Using Multiagent Coordination Techniques in the RoboCup Four-legged League

Vanessa Frias-Martinez
Department of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 11025
vf2001@columbia.edu

Marek Marcinkiewicz
Department of Computer Science
Graduate Center
City University of New York
New York, NY 10016
mom7@cs.columbia.edu

Simon Parsons
Department of Computer and Information Science
Brooklyn College
City University of New York
Brooklyn, NY 11210
parsons@sci.brooklyn.cuny.edu

Elizabeth Sklar
Department of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 11025
sklar@cs.columbia.edu

Abstract

We participate in the RoboCup four-legged league, and view this as a testbed for the use of multiagent coordination techniques. While the techniques we currently use are rather straightforward, our longterm goal is to adopt some of the more complex mechanisms we have been developing in other research. This paper describes these mechanisms, our work with RoboCup to date, and sketches the direction we see our research taking in the future.

Introduction

We are researchers in multi-agent systems who work in robotics. In particular, we participate in the RoboCup four-legged league, which aims to develop techniques for playing soccer using Sony AIBO robots. While our main goal in this work is to develop techniques that advance the science of robotics, and its application to the rather whimisical task of robots playing soccer, we view RoboCup as a testbed for many of the techniques that we have developed, and continue to develop, in the area of multi-agent coordination.

Some of the RoboCup leagues, notably the simulation leagues, have already seen considerable work on coordination (for example (Stone, Veloso, & Riley 1999; Tambe *et al.* 1999)). Other leagues, which have physical reality rather than a simulated world to contend with, have required more work on areas such as vision and localization (indeed in the four-legged league, even kicking the ball is not without its difficulties[1]). However, progress is being made. It is now usual for teams to coordinate their position on the pitch, creating the beginnings of positional play rather than the "little league" approach, with all the robots heading for the ball, seen in previous years. Some teams are managing to get robots to pass the ball to one another as the Georgia Tech Yellow Jackets[2] illustrated at the first RoboCup American Open, and the German team[3] showed at RoboCup 2003 in Padua[4]. We believe that such efforts will become widespread, and that the techniques for achieving this kind of play will benefit from the application of ideas developed in the multi-agent systems arena.

This paper explores the connections that we see between our work on multi-agent coordination and our work on soccer-playing robots. In the next section we describe the RoboCup challenges, the four-legged

[1] This point was amply illustrated by one of the four-legged league challenges this year. The challenge required teams to play with a black and white ball rather than the usual orange one. While intended as a vision problem, it turned out that because the black and white ball was rather elastic (much like a real soccer ball) rather than being hard like the usual orange ball, the regular repotoire of kicks were largely ineffective.

[2] http://www.cc.gatech.edu/~borg/robotsoccer/

[3] http://www.robocup.de/germanteam/

[4] The team from Georgia Tech used an innovative device whereby one robot rebounded the ball from the wall around the pitch into the goal area. The German team passed in a more conventional manner—one robot kicking the ball down the pitch towards a teammate.

league in general, and our team in particular, along with the initial coordination mechanisms we are using.

Multi-robot soccer

This section provides the background to our discussion of the use of multi-agent coordination techniques for multi-robot problems.

The RoboCup challenges

RoboCup[5] is an international scientific effort to advance the state of the art in robotics. It offers a series of challenges and yearly gatherings in which participants compete against each other in striving to meet the challenges. Through this competition, and the gradual increase in difficulty of the challenges over time, it is intended that the participants will steadily "raise the bar". Comparing the capabilities of the teams over the years, this increased performance is easily discernable, as is the popularity of the competitions (RoboCup 2003 attracted 1250 participants in 224 teams).

One of the challenges—the first to be offered, and the one that is most popular amongst university participants—is that of robot soccer. The ultimate aim of this challenge is to produce a team of humanoid soccer playing robots that can play with the human world champions by 2050. In the interim, the soccer challenge is divided into a number of leagues, each of which competes on a different kind of robot platform (and thus deals with different aspects of the problem of producing humanoid soccer players). The league that we participate in is the Four-legged League[6], in which all teams use the Sony AIBO robots. Such robots are pictured in Figure 1[7].

