Applications of a simplified protocol of RoboCup 2D Soccer Simulation

Norbert Bátfai, Roland Dóczi, János Komzsik, András Mamenyák, Csaba Székelyhídi, József Zákány, Márton Ispány and György Terdik

***Abstract*—The RoboCup 2D Soccer Simulation (RCSS) is a sophisticated soccer simulation environment introduced in [1] and implemeted in [2], [3]. For educational and other sport science purposes, we have simplified the protocol of the RCSS. With raw phrasing we would say that we have eliminated the main AI part of the AI-based RCSS simulation model in order that it can be used easily in sport science and in education of programming. In compliance with this aim, we must modify the RCSS environment. In our terminology, the suitable modified rcssserver is referred to as ”lighter RCSS” [4], [5]. In this paper, the experiences of our first ”lighter teams” will be presented.**

***Index Terms*—RoboCup 2D Soccer Simulation, rcssserver/sampleclient, Lighter RCSS, sport science soccer simulations, Football Avatar, education of programming.**

I. Introduction

The purpose of this paper is twofold. First, we are developing a simulation based decision making support expert system for professional soccer in the framework of an industrial project called Football Avatar. It involves the investigation of the existing soccer simulation environment of the sport science point of view. Second, whereas our part of the development is taking place in university environment, we need to prep students to participate in the development of the mentioned industrial project. The present paper can be seen as a step towards this twofold purpose.

A. Sport Science Purpose

The initiative for development of a sport science simulation based system came from the FerSML (Footballer and Football Simulation Markup Language) platform [6] introduced in the paper [7]. In the terminology of FerSML, the abstractions of soccer players, coaches and matches are called avatars. The avatars contain all information required to play such simulated soccer matches with probability properties corresponding to real matches of the abstracted players and coaches. In the special case of Football World Cups, an introductory statistical comparison of real and simulated matches can be found in [8]. The used simulation engine is based on our former mobile soccer game [9] and absolutely independent from the RoboCup (Robot World Cup) introduced in [1]. This is not surprising, because RoboCup focuses on AI, and it is purely AI in such sense that agents have to build up themselves, from scratch, the simulated world from their very limited sensory input. In contrast, the FerSML avatars are automatically fully aware of the following data: the exact position of the ball, players and parts of the pitch, the used tactics, etc. Accordingly, the building of this sort of knowledge would be a superfluous process for the purpose of an avatar based simulation, because avatars are apriori aware of this knowledge. In parallel, using the artificially limited sensory input may be waived for ”sport science” avatars that will be defined on a much higher abstraction level. However, the RCSS environment is a sophisticated collection of GPLed open source software components (like rcssserver, rcssmonitor or rcsslogplayer [2]). The simulation is made inside the rcssserver, the client agents are connected to it over UDP/IP. The rcssmonitor and the rcsslogplayer are visual software interfaces to the simulation. So it would also be useful to investigate the modification of the RCSS environment in order to serve our sport science or educational purposes easily. In summary, our motivation is to try to eliminate the artifical intelligence-specific limitations from the RCSS environment. The mathematician, Gregory Chaitin, wrote in his book [10], *”you understand something only if you can program it. (You, not someone else!) Otherwise you don't really understand it, you only think you understand it.”* In the spirit of this, although we are early in the investigation of soccer simulation models, we have modified the software of the RCSS environment. This paper is built on a modification of the rcssserver. In parallel, we worked on a modification of rcsslogplayer, where the Brillinger potential field [11]-based model are applied for RCSS matches [12].

B. Educational Purpose

Programming RCSS agents is a hard task because they have a very limited sensory input. For example, the following log snippet shows what can be seen by an agent at every 150 millisecond:

```
(see 0 ((f c b) 23.1 38 0 0) ((f r b) 54.6
-35) ((f p r b)  ...
```

where the letter f denotes the appropriate circular flag shown in Fig. 1. A classical RCSS agent must determine its position only from this information. But we are interested in building agents that know, for example, their own position on the pitch, rather than classical agents. In compliance with this, in the lecture notes [4], [5] we introduce a new client protocol command `(pos x y power)`, which will move the agent towards the (x, y) position of the pitch with the velocity

N. Bátfai is with the Department of Information Technology, University of Debrecen, H-4010 Debrecen PO Box 12, Hungary, e-mail: batfai.norbert@inf.unideb.hu.

