Lund, H. H., and Pagliarini, L. (2001) Edutainment Robotics: Applying Modern AI Techniques. In Proceedings of International Conference on Autonomous Minirobots for Research and Edutainment (AMIRE-2001).

Edutainment Robotics: Applying Modern AI Techniques

Henrik Hautop Lund   Luigi Pagliarini

Maersk Mc-Kinney Moller Institute for Production Technology
University of Southern Denmark, Campusvej 55, 5230 Odense M., Denmark
Tel: +45 6550 3574, fax: +45 6550 7697
hhl@mip.sdu.dk   luigi@mip.sdu.dk
http://www.mip.sdu.dk/~hhl

Abstract

With the development of cheap robotic tools, it has become possible to allow children to use robots as a toy and in educational environments. For the purpose of increasing children's awareness and knowledge about technology, we have developed tools that allow them to interact with robots in an easy and straightforward manner, e.g. exemplified through our design and realisation of RoboCup Junior. Some of these techniques arise from the fields of evolutionary computation, adaptive systems, agents, and artificial neural networks and we show how they can be used in edutainment robotics in order to provide easy access to the robot technology. The user-guided approaches that we developed include user-guided behaviour-based systems, user-guided evolutionary robotics, user-guided co-evolutionary robotics, and morphological development. All these techniques are applied to allow children to develop their own robot behaviours in a very easy and fast manner.

Introduction

For the last 15 years, there has been a strong development of adaptive robotics, starting from the introduction of behaviour-based robotics by Brooks [3] and the more popular description of such systems by Braitenberg [2] (see also [4] for a review of the use of behaviour-based systems for robot control, and [1] for a thorough text book). There have been numerous examples of how to use artificial neural networks as robot controllers (e.g. [8, 19]), evolutionary computation for development of the robot controllers (e.g. [5, 6, 20]), and different kinds of behaviour-based systems for control (e.g. [1,3]).

Most of these systems are supposed to adapt to the environment, and often one can consider an adaptation to the ecological niche (see [22]). The advances in making adaptation have been considerable. However, most often the robotic systems and the adaptation of these are tested only on simple problems such as obstacle avoidance, homing, line following, etc. which is in line with Braitenberg's [2] suggestions, but not complex enough to attract the attention over longer periods of time in an entertainment or edutainment application or to fulfil the goals of most industrial applications.

A second problem appears in the autonomous systems approach. Often, the goal is to achieve fully autonomous robots, both in the development and the behaviour. This is highly desirable from a theoretical point of view and in some fully autonomous system applications, but sometimes, in other applications, it may turn out to be less desirable. For instance, in entertainment or edutainment that involves construction, the user would like to be able to direct the development of the system, and in production systems, the worker in a production hall should be able to re-configure the robot for flexible production.

We have addressed these two issues regarding adaptive robotic systems by developing methods for achieving complex robot behaviours and for interaction with the autonomous system. In the following, this development is described through example implementations of entertainment robots and user-guided approaches. Essentially, all examples implement intelligent agents in entertainment and edutainment.

Intelligent Agents in Entertainment

An intelligent agent can be defined as a system with "intelligence" that performs actions or interactions. Previously, such agents have been used in the entertainment industry mainly in computer games for achieving life-like behaviours (e.g. [25, 26]). For instance, in a computer game an agent may be used as a character with whom the user is supposed to interact. The agent may learn about the reactions of the user in order to increase its skills and in this way better challenge the user (as opposed to games where the agent is static and not adapting through the user's interactions).

If we define the user as an agent, we can say that the traditional use of intelligent agents has been a set-up involving two intelligent agents, namely the user and the computer game agent. We envision the modern use of intelligent agents to broaden this use to physical interactions with the artificial agent and the interplay between more intelligent agents (e.g. [9]). In order to achieve physical interactions, it is necessary to move the intelligent agent from the pure, virtual reality in the computer to the physical reality. It is possible to move the actions and interactions between a user and a virtual, intelligent agent to a physical interaction through haptic interfaces. Another possibility is to make the complete transfer to the physical reality by constructing robots for this purpose. Examples of such robots are the SONY AIBO, LEGO MINDSTORMS robots, I-Cybie, etc. These physical robots facilitate interaction between two intelligent agents (user and robot) in the physical world.

