Using Simulated RoboCup to Teach AI in Undergraduate Education

Fredrik Heintz[†], Johan Kummeneje[‡], and Paul Scerri[†]
[†]*Department of Computer and Information Science*
Linköping university
SE-581 83 Linköping, Sweden
E-mail: {frehe, pausc}@ida.liu.se
[‡]*Department of Computer and System Sciences*
Stockholm University and the Royal Institute of Technology
Electrum 230, SE-164 40 Kista, Sweden
E-mail: johank@dsv.su.se

Abstract. In this paper we argue that RoboCup is a useful tool for the teaching of AI in undergraduate education. We provide case studies, from two Swedish universities, of how RoboCup based AI courses can be implemented using a problem based approach. Although the courses were successful there are significant areas for improvement. Firstly, to help students cope with the complexity of the domain we developed RoboSoc, a general software framework for developing simulated RoboCup agents. Secondly, we propose creating close co-operation between the teachers and researchers at Scandinavian Universities with the aim of increasing the motivation of both students and teachers by providing accessible information and competence.

1 Introduction

RoboCup poses the problem of having teams of either autonomous robots or software agents play a game of football against each other. The long term goal of the project is to have a team of football playing robots take on the reigning human world champions in 2050. However, setting the aim of beating the human world champions is merely intended to provide a framework for researchers world wide. The primary goal of the RoboCup initiative is to challenge researchers to advance state-of-art technology, by posing a grand challenge with concepts familiar to the researchers[KAK+95]. A desirable side-effect of the initiative is to bring AI research to the attention of the public.

Teams from each of the RoboCup World Cups have made extensive use of undergraduate skills and time in their development, occasionally resulting in undergraduates theses (e.g. [Hei00, Lyb99, Ril99]). Many more undergraduate students come into contact with RoboCup during their undergraduate education. RoboCup has been successfully used in an AI course at Linköping university since 1997[CM99] and in two courses on agents at Stockholm University in 1999 and 2000. The courses focus on the simulated RoboCup competition, i.e. the league using software agents, because it allows students to study AI and agent technologies in an exciting framework without the expense and expertise required to work with real robots. As shown by Lund et. al. [LP99] it is also possible to use real robots in computer science

courses, even though they are more focused on robotics and the integration of hardware and software than agent oriented AI. Our experience has shown that courses using problem based learning (PBL) better exploit the educational value of RoboCup. However, due to the complexity of the domain a solid software framework (e.g. RoboSoc, see section 3) needs to be provided to the students, so that their efforts can be concentrated on the main topics of the courses.

The aim of this paper is to show the utility of using RoboCup in education. We focus particularly on the teaching of AI and of undergraduate students, but RoboCup may also applicable to other areas and groups of students. RoboCup Jr. [Lun99], which is a RoboCup league aimed at high-school students, is such an example. Section 2 describes two RoboCup based undergraduate AI courses run at different universities. We found that a significant problem with using RoboCup in education was for students to find and consume the breadth of information required, i.e. getting the background required to grasp the problems of the domain, finding solutions to those problems and using a poorly documented, code base as the basis for their development. The rest of the paper presents our proposed solutions to those problems. Section 3 describes a software library specifically designed for undergraduate students to quickly begin doing interesting work with RoboCup, thereby removing the need to work with the poorly documented research code base. Section 4 describes our ideas for a Scandinavian competence center to concentrate the Scandinavian RoboCup educational resources.

1.1   The RoboCup Simulation League

The RoboCup simulation league uses a simulator called the Soccer Server to do the soccer simulation. The Soccer Server provides a standard platform for research into multi-agent systems. The Soccer Server simulates the players, the ball and the field for a 2D soccer match. 22 clients (11 for each team) connect to the server, each client controlling a single player. Every 100ms the Soccer Server accepts commands, via socket communication, from each client. The client sends low level commands (dash, turn or kick) to be executed (imperfectly) by the simulated player it is controlling. Clients can only communicate with each other using an unreliable, low bandwidth communication channel built into the Soccer Server. The Soccer Server simulates the (imperfect) sensing of the players, sending an abstracted (objects, e.g. players and ball, with direction, distance and relative velocity) interpretation of field of vision to the clients every 150ms. The field of vision of the clients is limited to only a part of the whole field. The Soccer Server enforces most of the basic rules of (human) soccer including off-sides, corner kicks and goal kicks and simulates some basic limitations on players such as maximum running speed, kicking power and stamina limitations.

An extra client on each team can connect as a "coach", who can see the whole field and send strategic information to clients when the play is stopped, for example for a free-kick.

