matthias Nickles, Michael Rovatsos, and Felix Fischer. Specifying the intertwining of cooperation and autonomy in agent-based systems. *Journal of Networks and Computer Applications*, 28, 2005.

Michael Wooldridge. *An Introduction to Multi-agent Systems*. Wiley, 2002.

Yang Xu, Paul Scerri, Bin Yu, Steven Okamoto, Michael Lewis, and Katia Sycara. An integrated token-based algorithm for scalable coordination. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 407–414, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-093-0.