There are significant differences in between the different entertainment robots mentioned above. In some cases, the robots are fully autonomous both in development and behaviour (e.g. Furby) and so give *no* possibility for development by the user, in some cases there are *limited* possibilities for development by the user (e.g. I-Cybie, AIBO), and in other cases there are *extensive* possibilities for development by the user (e.g. LEGO MINDSTORMS, FischerTechnic robot). Here, we will concentrate on the latter kind of entertainment robots, since we view these systems to best facilitate an educational approach in applications for children.

Figure 1 The Pacman game with LEGO MINDSTORMS

In order to investigate the possibilities of transferring the intelligent agent from the traditional use in a computer game to the physical reality, we made a number of initial experiments with LEGO MINDSTORMS robots. They were displayed and used by children during RoboCup'98 in Paris as the first RoboCup Junior event [11, 15]. Especially the well-known computer game from the early 1980'ies called Pacman was developed as a robot game. Pacman is the game where a yellow cheese is controlled to move around in a maze while being chased by a number of ghosts. In the robot game, there would be two red ghost robots and a yellow cheese robot moving around in a maze made by attaching black adhesive tape on a white floor (a white plastic tablecloth). All three robots were programmed using the behaviour-based approach. In the behaviour-based approach, the robot programmer first designs a low level of competence, implements and debugs this level. When this level is fully functional, the programmer can start adding new levels of competence one on top of each other. In the case of the Pacman game, the robots had to be programmed to have the following competencies: avoid lines, avoid when colliding, move forward, and turn in junctions. These behaviours would allow the robots to move around in the maze. By adding an extra layer, one could design a goal-directed behaviour. In this way, it was possible to design a fully autonomous display of three robots moving around in the maze. However, a fully autonomous Pacman game would be no fun for the children. So an extra layer of competence was introduced in the yellow cheese robot, namely the goal-directed behaviour through interaction by the child user. A joystick was made out of LEGO MINDSTORMS, and the commands received by the yellow cheese robot through infra-red communication from the joystick would enter as high-level commands in the behaviour-based system. In this way, the user would be able to direct the behaviour of the (no longer fully autonomous) robot, and try to direct it to the centre of the maze (the goal position) while avoiding being hunted down by the two red ghost robots, i.e. the two fully autonomous robots. Apart from fulfilling the goal of providing a fun and interesting physical robot game, the approach showed one of the advantages of the behaviour-based approach, namely the division into behaviours and the construction of layer after layer. It was comparably simple to add new layers of competencies in order to provide the necessary behaviours and to allow the interaction between the two intelligent agents: the child and the robot. Further, we have found this game to be, in some way, educative with respect to sensory-motor coordination learning and, since one has to work out his/her strategy within a very short time, useful for learning about decision making, focusing attention and, finally, the relationship between autonomous and supervised agents' behaviour. The success of the Pacman game prompted us to use similar approaches in other applications such as a robot fashion show, artist robots, and musician robots.

Interactive LEGO Football for RoboCup Junior

Interaction plays a major role, when we are concerned with children learning by getting hands-on experience. Indeed, there may be a conflict between much modern research on developing autonomous systems, and the educational research putting emphasis on interaction, e.g. in guided constructionism [10]. Also, classical constructionism with its roots in the work by J. Piaget suggests that the best way to learn about an artefact is to actually build the artefact.

Therefore, for our RoboCup Junior set-up during RoboCup'99 in Stockholm, we developed a LEGO MINDSTORMS robot soccer game with a wider focus on interaction. The use of any external sensing device (e.g. overhead camera as used in RoboCup Small-size League) was avoided by making a special transparent ball with infra-red transmitters of approximately the same wavelength as the detectors in the LEGO MINDSTORMS light sensors. This facilitated the perception of the ball, while the recognition of position in the field was facilitated by colour (grey) codes on the floor.