The Soccer Monitor connects to the Soccer Server as another client and provides a 2D visualization of the game for a human audience. Other clients can connect in the same way to do things like 3D visualization, automated commentary and statistical analysis.

2 Teaching AI with Problem Based Learning

A problem with teaching abstract science is finding illustrative examples for the concepts presented, i.e. finding real problems to solve with the abstract methods provided by the sciences. The prevailing teaching paradigm is based on pedagogy (meaning to teach children). Within this paradigm, the teacher instructs the student on how to solve a problem step by step as well as telling what problems are solvable with this problem-solving method. Thus, the teacher performs the greatest part of the cognitive processing. This is the way most children are taught in first grade.

Andragogy (the teaching of adults) differs from pedagogy in that it is more of a process. In the process the teacher becomes a coach and the students are the driving force. Closely related to andragogy is the concept of *problem based learning (PBL)*. For the purposes of this paper we take the definition of PBL to be *posing questions to be answered or problems to be solved, with the role of the teacher changed to be a coach rather than an expert teacher*[Amb92]. In PBL the student acquires knowledge in a search for solutions to a problem. The problem is formulated and given to the students who should find suitable solutions to the problem. This contrasts to pedagogy where the task is to first acquire knowledge then, hopefully, find a problem requiring the acquired knowledge.

PBL usually decreases the course-specific preparational load but increases the general preparational load for a teacher. For the RoboCup courses, as is common with PBL courses, the teachers are or were active researchers in the field hence had already done basic preparational work[Amb92]. Furthermore, as Ambury[Amb92] notes in the domain of medical education, PBL is also "fun" for everyone involved. The teacher is also made to look at the area from a new perspective, as the subject is discussed and innovative solutions proposed [Amb92]. This students-teacher discussion constitutes yet another academic conversation, yet often one with a novel angle. This has actually led to the researcher/teacher being aided in his current RoboCup research by teaching a PBL course.

A problem with PBL is that the students usually needs more support during the course. To minimize the extra work for the teachers it is very important to make goals, assignments and expectations on the students as clear and exhaustive as possible. If the requirements are unclear the students might lose interest in the course or do something totally different than what was expected by the teachers.

RoboCup is well suited for PBL because there are easily defined problems with a wide variety of potential solutions requiring further investigation and a deeper understanding. We have used PBL in all the courses and found that great interest is stimulated amongst students when they are allowed to formulate their own specific problem and solutions. As RoboCup poses a problem that has no "right" solutions, students are challenged to try to improve on both the ideas they read about and hear from their peers. In RoboCup, AI is mixed with more traditional computer science concepts such as networks and multi-threading, exposing students to a variety of topics at the same time – something with both positive and negative consequences. Murphy [Mur00] similarly argues that robot competitions, including RoboCup, can be used to improve robotics related courses.

2.1 Description of the Courses

In recent years three different AI courses have been held at Swedish universities using RoboCup as a vital element, with each course having a slightly different focus. The courses *Agent Pro-*

gramming I and II were taught at Stockholm University and the course *AI Programming* was taught at Linköping university.

In *Agent Programming I* [Int00], held Spring 2000, 10 of the 40 participants chose to take the RoboCup based assignment. The aim of the assignment was to let the students learn more about agents and AI through hands on experience. To this end students used an existing team, UBU[KLY99], to design and perform their own experiments. The resulting student reports will be made available to future students as a source of inspiration, for reference and as a launching pad for future experimentation.

Agent Programming II[Int99], held Spring 1999, had 14 participants. The students chose from three tracks, namely *Social Aspects of Agents*, *RoboCup*, and *Artificial Decision Makers*. Six students chose to take the RoboCup-track and three of ended up working on Stockholm University's 1999 RoboCup World Cup entry, UBU[BKLY99]. The other three students investigated aspects of creating intelligent agents within the context of RoboCup, looking at topics including genetic programming and neural networks.

Since 1997 an AI programming course based on RoboCup has been taught at Linköping university [CM99, AIP]. The course is project based on having students in groups of two or three designing and implementing a RoboCup team. Initially, a series of lectures on relevant topics are given. At the end of the course a competition is run among the student teams as well as recent World Cup participants. Together with a written report, the team's design and performance in the competition constitutes the basis for the student's examination and grade. Some of the AI techniques looked at in the Fall 2000 version of the course are agent architectures, agent communication languages, team work, opponent modeling, reinforcement learning and planning.