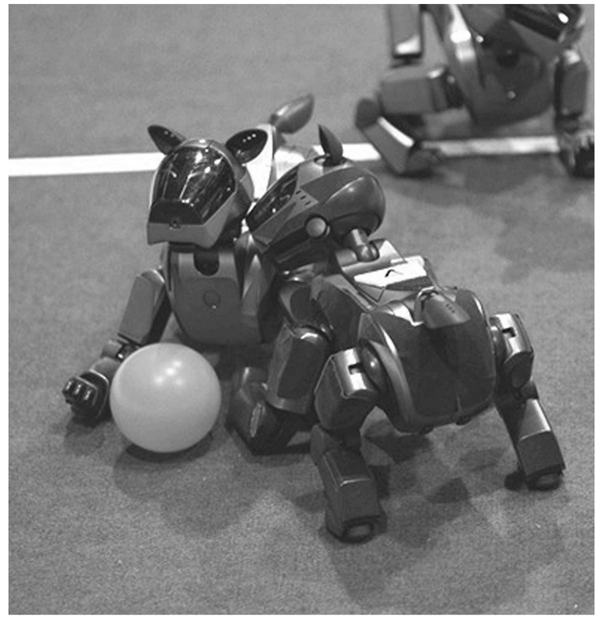

Figure 1: AIBO robots playing soccer

The big advantage to participants in this league is that the same hardware is easily available to everyone, so that teams can share code (with most teams releasing their code at the end of the yearly competition) and no team can gain an advantage through hardware innovation. However, there are disadvantages as well—the hardware is limited—both processor speed and resolution of the built-in camera are considerably behind the state of the art—and getting the robots to move around is somewhat harder than with wheeled platforms.

MetroBots and the four-legged league

In the four-legged league, teams comprise four robots, playing on a pitch that is approximately twelve feet long and nine feet wide. A large part of the problem of getting the robots to play soccer is classic robotics work. As mentioned above, getting legged robots to move is not trivial, and the fact that the robots only really have image data to work from (they have a laser range finder as well, but it is not useful enough to bother with) means that teams have to deal with the computer vision problem as well. In order to ameliorate the difficulty of the vision problem, many of the objects the robots have to see are conveniently colour-coded. The ball is orange[8], the goals are painted blue and yellow, and reference markers at the corners and middle of the pitch are also coloured (all markers include a pink area, and an area that is blue, yellow, or green). These colours are more or less standard across the RoboCup leagues.

The combination of the colours and the speed of the processor means that robots are able to identify objects and localize themselves—most teams using a version of Monte-Carlo Localization (Thrun *et al.* 2000)—sufficiently well to play without any obvious delay. As a result, it is perfectly possible for teams to play a creditable pastiche of little-league soccer. All the robots can see the ball (when it is not occluded) more or less anywhere on the pitch, head towards it, and kick it towards the opposing goal. This level of play can be relatively easily achieved, for example our MetroBots[9] team was capable of this style of play after 6 months of development when we competed at the first American Open.

The AIBO robots also allow one to go beyond this simple approach to soccer (as most teams, including the current version of MetroBots, do as well). The robots are equipped with wireless ethernet cards, and so are capable of communicating with one another. This capa-

[5] `http://www.robocup.org`

[6] `http://www.openr.org/robocup/`

[7] This image shows the ERS-210A AIBO which was the only robot used in the four-legged league in 2002 and 2003. Since Sony is phasing this out, in 2004 teams will start to move to the ERS-7.

[8] Though, as mentioned above, one of the league challenges at RoboCup 2003 was devoted to playing with a ball coloured in a more conventional black and white pattern, and many teams favour using this kind of ball for matches in the near future.

[9] `http://satchmo.cs.columbia.edu/metrobots/`

bility is used to start and stop games, and for robots to pass information to one another (for example about the position of the ball), and obviously makes it possible to have inter-robot coordination as well.