The aim of our RoboCup Junior game was to allow children to get hands-on experience with robotics, and for this purpose we set up a LEGO MINDSTORMS robot soccer game for children. We developed the *user-guided behaviour-based approach* [16] in order to allow non-expert users to develop their own robots in an easy and fast manner. Indeed, using this approach, children of the age 7-14 were able to develop their own LEGO MINDSTORMS robot soccer players to play in nice and friendly tournaments with 60-90 minutes of development time! In a user-guided behaviour-based system, it is the system developer who takes care of the difficult robotic problems, while the end-user is working on a higher abstraction level by making the coordination of primitive behaviours.

As mentioned, the programming environment for the RoboCup Junior was made with emphasis on allowing children (between 7 and 14 years of age) to develop their own robot soccer players' behaviour. We found the behaviour-based approach to be an excellent inspiration for achieving this. Especially, we used the concepts of low and high levels of competence, or primitive behaviours and arbitration. We, as developers, provide the primitive behaviours to the children, while they work (play) on a higher level with the arbitration of the primitives. Hence, the difficult task of designing low level primitives that includes sensor interpretation is done a priori by the programmer (so the children get to do the easier and funny part of coordination rather than doing low level programming). For instance, the interpretation of analogue values on the input channels is done in the primitive behaviours, which might provide the user with a behaviour such as "Find the Ball". The designer of the system programs the motors to allow the robot to, for example, turn around and stop when receiving values such as 637 and 655 on two of the input channels. But the user is simply coordinating the primitive behaviours. This user-guided behaviour-based system is described in further details in [16].

Figure 2. The programming environment for Interactive LEGO Football as used in RoboCup Junior. We used the behaviour-based approach, and developed primitives (the behaviours on the left), and allowed the children to make higher level strategies using these primitives. With this system, children from 7 to 14 years of age were able to develop their own robot footballers within 30-60 minutes. Copyright Lund and Pagliarini, 1999.

The user-guided behaviour-based system for RoboCup Junior is called Interactive LEGO Football (ILF), and has been used with great success at tournaments during RoboCup'99, MindFest'99, RoboCup EURO 2000, and numerous local events. This game has been found to be useful when learning about a behaviour plan and its relationship with a "noisy" world and a "noisy" machine (i.e. the robots).

Figure 3. Some children playing with the LEGO robot soccer players that they have developed with ILF within 60 minutes.

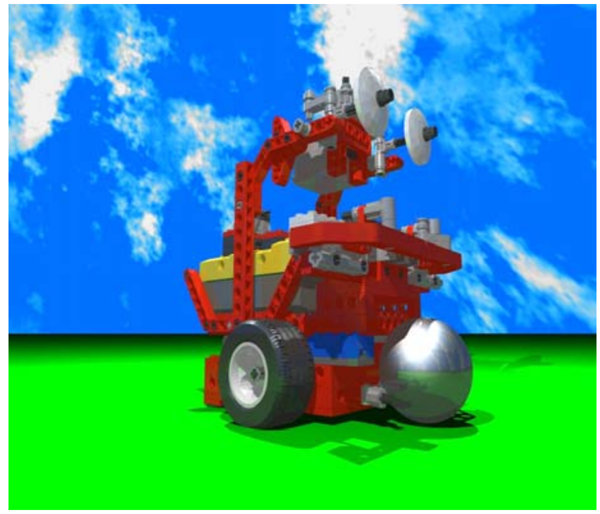

Figure 4. The LEGO robot soccer player for ILF. We provide building plans for making this robot in order to facilitate the participation in the game. See http://www.mip.sdu.dk/~hhl/RoboCupJr/Build/

Breeding Robotics

In the case of ILF, the easy development of these robot behaviours is dependent on the available tool. In the RoboCup Junior set-up, the childrens' task was facilitated by our programming environment, in which children would coordinate primitive behaviours rather than hand-coding complex behaviours from scratch. However, one major drawback remains, namely that the children have to be able to read in order to use the system. Therefore, we have worked on developing user-guided evolutionary robotics to fully avoid the necessity to learn syntax and semantics of a programming language before being able to develop robot behaviours and, doing so, allow children to concentrate on robots' behavioural strategies, the most educational aspect of the game.