2.2 The Students' Perspective

From a student's perspective, RoboCup based courses are an exciting, challenging opportunity to take on a very non-trivial problem via whatever approach they deem most appropriate (within the constraints of course, though many students like to try to stretch those constraints!). The prospect of a competition with their peers motivates students to search for effective approaches and expend considerable energy implementing their approach. The effect is that students quickly get a relatively in depth understanding of a variety of issues and the complexities of (at least one) of the approaches in practice.

However, a major stumbling block and source of frustration and wasted time is the range of non-AI knowledge, like process programming and networking, and implementation required to try out their ideas. A large amount of code and effort is required just to get a simple agent doing things like communicating with the server, parsing the server messages, calculating its position and sending simple commands at the right time[1]. Furthermore, significant effort is required to adjust to the peculiarities of the simulation environment[Kum99, Bom99]. It is perhaps not surprising that the most requested RoboCup course improvement from students has been for more help with the practical problems of developing RoboCup agents[CM99].

To address the practical problems encountered by students, RoboSoc, a framework for developing RoboCup teams was developed[Hei00]. In 1999, the AI Programming course in Linköping used an incomplete version of RoboSoc. In an informal post course evaluation

---

[1]The best World Cup teams might have several thousand lines of code performing such tasks

students reported that RoboSoc allowed them to focus on the issues they were interested in[2]. The course leader observed that students had less problems creating working teams and implementing their own skills and decision making within the RoboSoc framework. The developed teams were of a surprisingly good quality, being able to compete with mid-field teams from the 1999 World Cup.

At Stockholm University the existing team UBU was made available to the students as a basis for their work. The students were positive about the RoboCup assignment, generally finding it to be very relevant to the course material. For example, some students found using RoboCup gave them insights in the application of neural nets in real-time environments they might not have gotten from a textbook or carefully structured assignment. The students appreciated working on a real software project as opposed to working on an artificial assignment, though some commented that it was a too large a system to grasp in a couple of weeks Other students were, however, a little disappointed that they could not implement their own teams from scratch.

3 RoboSoc

State of the art RoboCup simulation teams involve tens of thousands of lines of code. Usually the agent is divided into different sub-systems for things such as the server interface, world model, low level control, high level tactical control and co-ordination with teammates. The sub-systems work together to produce the overall agent behavior. Clearly, the quality of the overall agent behavior is intimately dependent on the quality of each of the subsystems – good tactical choices are irrelevant if based on an incorrect view of the world, good low level control decisions are useless if not communicated in a timely manner to the Soccer Server.

Developing each of these sub-systems to a high quality takes more effort than is reasonable for a single undergraduate course. Hence there is a need for a well documented, well structured and generic library which helps and encourages developers to create and share their code. RoboSoc[Hei00] is such a library. Because of the modular design the students can look in depth at a single area rather than being required to learn a little about a all the areas. And since RoboSoc is designed to make it easy to share code and modules between teams the developers can contribute to the library incrementally improving the library over time, giving the future users a better base on which to build.

When used with PBL RoboSoc reduces the chance of students getting frustrated or confused by having too many possibilities and tasks, since it creates a framework within which the students can structure their ideas and development efforts. It also makes it possible for the students to help each other since they all work with same system.

As well as being useful for PBL style course, we believe that a RoboSoc type library can be used by teachers who want to develop more structured laboratory exercises. RoboSoc provides the framework within which specific tasks can be defined for students. For example, one exercise might be to use a rule based system for determining the current game status given the agents incomplete, uncertain view of the world. The output of the rules might be used to turn on appropriate, existing agent behaviors. Hence, the students would have the opportunity to develop a relatively small sub-system (i.e. the high level tactics selection) but see the results within a much larger, more complex and more interesting system.

---

[2]There were, however, some concerns about the implementation and documentation of RoboSoc.

Designing exercises around a standard library makes it possible for teachers to share their exercises with other courses that also uses that library. Being able to share exercises shares the manpower required to develop new and interesting exercises. Since RoboSoc has a modular structure, specific sub-systems can be considered independently and used in laboratory exercises without students needing to have an in depth understanding of the rest of the system. Conversely, the modularity allows the student implementations to be used with the larger system so that they can see the global effect of their local changes. Sub-systems developed or improved in one year can be made available to other students both in the current year and in the future, allowing the library, and hence the exercises, to improve over time.

4   Extending the Use of RoboCup in Scandinavian Education

Above we have argued that simulated RoboCup is useful for teaching AI to undergraduate students and that we believe that RoboCup is applicable to other subjects as well. At Linköping we are investigating how the RoboCup initiative might be used for real-time systems courses. The basic idea is to use the same framework for both AI courses and real-time courses. It is likely that real robots (rather than simulated ones) would be used. Building on the idea of RoboSoc a software and hardware platform would be developed. Carefully designed modularity would be required to allow both AI and real-time students to focus on the aspects relevant to their courses while leveraging the efforts of the other students to create an overall team that plays good robot football. The overriding goal of creating a good football team maintains the original motivation of using RoboCup by providing incentive and scope for students to push themselves.