As things stand, not much use is made of this ability to communicate to coordinate. The standard approach is to have the robots communicate in order to allow them to dynamically adopt different roles. For example, we use a scheme whereby the robot that is nearest to the ball takes on the responsibility of moving the ball towards the opposing team's goal, while the others stand off so as not to interfere. The robots communicate to decide how to do this—each robot broadcasts the distance its estimate of the distance between it and the ball. Other teams use a similar mechanism to have a nominated defender that positions itself between the ball and its own goal to interfere in any attack on that goal. However, as things stand, this is about the limit of what is done.

This is understandable since the wireless communication, at least in competitions where interference comes from the many laptops and mobile telephones in the arena, can be unreliable and can lead to visible delays in the game[10]. In addition, coordination is possible without direct communication. Since all the robots know the state of the game, it is possible to implement "locker room agreements" (Stone & Veloso 1999), whereby the robots adapt their style of play depending on the state of the game. Such strategies, for example to defend in depth if the team is ahead and it is late in the game, can be pre-programmed and triggered in all robots by events they are all independently aware of (the team is 3-0 up and there are only 2 minute to go).

This is the state of the art. As mentioned above, some teams are beginning to bring in additional coordination, and we believe that this is an area in which the league will grow in the next few years.

Coodination techniques

In this section we describe our previous work on multi-agent coordination, and how we intend to apply this work to improve the coordination of the MetroBots team.

Dialogues for robot coordination

One major aspect of our work on multi-agent systems is that on dialogues between agents. Here we start from the observation that many techniques for coordinating agents require that the agents communicate, and many of the requisite communications need more than the exchange of a few terse illocutions. In other words they require some form of dialogue.

Now, when we humans engage in any form of dialogue it is natural for us to do so in a somewhat skeptical manner. If someone informs us of a fact that we find surprising, we typically question it. Not in an aggressive way, but what might be described as an inquisitive way. When someone tells us "X is true"—where X can range across statements from "It will snow heavily tomorrow" to "The Dow Jones index will continue falling for the next six months"—we want to know "Where did you read that?", or "What makes you think that?". Typically we want to know the basis on which some conclusion was reached. In fact, this questioning is so ingrained that we often present information with some of the answer to the question we expect it to provoke already attached—"I just looked up the weather on `weather.com` and it says we are in for a big snowstorm", "The editorial in today's Guardian suggests that consumer confidence in the US is so low that the Dow Jones index will continue falling for the next six months."

This is exactly the kind of *argumentation-based* communication in which we are interested. It is increasingly being applied to the design of agent communications languages and frameworks, for example: Dignum and colleagues (Dignum, Dunin-Kęplicz, & Verbrugge 2001; Dignum, Dunin-Kęplicz, & Verbrugge 2001); Grosz and Kraus (1999); Parsons and Jennings (Parsons & Jennings 1996; Parsons, Sierra, & Jennings 1998); Reed (1998); Schroeder *et al.* (Schroeder, Plewe, & Raab 1998); and Sycara (1989). Our recent work in this area has been to identify how this kind of approach can be used to carry out a range of common dialogues, including those to elicit and exchange information, and to determine the properties of such dialogues (Parsons, Wooldridge, & Amgoud 2002; 2003).

Apart from its naturalness, there are two major advantages of this approach to agent communication. One is that it ensures that agents are *rational* in a certain sense. As is argued at length in (McBurney 2002), argumentation-based communication allows us to define a form of rationality in which agents only accept statements which they are unable to refute (the exact form of refutation depending on the particular formal properties of the argumentation system they use). In other words agents will only accept things if they don't have a good reason not to.

The second advantage builds on this and, as discussed in more detail in (Amgoud, Maudet, & Parsons 2002), provides a way of giving agent communications a *social semantics* in the sense of Singh (1998; 1999). The essence of a social semantics is that agents state publicly their beliefs and intentions at the outset of a dialogue, so that future utterances and actions may be judged for consistency against these statements. The truth of an agent's expressions of its private beliefs or intentions can never be fully verified (Wooldridge 2000), but at least an agent's consistency can be assessed, and, with an argumentation-based dialogue system, the reasons supporting these expressions can be sought. Moreover, these reasons may be accepted or rejected, and possibly challenged and argued-against, by other agents.