R. Dóczi, J. Komzsik, A. Mamenyák, Cs. Székelyhídi, J. Zákány, M. Ispány and Gy. Terdik are with the University of Debrecen.

derived from the given power. And vice versa, if the command line option `server::light_response` is switched on, the ”lighter RCSS server” returns the x, y coords of an agent. Applying these and similar modifications makes it rather easy to use RCSS environment to investigate sport science questions or to use it in education of programming.

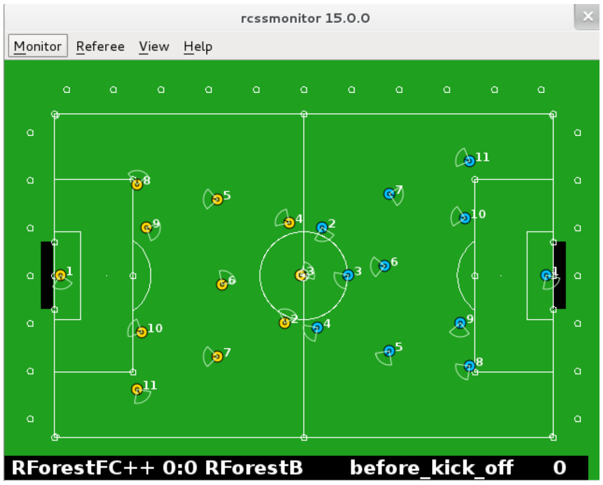

Fig. 1. The RCSS field of play in the program rcssmonitor.

C. Comparison with Existing Works

It is important to note that we have no aim to create classical RCSS 2D robocup agents. Our key motivation is to consider a relaxation of the agents’ very limited sensory input together with the RCSS protocol, because the limitations of RCSS environment are unnecessary from the point of view of sport science. The sub-objective of our research is to have a better understanding of server operations. On the one hand, if we begin from scratch, the development time of such robocup team that can apply tactics (in the sense of the sport science) is many years. More recently, a good example of a multiannual development is the Romanian team OXSY [13]. On the other hand, we may customize a ready-made agent by using the GPLed Agent2D sources [14], for example, as some successful teams (e.g. EdInferno.2D [15]) have already did. But in our case, it would raise licensing questions and concerns for the mentioned industry project.

II. The Simplification of the RCSS and the First ”Lighter Teams”

In this section, we present the first teams which already use the ”lighter RCSS” protocol. The Debrecen Round Forest FC++ team is one of the many educational teams of the book [4], [5]. It introduces the usage of the ”lighter RCSS” environment. The other presented teams are developed by students. These teams are based on the educational teams. This kind of student teams can be regarded successful if it can confidently win against the educational teams. The development of the teams, titled ”Kő papír metál FC”, ”Rozsdás FC” and ”Deadly Team”, satisfy that condition.

In particular, the ”lighter teams” can be tested easily because these teams are packaged together with the ”lighter RCSS” server. In addition, we will emphasize some video links where the teams in question can be seen during the matches, as well as associated RCG (record game) files are available for all teams. Based on these classical RCG files the matches are reproduced exactly by rcsslogplayer [2].