Essentially, we are exploring the concept of *development without programming* by children, and especially at the case of developing robot control systems, so this is a case study of breeding robotics. In breeding robotics, the machines are products of the interaction of the artificial evolutionary process and the breeders (in this case children) that try to help, direct and select [13]. The evolutionary robotics approach has shown that in some cases, given a mathematically described fitness function, it is possible to achieve an automatic development of robot controllers. However, it makes no sense to demand children to construct the mathematical fitness function. So we applied an interactive genetic algorithm to the problem of developing robot controllers and achieved an evolutionary robotics approach that allows children without any programming knowledge to develop controllers for LEGO robots [14]. We used neural networks as robot controllers, and found that combining the interactive genetic algorithm with a kind of reinforcement learning -- development at the evolutionary time scale combined with life-time development -- reduces the development time drastically. Hence, we overcome one of the major drawbacks of the interactive genetic algorithm, namely the development time.

The general idea is a *User-guided Evolutionary Robotics* approach by which children can develop robot controllers in the simulator by choosing among different robot behaviours that are shown on the screen, and then, when they are satisfied with the simulated robot's behaviour, download the developed control system to the real LEGO robot and further play with it in the real environment.

The user-guided evolutionary robotics approach is inspired by our previous work using interactive genetic algorithms to evolve simulated robot controllers, facial expressions and artistic images (see e.g. [21, 24]). In this approach, there is no need of programming knowledge, since all the end-user has to provide is a specification of preference of the solutions suggested graphically on the screen. Hence, there is no description of a fitness function, but the user performs the selection in the genetic algorithm.

In order to use the user-guided evolutionary robotics approach, it is necessary to simulate the robot in its environment, make selective reproduction in the simulator, and then transfer to the physical robot. As described in [12, 18], it is possible to build an accurate simulator that allows very good transfer from simulation to reality by basing the simulator on the robot's own samplings of sensor and motor responses. The disadvantage is that data has to be collected. In the construction of the simulator, this data had to be collected for the different sensors and different motor configurations. For instance, we had to measure the motor response for each individual LEGO robot design that we wanted to use in the simulator. This is the disadvantage of the approach. (However, we are currently exploring new adaptive techniques to overcome this problem.)

The sensor and motor data was collected in a similar way to that described in [12, 18], and the collected data was put into look-up tables that is used by the simulator to look up specific sensory readings and displacements of the simulated LEGO robot.

Our first experiments showed that we could develop simple robot behaviours such as obstacle avoidance, line following, etc. for LEGO robots with the user-guided evolutionary approach [14]. Here, children chose a subset (three) of simulated robots in a population (of nine simulated robots) to reproduce generation after generation before downloading the final result to the real LEGO robot. In order to show the feasibility of the user-guided evolutionary robotics approach, we wanted to test it with more complex tasks such as the RoboCup Junior game. Hence, we extended the approach to allow children to evolve complex behaviours for the LEGO MINDSTORMS robots. We call this implementation the Toybots Breeder. Inspired by the successes of previous work on evolvable behaviour-based systems [7] and the user-guided behaviour-based system for the RoboCup Junior [16], we decided these two approaches to be the starting point of our user-guided evolutionary robotics approach for allowing children to develop complex behaviours.