[10]For example, the mechanism we use to decide which robot heads to the ball can leave two robots that are both relatively close marking time while they decide who should go and kick it.

In the context of MetroBots, our aim is to use argumentation-based dialogues to improve communication between robots. This might sound unlikely to benefit the robots, but the idea that it is useful for agents to explain what they are doing, which the use of arguments provides, has already been proved useful in the RoboCup arena (Riley, Stone, & Veloso 2001)[11]. Furthermore, we are not suggesting equipping the robots with the ability to engage each other in logic-driven dialogue during a game. Instead, our aim is to use the kind of dialogues systems we have explored as a specfication for the communication components of the robots, allowing the kinds of guarantee we can obtain for these systems—about the desirable outcomes of the dialogues for instance—to be carried over the dialogues between robots, and to establish dialogues that allow more or less explanation to be provided as required.

Market-based coordination

One of the major tasks that the robots on a RoboCup team have to deal with, is deciding which robots will undertake which roles. The role of goalkeeper is decided before the match begins—one robot is designated goalkeeper and only it of its team is allowed in the team's goal area—but the other robots typically change role as the game progresses.

The standard approach with the three outfield robots, adopted, for example, by the CMUPack '02 team which won the 2002 four-legged league competition, is to have one *primary attacker robot* that walks to the ball and moves it towards the goal, an *offensive supporter* which places itself near the primary attacker, but avoids getting in the way, and a *defensive supporter* which occupies space in front of the goalkeeper but behind the other players. As mentioned above, these roles are not fixed, but are dynamically allocated based upon the location of the ball and other players. Deciding the allocation of roles to robots is a resource allocation problem, and we intend to investigate the use of market-based programming mechanisms (Wellman 1993) to this problem. To do this, we will build on our ongoing work on evolving auction mechanisms (Phelps *et al.* 2002b; 2002a; 2003).

In this work, we have been using genetic programming to evolve both strategies for buyers and sellers—that is functions that determine what offers to buy and sell agents should make—and strategies for the auctioneer—that is what functions auctioneers should use to determine the price of goods given what the buyers and sellers offer. Our work so far demonstrates that such an approach can generate efficient auctions (Phelps *et al.* 2002a) and can come up with sensible pricing rules (Phelps *et al.* 2003), and we are continuing to develop tools to evolve other parts of auctions. This line of work is also capable of discovering new kinds of auction (Cliff 2001b; 2001a) which have not previously been studied.

Although market-based mechanisms like auctions can be applied to resource allocation problems (as argued in (Cliff & Bruten 1997)), it is not obvious at first sight how auctions might be used in robot soccer. However, consider a role, such as primary attacker, being a scarce resource which can be allocated to exactly one robot. If no robots are allocated this resource, then the team suffers since no robot will try to move to the ball. If several robots are allocated this resource, then the team will suffer as they interfere with one another. So auctioning, in some form, the right to take a role can be a useful mechanism. Indeed, the mechanism we use now can be considered such an auction—each robot offers its distance to the ball as the "payment" it requires to undertake the role, and the lowest offer wins. What we aim to do is to use our auction evolution tools to investigate whether there are "coordination" auctions that are more effective at allocating roles than the techniques that are already in use.

Engineering good protocols

Once we have established coordination protocols, whether by evolution or from argumentation-based specifications, we want to ensure that the protocols are sound. By that we mean that we need to ensure that the protocols do not lead to deadlock—leaving the robots unable to coordinate and thus unable to play properly—or in a situation where resources are overcommited—and, for example, several robots have taken on the same role. One way to check protocols to ensure that this does not happen is through the use of model checking (Clarke, Grumberg, & Peled 2000; Holzmann 1991).