A. Debrecen Round Forest FC++

The Debrecen Round Forest FC++ team is the last example of the lecture notes [4], [5]. Here we should remark that [4] is a Hungarian language book and it is ready to be published, but the English translation [5] is only in work in progress state at this moment. But of course, the package of the modified (i.e. the lighter) RCSS server is available for download from http://www.inf.unideb.hu/~nbatfai/rcssserver-15.1.0.light5.rf.tar.bz2. This package is based on the classical RCSS distribution and it contains some modifications to implement the new positioning command and contains a few other files to implement the behavior of a lighter sample team called Debrecen Round Forest FC++. The exactly detailed description of changes can be found in the file NEWS in the distribution of the package. This team is gradually developed by simpler examples like ”Bolyongó SE”, ”Bolyongó FC++”, ”Debreceni Lobogó FC++”, ”Debreceni Egyetértés FC++”, ”Debreceni Hivatásos FC++”, ”**Light FC++**”, ”Debrecen Great Forest FC++” and ”Debrecen Deep Forest FC++”. In that order these teams give a programmed introduction to building a C++ based RoboCup team. All teams are based on rcssserver (15.1.0) [2] and its sample client [3]. They are simple threaded versions of client.cpp of [2], [3], but all of them have their own lexer part to analyse sensory data. The ”Bolyongó SE” starts from scratch to control the behavior of agents. The Light FC++ is the first lighter team. The mentioned teams can be downloaded from the links contained in the lecture notes [4], [5] .

The Debrecen Round Forest FC++ and the ”lighter RCSS” server can be found directly at URL http://www.inf.unideb.hu/~nbatfai/rcssserver-15.1.0.light5.rf.tar.bz2

B. The simplified server

We have modified some source files in the original rcssserver [2] to implement the simplification. (A description of the modifications can be found in the file NEWS in the distribution of the package at URL http://www.inf.unideb.hu/~nbatfai/rcssserver-15.1.0.light5.rf.tar.bz2.) As a result the rcssserver can be started with the following options:

- If the optional command line option `server::light_response` is set to true, the x and y coordinates of the client agent will be inserted into the server’s response. (To be more precise, into the client sensor response see [3].)
- If the optional command line option `server::light_response_with_angle` is set to true, the x and y coordinates and the body angle of the client agent will be inserted into the server’s response.
- If the optional command line option `server::light_response_with_angles` is set to true, the x and y coordinates and the body and neck angles of the client agent will be inserted into the server’s response.

- In all the three cases, the server can interpret the newly introduced client control command `(pos X Y power)`.

The implemented command `pos` has the same uses as the original client control commands. And of course, the simplified server can work with original teams transparently without any change.

C. Deadly Team

Our team is developed by András Mamenyák and János Komzsik. The team is based on the Debreceni Hivatásos FC++ team. The strenght of the team comes from the ability to pass the ball from one agent to the other. To perform a pass, we need the coordinates of the players. The coordinates can be calculated easily if we know the position of our agent and the distance to the flags. These are recieved from the server automatically, so the only thing needed is the angle the agent is standing. We calculate this as seen on Fig. 2.

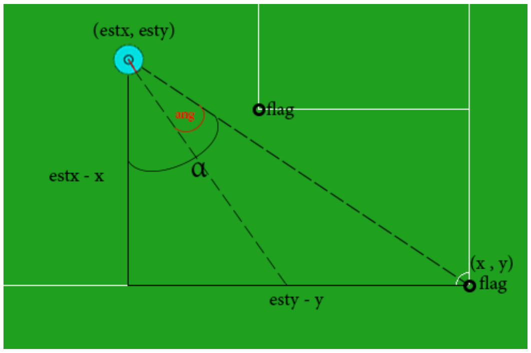

Fig. 2. The calculation of the body angle.

After this we rely on this universal passing method:

```
/*
* pass the ball to the given x, y coordinates
*/
void deadlyPassXY(float x, float y) {

    float dist = std::sqrt(
                   std::pow(dl.estx - x, 2) +
                   std::pow(dl.esty - y, 2)
                 );

    float power = dist * 5.0 / 2.0;

    float angle = std::atan((dl.esty - y)/(dl.estx - x));
    angle *= 180.0 / pi;
    angle -= dl.esta;

    if (x < dl.estx)
        angle += 180.0;
    if (angle > 180.0)
        angle -= 360.0;

    char buf[64];
    std::snprintf(buf, 64, "(kick␣%f␣%f)", power, angle);
    sndCmd(buf);
}
```

We now need to decide to which players to pass to, and to which players not to pass to. A teammate not receiving a pass is based on two things: there is an opponent player near our teammate or there is an opponent player close to the the path of the pass. If there is no one to pass to, then the player can dash with the ball if there are no opponent players near. If an opponent player is near, our player will try to go past him. If there are too many opponents near, then our player will turn around, and look for other teammates to pass to.

