The 2004 MetroBots Four-legged League Team

Vanessa Frias-Martinez[1], Marek Marcinkiewicz[2], Simon Parsons[2,3], and Elizabeth Sklar[1]

[1] Department of Computer Science
Columbia University
1214 Amsterdam Avenue, New York, NY 10027, USA
vf2001,sklar@cs.columbia.edu

[2] Department of Computer Science
Graduate Center, City University of New York
365 Fifth Avenue, New York, NY 10016
mom7@cs.columbia.edu

[3] Department of Computer and Information Science
Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210, USA
parsons@sci.brooklyn.cuny.edu

Abstract. This paper describes the MetroBots Four-legged league team as it stands in May 2004. The code we are currently running is a direct extension of that developed for RoboCup 2003, but now that the robots can localize we are already able to extend the functionality considerably. We briefly describe our RoboCup-related research directions, and give pointers to additional material that covers them in more detail.

1 Introduction

The MetroBots four-legged league team was formed in September 2002 as a collaboration between[4] Michael Littman at Rutgers University, Simon Parsons at City University of New York (CUNY), and Elizabeth Sklar at Columbia University. Initially the team also included four Ph.D. students, Paul Batchis (Rutgers), Vanessa Frias-Martinez (Columbia), Dave LeRoux (Rutgers), and Marek Marcinkiewicz (CUNY), who participated fully in the project and some additional Rutgers students who had a more part-time role. This team was responsible for the initial development of the MetroBots team that participated in the 2003 American Open. The further development of the team following the American Open, and participation in RoboCup 2003, was the work of the four authors alone and they have been responsible for the development of the team since RoboCup 2003 (with some help from other Columbia and CUNY students).

This paper summarises the current state of the team, highlighting improvements in the team since RoboCup 2003, and stressing the research that we are carrying out, and will carry out alongside the development of the team. For us, this research angle is crucial to our involvement. We are basically researchers in intelligent agents and

[4] In alphabetical order.

multi-agent systems rather than researchers in robotics, and are much more interested in developing, for example, new approaches to multi-robot coordination [5] than we are in becoming the best legged-league team. This is, of course, reflected in our progress since RoboCup 2003. We have concentrated on developing research directions that are interesting to us rather than developing research directions that are soccer-specific. However, we believe that we are much better able to play soccer than we were last year and hope to demonstrate this at RoboCup 2004.

2 Current state of the team

The basic elements of the code are largely unchanged from our 2003 team. We still use the CMU motion code (though updated to use motion from the 2003 CMU team), and the rest was developed by the MetroBots team from scratch. The main change to the team is that, finally, our robots can localize, using the standard [13] Monte-Carlo approach. Though this was initially set up for the 2003 pitch with six markers, it works reasonably well when the mid-pitch markers are removed, and we are continuing to tune performance. Now that the robots can localize, we have been able to develop a wider range of robot behaviors in addition to the "run and kick" style that we adopted last year as well as being able to have the robots move to their setup positions as reqired

In particular, our current setup has two of the three outfield players play a form of "zone coverage", where they occupy broadly defensive (in their own half, between the goal and the center circle, facing the center circle) and offensive (in their opponent's half, between the goal and the center circle, facing the goal) positions, switching between these and the "run and kick" behavior depending on which robot is closest to the ball (this is determined by inter-robot communication through the game controller). At this stage of development we started to experiment (as described below) with different mechanisms for coordination.

3 Research directions

In this section we briefly describe our research agenda for this year as regards the MetroBots team. Space restrictions limit the discussion. More detail can be found in [5] (available from the team webpage `http://agents.cs.columbia.edu/metrobots`). This work can be considered as falling into two broad areas—work on individual intelligent agents and work on multi-agent systems.

3.1 Decision making and agent models

Decision making models have largely been developed for rather simple tasks (for example [7] where the case study is a one-on-one soccer game played in a small grid-world), and the techniques do not scale for real-time, complex applications like the RoboCup task. In contrast, techniques from the multi-agent systems world, like the belief/desire/intention (BDI) model, have been designed to be scaleable, but do not deal well with the uncertainties of interactions in real, physical environments. Our previous work has investigated the use of BDI models in robotics [8] and on the integration between BDI models and POMDPs [12]. The AIBOs seem to be an ideal platform on which to carry out further research on this topic.