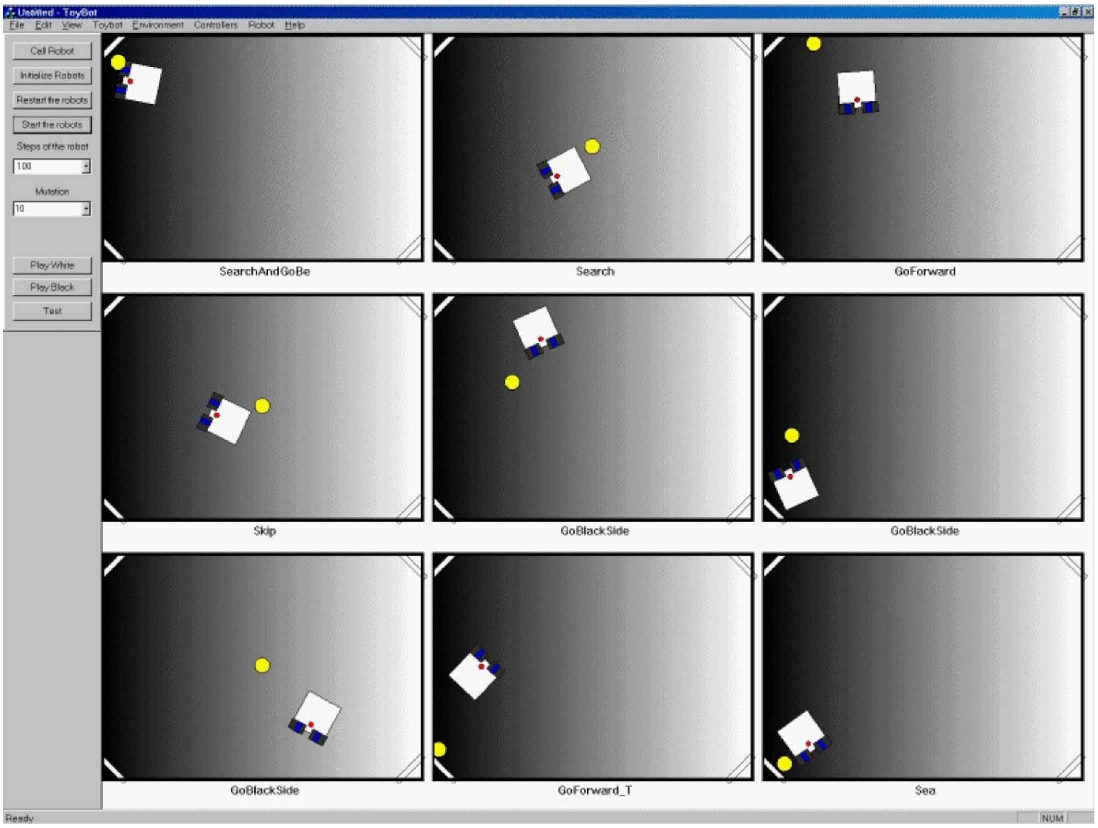

Figure 5. The Toybots simulator for the LEGO MINDSTORMS RoboCup Junior. Here, children can develop complex robot soccer behaviours before downloading these behaviours to the real LEGO MINDSTORMS robot to be used in competitions. Copyright Lund 1999.

Essentially, we use the primitive behaviours (e.g. 'Go Forward', 'Find Ball', 'Go Midfield') that we developed for the ILF program as the building blocks in the genotype, and allow the interactive evolution to develop the coordination of these behaviours. In the simplest case, the coordination can be sequencing a number of the primitive behaviours. In this case, in order to build a simulator, we simulate each of the primitive behaviours, and we simulate the ball movement --- using the simulation technique described above. As before, we can now show a population of simulated LEGO MINDSTORMS robots on the screen with the simulation of the field and the ball, and allow the children to select the ones that play the kind of soccer that they are interested in, see Figure 9. User tests in our lab showed that children of 8 years of age were able to develop robot soccer players and enjoy the game with the Toybots Breeder.

There are a number of research issues that have to be addressed in order to ensure feasible user-guided evolutionary robotics. The most important issue is user fatigue: how can we avoid demanding the user to select from a big population for numerous generations. If the selection process becomes too long, the user will experience fatigue. In our case, by using primitive behaviours as the building blocks, we achieve fairly fast evolution, which is essential when children are involved. However, it is not given that the building blocks should necessarily be at this level. For the simple behaviours such as obstacle avoidance and line following, we used connection weights in a neural network as the building blocks, but for the more complex task of robot soccer we found it necessary to increase the complexity of the building blocks to become primitive behaviours.