Meanwhile the other players seek their predefined attacking or defensive positions if not involved in a pass. If the opposite team is attacking, the players try to mark the attackers or tackle if close enough. To determine the possession of the ball, we utilize the possibility to communicate, so our players tell each other which tactic to use.

The main weakness of our team is simply stamina. The players get exhausted quickly because they run too much, but sometimes they are running a bit slowly, because they are constantly adjusting their positions to the ball.

The developement of the presented team took more than two months, but it is continuously improved and future developments are planned.

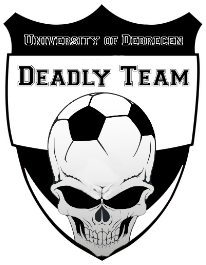

Fig. 3. The logo of the Deadly Team.

D. Tortoise Formation FC

As we played matches against the famous HELIOS (more precisely, the HELIOS_base [14]), we discovered its weakness, so this resulted in creating a team, based on the core of the Deadly Team. This team is called Tortoise Formation FC, because it uses a tactic in which the players sorround the ball, creating an inpenetrable fortification for the HELIOS players. This way, the players can simply march with the ball, and score, if they are close enough to the goal. The success of the Tortoise Formation FC demonstrates that, using the positioning command of the ”lighter RCSS”, the teams’ approaches to the game may be formed very easily. But it has no sport scientific importance, as shown in the video at http://youtu.be/5ng5W_2Rofs, and in Figure 4.

The teams can be downloaded at the URL http://robocup.inf.unideb.hu/~andras/RoboCup/Teams/.

E. KőPapírMetál FC

The first KőPapírMetál FC was developed by Roland Dóczi and Tamás Józsi. This team was an Atan [16] (Java language) based team and it was developed to take place in two local RoboCup events. After this, one of the teammates left the team and it is rewritten in C++ as a DForest based team, which supports the simplified protocol of RoboCup2D Soccer Simulation. The advantages of the new protocol are used by the team. With this, the team has coordinate geometry based methods.

When the ball is not near, the agents try to stay in their correct positions. When it is near the goal, the goalkeeper

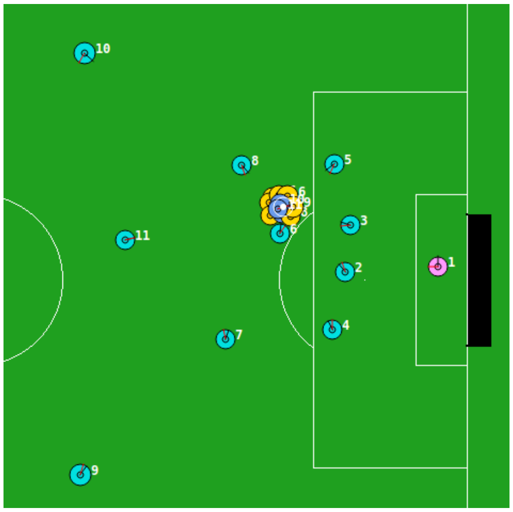

Fig. 4. The Tortoise Formation FC's approach to the game.

rushes for the ball and tries to catch it. The defenders stand before the attackers to block them getting the ball. After this, the defenders pass the ball to the furthest attacker. The midfielders do the same. When an attacker agent gets the ball, it rushes in the direction of the goal, and when the agent is close enough, it tries to score. The target is the furthest goalpost from the goalkeeper.

The main advantage of the team lies in the attacking strategy, but it has its disadvantages too. The goalkeeper can not analyze the situations as fast as it should and because of this, we concede more goals.

Fig. 5. The logo of the KőPapírMetál FC.