3.2 Improving decision making over time

Our second aim in the area of individual agents is to use the AIBOs in developing and testing new approaches to learning behaviours from experience (reinforcement learning and evolutionary learning). Both evolutionary and reinforcement learning are traditionally difficult on real robots because the number of iterations it takes for a learning algorithm to converge is typically longer than the battery life of a real robot. In addition, evolutionary learning in simulation has traditionally been a problem for robotics, because the behaviours learned in simulation do not transfer well onto robots due to the uncertainties in the environment which the simulator does not model. Evolutionary learning has the additional problem of needing sometimes large populations of agents to learn from, and these cannot be run on an on-board processor. We plan to take advantage of the wireless connection with the AIBO to feed real-time learning engines that run in parallel with the AIBO. As the AIBO experiences the world, it sends environmental parameters to the learning engines. As the learning engines progress, they will send improved behaviours to the AIBO.

3.3 Dialogues for robot coordination

When robots communicate during a soccer game, it can be helpful to have them share the reasons behind their actions [11], but the limited bandwidth for communication means that such explanation should be restricted to relevant information. This is exactly the kind of *argumentation-based* communication that is increasingly being applied to the design of agent communications languages and frameworks and has been studied in our previous work [9]. In the context of MetroBots, our aim is to use argumentation-based dialogues to improve communication between robots. However, 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 dialogue systems we have explored as a specification 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 to the dialogues between robots.

3.4 Market-based coordination

One of the major tasks that our robots have to deal with is deciding which robots will undertake which roles. Deciding the allocation of roles to robots is a resource allocation problem, and we intend to investigate the use of market-based programming mechanisms [14] for this problem. To do this, we will build on our ongoing work on evolving auction mechanisms [10].

Although market-based mechanisms like auctions can be applied to resource allocation problems (as argued in [3]), 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 used in 2003 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 are doing is aim to investigate whether there are "coordination" auctions that are more effective at allocating roles than the techniques that are already in use. Our initial experiment demonstrate the value of coordination and information sharing [4].

3.5 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 [2, 6].

Our previous work [15] extended the SPIN model checker to work for programming languages in which one might specify agents, and this has subsequently been extended [1] 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 [9], 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 guarantees than the evolutionary process alone.

4 Summary

This paper has described the MetroBots Four-legged league team as it stands in May 2004. The code we are currently running is a direct extension of that developed for RoboCup 2003, but now that the robots can localize we are already able to extend the functionality considerably. Some of these extensions are illustrated by videos submitted as part of our qualification material. We briefly describe our RoboCup-related research directions, and give pointers to additional material that covers them in more detail.

References

1. R. Bordini, M. Fisher, C. Pardavilla, W. Visser, and M. Wooldridge. Model checking AgentSpeak. In Jeffrey S. Rosenschein and Michael Wooldridge, editors, *Proceedings of the 2nd*

International Joint Conference on Autonomous Agents and Multi-Agent Systems, New York, USA, 2003. ACM Press.
2. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, Cambridge, MA, 2000.
3. D. Cliff and J. Bruten. Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical Report HP-97-91, Hewlett-Packard Research Laboratories, Bristol, England, 1997.
4. V. Frias-Martinez, E. Sklar, and S. Parsons. Exploring auction mechanisms for role assignment in teams of autonomous robots. In *Proceedings of the RoboCup Symposium*, 2004.
5. V. Frias-Martinez, E. Sklar, and S. Parsons. Using multiagent coordination techniques in the robocup four-legged league. In *Proceedings of the AAAI Spring Symposium on Bridging the Multi-agent and Multi-robotic Research Gap*, Stanford, 2004.
6. G. Holzmann. *Design and validation of computer protocols*. Prentice Hall, Hemel Hempstead, 1991.
7. Michael L. Littman. Value-function reinforcement learning in Markov games. *Cognitive Systems Research*, 2(1):55–66, 2001.
8. S. Parsons, O. Pettersson, A. Saffiotti, and M. Wooldridge. Intention reconsideration in theory and practice. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000)*. John Wiley, 2000.
9. S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, (to appear), 2003.
10. S. Phelps, P. Mcburney, S. Parsons, and E. Sklar. Co-evolutionary mechanism design: A preliminary report. In J. Padget, D. C. Parkes, N. M. Sadeh, O. Shehory, and W. E. Walsh, editors, *Agent-Mediated Electronic Commerce IV*. Spinger Verlag, Berlin, Germany, 2002.
11. P. Riley, P. Stone, and M. Veloso. Layered disclosure: Revealing agents' internals. In C. Castelfranchi and Y. Lespérance, editors, *Intelligent Agents VII*, pages 61–72, Berlin, Germany, 2001. Springer.
12. M. C. Schut, M. Wooldridge, and S. Parsons. Reasoning about intentions in uncertain domains. In D. Dubois and H. Prade, editors, *Proceedings of European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Toulouse, France, 2001.
13. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.
14. M. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1(1):1–23, 1993.
15. M. Wooldridge, M. Fisher, M-P. Huget, and S. Parsons. Model checking multi-agent systems with MABLE. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, New York, USA, 2002. ACM Press.