Context development

Based on the positive experience with the robot soccer experiments, we developed a full RoboCup Junior league for RoboCup. Before RoboCup 2000 in Melbourne, Brian Thomas a.o. interacted with almost 40 school classes who used RoboCup Junior as an activity either during school hours or after. As we expected, interviews with teachers (see [23] for a report describing response by teachers regarding the use of RoboCup Junior) showed an undesirable gender gap. It turns out that most participants for the robot soccer games are boys.

Figure 6. Robot Dance Performance made by children from Australia during RoboCup Junior 2000. Here, the children design both the story line, the environment, the robot, etc. See the RoboCup Junior Official Site (http://www.artificialia.com/RoboCupJr/)

Oppositely, as we experienced a couple years earlier with internal tests and group meetings with school children, there is a wide range of tasks that are attractive independently from gender. Among those, we noticed that one of the most inspiring robot based games could be a dance performance. We exemplified such performances through the robot fashion shows and internal dance performances in which robots would communicate in order to coordinate the dance. Robot dancing puts emphasis on combining technical skills with other skills, e.g. putting things into the right context, cooperation and performance (rather than competition). For the robot dancing, we experience a much more equal distribution of participants between the two genders.

This may be because dance itself is practiced by both gender in real life and, therefore, the level of projection of children Ego should be at a higher level. Nevertheless, educational issues are not missing at all since the easy enthusiasm children reach while working at a robot dancer (this is also due to the music, colour, creativity, fashion design, and emotion expression) nicely drive them to overcome the task's technical difficulties.

Conclusion

The numerous practical experiments described here have shown that it is possible to use modern artificial intelligence in edutainment robotics (see also [17]). However, most often the methods have to be re-designed in order to meet the demand of providing both fun and educational experiences for the users. Hence, we have developed a number of user-guided approaches based on behaviour-based systems, evolutionary computation, neural networks, and multi-agent systems. Further, research in the field of embodied artificial intelligence suggest that we should put emphasis not only on optimisation of the robot controllers but look at the interaction between controller, morphology, material, environment and users' characteristics. So we should allow the user the freedom to manipulate as many of these parameters as possible, and not only interact with a "static" robot or only develop the robot controller. This corresponds with the educational practises of allowing the user/student to construct the artefact, the functionality and the context in which the artefact is placed.

Acknowledgements

A number of experiments were developed in collaboration with colleagues and students. They include Orazio Miglino, Kasper Støy, Esben Østergaard, Jesper Blynel, Mikkel Yde Kjær, Rene Schalburg, Henrik Madsen, Jesper Rasmussen, Jakob Vesterstrøm, and M. Rooker. All views presented in this article are those of the authors, and do not necessarily present those of any sponsor or funding agency.

References

[1] R. Arkin. Behavior-Based Robotics. MIT Press, Cambridge, MA, 1998.

[2] V. Braitenberg. Vehicles: Experiments in Synthetic Psychology. MIT Press, Cambridge, MA, 1984.

[3] R.A. Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, 2(1):14--23, 1986.

[4] R. A. Brooks. Evolutionary Robotics; Where From and Where To. Evolutionary Robotics: From Intelligent Robots to Artificial Life ER'97, AAI Books, Ontario, Canada, 1997, pp. 1--19.

[5] D. Cliff, I. Harvey, and P. Husbands. Explorations in evolutionary robotics. Adaptive Behavior, 2(1):73--110, July 1993.

[6] D. Floreano and F. Mondada. Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural Network Driven Robot. In D. Cliff, P. Husbands, J. Meyer, and S. W. Wilson, editors, From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB94), Cambridge, MA, 1994. MIT Press, Bradford Books.

[7] W.-P. Lee, J. Hallam, and H. H. Lund. Learning Complex Robot Behaviors by Evolutionary Approach. In A. Birk and J. Demiris, editors, Proceedings of 6th European Workshop on Learning Robots, LNAI 1545, Heidelberg, 1997. Springer Verlag.