As initially introduced, model checking is a method of ensuring that distributed processes run correctly. The processes are specified in some programming language (the PROMELA language in the case of the model checking system in (Holzmann 1997)), and conditions against which the processes need to be checked are specified (in linear temporal logic (Manna & Pnueli 1992) in the case of (Holzmann 1997)). These conditions can either be achievement conditions, in other words conditions of the form "ensure that Process A gets the resource eventually", or maintenance conditions, such as "ensure one process always has the resource". The program and conditions are put into the model checker, SPIN in the case of (Holzmann 1997) which literally constructs a dynamic logic model of the processes, and checks through their models, and then either verifies the conditions hold or gives a counter-example under which they fail.

Our previous work (Wooldridge *et al.* 2002) extended the SPIN model checker to work for programmimg languages in which one might specify agents, and this has

[11] The central idea, in brief, is that some form of explanation of an action can help one player to determine what another is doing when it would otherwise be unclear, so the additional communication and computation overhead is outweighed by the speed with which the ambiguity is resolved.

subsequently been extended (Bordini *et al.* 2003) to allow the agents to be specified in an even richer agent programming language which includes the kind of constructs we will need to use to communicate coordination information between agents. Our aim is to take this work, and use it to check our coordination protocols.

Clearly, we will already have some guarantees about the protocols from the work described above. For simple protocols we can prove, as in (Parsons, Wooldridge, & Amgoud 2002; 2003), the validity of the protocols, and the evolutionary process gives some guarantees about the protocols we evolve. However, for protocols more complex than those we can handle analytically, we believe that model checking can give us better gurantees than the evolutionary process alone.

Summary

This paper has described ways in which we aim to bring techniques from the multi-agent systems domain into the multi-robot systems of RoboCup four-legged league soccer. Drawing on our previous and continuing work on multi-agent systems, we see three lines of work which have direct applicability to our RoboCup team:

- Using argumentation-based dialogue games to develop specifications for inter-robot coordination dialogues;
- Evolving resource-allocation mechanisms for robot coordination; and
- Model-checking robot interaction protocols.

All of these, we believe, can help to improve team coordination, and we will test this hypothesis in the coming year.

References

Amgoud, L.; Maudet, N.; and Parsons, S. 2002. An argumentation-based semantics for agent communication languages. In *Proceedings of the 15th European Conference on Artificial Intelligence*.

Bordini, R.; Fisher, M.; Pardavilla, C.; Visser, W.; and Wooldridge, M. 2003. Model checking AgentSpeak. In Rosenschein, J. S., and Wooldridge, M., eds., *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*. New York, USA: ACM Press.

Clarke, E. M.; Grumberg, O.; and Peled, D. A. 2000. *Model checking*. Cambridge, MA: MIT Press.

Cliff, D., and Bruten, J. 1997. Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical Report HP-97-91, Hewlett-Packard Research Laboratories, Bristol, England.

Cliff, D. 2001a. Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, HP Labs.

Cliff, D. 2001b. Evolutionary optimization of parameter sets for adaptive software-agent traders in continuous double auction markets. Technical Report HPL-2001-99, HP Labs.

Dignum, F.; Dunin-Kęplicz, B.; and Verbrugge, R. 2001. Agent theory for team formation by dialogue. In Castelfranchi, C., and Lespérance, Y., eds., *Intelligent Agents VII*, 141–156. Berlin, Germany: Springer.

Dignum, F.; Dunin-Kęplicz, B.; and Verbrugge, R. 2001. Creating collective intention through dialogue. *Logic Journal of the IGPL* 9(2):305–319.

Grosz, B. J., and Kraus, S. 1999. The evolution of SharedPlans. In Wooldridge, M. J., and Rao, A., eds., *Foundations of Rational Agency*, volume 14 of *Applied Logic*. The Netherlands: Kluwer.

Holzmann, G. 1991. *Design and validation of computer protocols*. Hemel Hempstead: Prentice Hall.

Holzmann, G. 1997. The SPIN model checker. *IEEE Transactions on Software Engineering* 23(5):279–295.

Manna, Z., and Pnueli, A. 1992. *The temporal logic of reactive and concurrent systems*. berlin: Springer-Verlag.