Undertaking to develop a RoboCup infrastructure upon which a variety of interesting, challenging course assignments can be based is no simple task. Many pedagogy questions need to be answered. For example how do we create an environment which best supports the students in their learning activities? How do we express the goals and requirements to minimize the misunderstandings and needs for extra support? And, what assignments should we give the students in order to motivate, direct and test their learning? Regardless of the specifics of the answers much infrastructure, both physical (i.e. software and hardware) and intellectual (i.e. expertise, experience and support material) will need to be developed to provide a useful, stimulating environment.

However we firmly believe that it is worth pursuing answers to these questions, the reward will be worth the effort. As a concrete proposal for how Scandinavia can leverage the competence we already have to develop RoboCup as an important tool for Computer Science (emphasizing AI) education we propose a Scandinavian RoboCup Competence Center.

4.1   Scandinavian RoboCup Competence Center

The purpose of a Scandinavian RoboCup Competence Center (SRCC) would be to provide a focal point for RoboCup activities in Scandinavia. It is envisioned that the SRCC would be an active network of teachers who shared their resources and energy. Working together in synergistic co-ordination the members of the SRCC could develop RoboCup as a basis for a variety of courses across Scandinavia without the infrastructure and competency problems any university on its own would face.

Notice that, RoboCup per se, as the standard framework for developing such a competence

center is not the critical point. The critical point is to combine the efforts and competence of educators across Scandinavia to develop an educational environment that is exciting and stimulating but that could not be developed by any group on their own. RoboCup happens to be an ideal environment for such work (for reasons given above) but other environments may be found which provide similar benefits.

An indication of the potential for a successful SRCC is the ongoing effort between Stockholm University, the Royal Institute of Technology, Örebro University, and Blekinge Institute of Technology in Team Sweden [Gut99]. The group, consisting of several professors, doctoral students, and a fairly large number of undergraduate students distributed over 5 universities, collaborates at a research level on RoboCup. The team collaborated on winning a grant from Sony for legged "AIBO" robots and jointly developed the software to compete with the robots in both the 1999 and 2000 World Cups.

A second indication of the benefits a SRCC might have is the success of the existing Scandinavian RoboCup Committee. Unlike the proposed SRCC, the Scandinavian RoboCup Committee is focused on the research aspects of RoboCup. The utility of having such a committee was clearly shown with the committee's highly successful running of RoboCup World Cup in 1999 in Stockholm – something that any single university may not have had the resources to do alone.

To make more concrete the idea of SRCC we detail two functions that such an organization might have. One of the things the competence center could do is organize a Scandinavian RoboCup tournament. Such a competition would promote RoboCup as well as providing motivation for students involved in courses. The World Cup provides a similar function for international researchers. The World Cup both give teams a specific development goal, i.e. achieve some performance level in the competition and provides a fantastic forum for discussion – because all researchers have worked on the same platform. A Scandinavian Championships would offer similar benefits to undergraduate students. Since the tournament would be held in Scandinavia the costs for travel can be limited and thus make it possible for the universities to allow the students to attend the tournaments on site. The tournament will most probably increase the chances of the Scandinavian teams in the RoboCup World Cup, which has the desirable consequence of improving the image of Scandinavian universities in the international community.

To test this idea on a smaller scale the Swedish AI Society (SAIS) organized the first Swedish Championship in simulated RoboCup in May. It was quite a success with six teams from three different universities in Sweden. Due to the success of this years competition the second championship will be held in Skövde the next spring, also organized by SAIS at their yearly workshop.

4.2   A RoboCup Laboratory

The second concrete idea for SRCC would be the creation of a physical RoboCup laboratory. The laboratory might be either a common infrastructure realized in multiple locations or a single physical location. The primary goal of such a laboratory would be to further concentrate the efforts of students and educators. However, creating such a laboratory, might encourage researchers and teachers from other disciplines to become interested in RoboCup, allowing interesting synergies to emerge. We believe that the RoboCup framework can be used as a teaching aide for courses related to AI, real time systems, distributed systems, robotics and

possibly even others. Students, may, as a side effect, become aware of the bigger picture of their education, instead of seeing the education as series of unrelated courses aimed at solving very specific problems. If the appropriate architecture could be set up it would be a major improvement to the use of RoboCup in education, since it would provide a unified framework for a very large class of problems and subjects.