[8] F. L. Lewis. Neural Network Control of Robot Manipulators. IEEE Expert/Intelligent Systems & Their Applications, Vol. 11, No. 3, June 1996

[9] H. H. Lund. AI in Children's Play with LEGO Robots. In Proceedings of AAAI 1999 Spring Symposium Series, AAAI Press, Menlo Park, CA, 1999.

[10] H. H. Lund. Robot Soccer in Education. Advanced Robotics Journal,13:8, 737-752, 1999.

[11] H. H. Lund, J. A. Arendt, J. Fredslund, and L. Pagliarini Ola: What Goes Up, Must Fall Down. In Proceedings of Artificial Life and Robotics 1999 (AROB'99), pp. I-9 - I-15, ISAROB, Oita, 1999.

[12] H. H. Lund and O. Miglino. From Simulated to Real Robots. In Proceedings of IEEE 3rd International Conference on Evolutionary Computation. IEEE Press, 1996.

[13] H. H. Lund and O. Miglino. Evolving and Breeding Robots. In Proceedings of First European Workshop on Evolutionary Robotics. Springer-Verlag, 1998.

[14] H. H. Lund, O. Miglino, L. Pagliarini, A. Billard, A. Ijspeert. Evolutionary Robotics – A Children's Game In Proceedings of IEEE 5th International Conference on Evolutionary Computation. IEEE Press, 1998.

[15] H. H. Lund and L. Pagliarini. Robot Soccer with LEGO Mindstorms. In Asada (ed.) Proceedings of RoboCup'98, Springer-Verlag, Heidelberg, 1999.

[16] H. H. Lund and L. Pagliarini. RoboCup Jr. with LEGO Mindstorms. In Proceedings of Int. Conf. On Robotics and Automation (ICRA2000), IEEE Press, NJ, 2000

[17] O. Miglino, H. H. Lund, and M. Cardaci. Robotics as an Educational Tool. In Journal of Interactive Learning Research 10:1, 25-48, 1999.

[18] O. Miglino, H. H. Lund, and S. Nolfi. Evolving Mobile Robots in Simulated and Real Environments. Artificial Life, 2(4):417-434, 1996.

[19] T.M. Mitchell and S.B. Thrun. Explanation-Based Neural Network Learning for Robot Control. In Hanson, Cowan, and Giles (eds.), Advances in Neural Information Processing Systems 5, Morgan-Kaufmann Press, 1993, pp. 287-294.

Lund, H. H., and Pagliarini, L. (2001) Edutainment Robotics: Applying Modern AI Techniques. In Proceedings of International Conference on Autonomous Minirobots for Research and Edutainment (AMIRE-2001).

[20] S. Nolfi, D. Floreano, O. Miglino, and F. Mondada. How to evolve autonomous robots: Different approaches in evolutionary robotics. In R. Brooks and P. Maes, editors, Artificial Life IV, pages 190--197. MIT Press/Bradford Books, 1994.

[21] L. Pagliarini, H. H. Lund, O. Miglino, and D. Parisi. Artificial Life: A New Way to Build Educational and Therapeutic Games. In Proceedings of Artificial Life V. MIT Press/Bradford Books, 1996.

[22] R. Pfeifer and C. Scheier. Understanding Intelligence. MIT Press, Cambridge, MA, 1999.

[23] E. Sklar, J. Johnson, and H. H. Lund, "Children Learning From Team Robotics: RoboCup Junior 2000", Educational Research Report, Department of Design and Innovation, Faculty of Technology, The Open University, Milton Keynes, UK, 2000.

[24] V. Vucic and H. H. Lund. Self-evolving Arts --- Organisms versus Fetishes. In Muhely (The Hungarian Journal of Modern Art), 104, 69-79, 1997.

[25] S. Woodcock. Game AI: The state of the industry. Game Developer. Miller Freeman, August 1999.

[26] S.-Y. Yoon, B. M. Blumberg, and G. E. Schneider. Motivation Driven Learning for Interactive Synthetic Characters. In Proceedings of Fourth International Conference on Autonomous Agents. ACM Press, 2000.