McBurney, P. 2002. *Rational Interaction*. Ph.D. Dissertation, Department of Computer Science, University of Liverpool.

Parsons, S., and Jennings, N. R. 1996. Negotiation through argumentation — a preliminary report. In *Proceedings of the 2nd International Conference on Multi-Agent Systems*, 267–274.

Parsons, S.; Sierra, C.; and Jennings, N. R. 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3):261–292.

Parsons, S.; Wooldridge, M.; and Amgoud, L. 2002. An analysis of formal interagent dialogues. In Castelfranchi, C., and Johnson, W. L., eds., *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*. New York, USA: ACM Press.

Parsons, S.; Wooldridge, M.; and Amgoud, L. 2003. On the outcomes of formal interagent dialogues. In Rosenschein, J. S., and Wooldridge, M., eds., *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems*. New York, USA: ACM Press.

Phelps, S.; McBurney, P.; Parsons, S.; and Sklar, E. 2002a. Co-evolutionary mechanism design: A preliminary report. In Padget, J.; Parkes, D. C.; Sadeh, N. M.; Shehory, O.; and Walsh, W. E., eds., *Agent-Mediated Electronic Commerce IV*. Berlin, Germany: Springer-Verlag.

Phelps, S.; Parsons, S.; McBurney, P.; and Sklar, E. 2002b. Co-evolution of auction mechanisms and trading strategies: Towards a novel approach to microeconomic design. In *Proceedings of the 2nd Workshop on Evolutionary Computation and Multi-Agent Systems*.

Phelps, S.; McBurney, P.; Sklar, E.; and Parsons, S. 2003. Using genetic programming to optimise pricing rules for a double auction market. In *Proceedings of the Workshop on Agents and the Semantic Web*.

Reed, C. 1998. Dialogue frames in agent communications. In Demazeau, Y., ed., *Proceedings of the 3rd International Conference on Multi-Agent Systems*, 246–253. IEEE Press.

Riley, P.; Stone, P.; and Veloso, M. 2001. Layered disclosure: Revealing agents' internals. In Castelfranchi, C., and Lespérance, Y., eds., *Intelligent Agents VII*, 61–72. Berlin, Germany: Springer.

Schroeder, M.; Plewe, D. A.; and Raab, A. 1998. Ultima ratio: should Hamlet kill Claudius. In *Proceedings of the 2nd International Conference on Autonomous Agents*, 467–468.

Singh, M. P. 1998. Agent communication languages: Rethinking the principles. In *IEEE Computer 31*, 40–47.

Singh, M. P. 1999. A social semantics for agent communication languages. In *Proceedings of the IJCAI'99 Workshop on Agent Communication Languages*, 75–88.

Stone, P., and Veloso, M. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence* 110(2):241–273.

Stone, P.; Veloso, M.; and Riley, P. 1999. The cmunited-98 champion simulator team. In Asada, M., and Kitano, H., eds., *RoboCup-98: Robot Soccer World Cup II*. Berlin: Springer-Verlag.

Sycara, K. 1989. Argumentation: Planning other agents' plans. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 517–523.

Tambe, M.; Adibi, J.; Alonaizon, Y.; Erdem, A.; Kaminka, G.; Marsella, S.; and Muslea, I. 1999. Building agent teams using an explicit teamwork model and learning. *Artificial Intelligence* 110(2):215–240.

Thrun, S.; Fox, D.; Burgard, W.; and Dellaert, F. 2000. Robust monte carlo localization for mobile robots. *Artificial Intelligence* 128(1-2):99–141.

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

Wooldridge, M.; Fisher, M.; Huget, M.-P.; and Parsons, S. 2002. Model checking multi-agent systems with MABLE. In Castelfranchi, C., and Johnson, W. L., eds., *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*. New York, USA: ACM Press.

Wooldridge, M. J. 2000. Semantic issues in the verification of agent communication languages. *Journal of Autonomous Agents and Multi-Agent Systems* 3(1):9–31.