5   Summary

RoboCup is an excellent domain for demonstrating AI techniques and to give students an understanding of what AI is and how it is applied in real applications. Our experience with RoboCup in education has shown that it is useful in PBL-style courses but we also believe that it may be useful for traditional practical exercises in a variety of computer science disciplines. We, as have others, have found this way of teaching, especially when RoboCup is used, to be fun and rewarding for both teachers and students. A problem with using RoboCup in education is the time it takes to come to terms with the peculiarities of the domain and get away from uninteresting low level issues[Kum99, Hei00, Bom99]. To reduce this problem we have developed RoboSoc a framework for developing simulated RoboCup agents.

However, to fully realize the potential of the RoboCup concept in education we need to pool educator resources to develop a better infrastructure and share competence. Therefore we propose the establishment of a Scandinavian RoboCup Competence Center that can focus and concentrate the RoboCup resources around the region. Such a center would aim to cooperatively develop the infrastructure required for efficiently using RoboCup in education across a variety of disciplines. We hope that this infrastructure will improve the AI education in Scandinavia, interest more students in AI, and provide the researchers and teachers with a very interesting, challenging and rewarding framework to work in.

References

[AIP]   AI-Programming course web-page. `http://www.ida.liu.se/~TDDA14/`. verified January 5th 2001.

[Amb92]   George Ambury. Beginning to Tutor Problem-Based Learning: A Qualitative Investigation of Andragogy in Medical Curriculum Innovation. presented at the annual meeting of the CASAE, Ottawa, Canada., June 1992.

[BKLY99]   Magnus Boman, Johan Kummeneje, David Lybäck, and Håkan L. Younes. UBU Team. In *RoboCup-99 Team Descriptions*, pages 133–138, 1999.

[Bom99]   Magnus Boman. Agent Programming in RoboCup'99. *AgentLink NewsLetter*, (4), November 1999.

[CM99]   Silvia Coradeschi and Jacek Malec. How to make a challenging AI course enjoyable using the RoboCup soccer simulation system. In Minoru Asada and Hiroaki Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, pages 120–124. Springer Verlag, Berlin, 1999.

[Gut99]   Christian Guttmann. A software architecture for four-legged robots. Master's thesis, Department of Computer and Systems Sciences, Stockholm University and the Royal Institute of Technology, 1999.

[Hei00]   Fredrik Heintz. RoboSoc a System for Developing RoboCup Agents for Educational Use. Master's thesis, Department of Computer and Information Science, Linköping university, March 2000.

[Int99]   Agent Programming II Course Home Page.   `http://www.dsv.su.se/~mab/7v99/Index.html`, 1999. verified January 5th 2001.

[Int00]   Agent Oriented Programming Course Home Page. `http://www.dsv.su.se/˜mab/4v00/`, 2000. verified January 5th 2001.

[KAK+95] Hiroaki Kitano, Minoru Asada, Yasou Kuniyoshi, Itsuki Noda, and Eiichi Osawa. RoboCup: The robot world cup initiative. In *Proc. of IJCAI-95 Workshop on Entertainment and AI/Alife*, 1995.

[KLY99]   Johan Kummeneje, David Lybäck, and Håkan L. Younes. UBU - an object-oriented RoboCup Team. In Johan Kummeneje and Magnus Boman, editors, *Int7 1999 Papers*. 1999.

[Kum99]   Johan Kummeneje. Simulated Robotic Soccer and the Use of Sociology in Real Time Mission Critical Systems. In L. R. Welch and M. W. Masters, editors, *Proceedings of RTMCS Workshop*. IEEE, December 1999.

[LP99]   Henrik Hautp Lund and Luigi Pagliarini. Robot soccer in education. *Advanced Robotics Journal*, 13:8:737–752, 1999.

[Lun99]   Henrik Hautop Lund. Report from RoboCup'99. `http://www.daimi.au.dk/˜hhl/RoboCupJr.html`, 1999. verified January 5th 2001.

[Lyb99]   David Lybäck. Transient Diversity in Multi-Agent Systems. Master's thesis, Department of Computer and Systems Sciences, Stockholm University and the Royal Institute of Technology, September 1999.

[Mur00]   Robin Murphy. Using Robot Competitions to Promote Intellectual Development. *AI Magazine*, Spring, 2000.

[Ril99]   Patrick Riley. Classifying adversarial behaviors in a dynamic, inaccessible, multi-agent environment. Technical report, Computer Science, Carnegie Mellon University, 1999. CMU-CS-99-175.