The team can be downloaded at the URL http://shrek.unideb.hu/~roland10/RoboCup2D/team/

F. Rozsdás FC

The Rozsdás FC is developed by József Zákány. The team was born in a Java environment, based on Atan [16] framework. Thanks to the simplified protocol of RoboCup 2D Soccer Simulation, it could reborn as a C++ team, with the adventages of the new protocol. It is based on the Debrecen Great Forest FC++. The development lasted nearly a month. In this work, the individual behaviour of players was focused. For example, there is an imaginary circle around the middle goal flag. If the goalkeeper crosses the (red) circle line, he goes back to his place, in front of the goal. This can be calculated easily from the coordinates of the goalkeeper, and the equation of the circle.

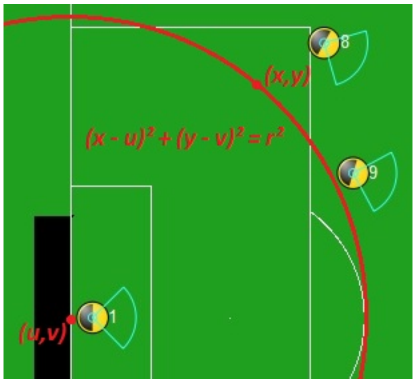

Fig. 6. Positioning of the goalie.

The team is basically built on defensive tactics. The defensive players stay on their own half of the field, they do not take part in the attack at all. They try to keep the ball away from the goal. If one of them gets the ball, he tries to pass it to a teammate, or just kick it into the direction of the opponent goal. The midfielders are playing similar like the defensive players, but they go forward and they help with the attack. And finally, the attackers wait for the ball on the opponent side. They pass the ball between each other, and if they have the chance they try to score.

The team can be downloaded at the URL http://shrek.unideb.hu/~joe42/RoboCup2D/rcssserver-15.1.0.light4.gf.tar.bz2.

G. NimFC

The NimFC is based on the Debrecen Round Forest FC++. The work was much easier with the lighter server. The NimFC is functionally divided into defenders, midfielders, strikers and goalkeeper. The methods of passing and positioning and the kicking habits of strikers were changed compared to the original RForest FC++.

The team can be downloaded at the URL http://web.unideb.hu/~szekelyhidi/rcssserver-15.1.0.light5.rf.tar.bz2.

III. Conclusion

How efficient and effective are the developed teams? In this environment, it is essential to survey the matches. For example, the playing style of our teams can be seen in the following YouTube videos or by viewing the standard rcg (record game) files of the next subsections.

DeadlyTeam:
http://www.youtube.com/watch?v=DsDnNwj78pw
http://www.youtube.com/watch?v=NZeP8BoOQxQ
Tortoise Formation FC:
http://www.youtube.com/watch?v=5ng5W_2Rofs

A. Results

The match results against each other can be found in the Table I. The Table II shows our match results against the world's best teams. In this latter table, the HELIOSbase [14] is a

TABLE I
RESULTS AGAINST EACH OTHER

Match	Result	
GreatForest vs DeadlyTeam	0	9
GreatForest vs DeadlyTeam	2	12
DeadlyTeam vs GreatForest	11	3
DForest vs DeadlyTeam	1	9
DForest vs DeadlyTeam	1	8
RForestFC++ vs DeadlyTeam	3	8
RForestFC++ vs DeadlyTeam	0	12
DeadlyTeam vs RForestFC++	12	2
DeadlyTeam vs RForestFC++	16	1
KoPapirMetal vs DeadlyTeam	2	5
KoPapirMetal vs DeadlyTeam	3	7
DeadlyTeam vs KoPapirMetal	14	3
RozsdasFC vs DeadlyTeam	1	4
NimFC vs DeadlyTeam	1	12
KoPapirMetal vs RozsdasFC	1	7

TABLE II
RESULTS AGAINST THE WORLD'S BEST TEAMS

Match	Lighter	Result	
HELIOS_base vs TortoiseFormation	Enabled	5	0
TortoiseFormation vs HELIOS_base	Enabled	2	12
KoPapirMetal vs HELIOS2010	Enabled	0	48
KoPapirMetal vs HELIOS2010	Enabled	0	46
RozsdasFC vs HELIOS2010	Enabled	0	35
HELIOS2010 vs RozsdasFC	Enabled	17	0
RozsdasFC vs HELIOS2010	Enabled	0	34
HELIOS2010 vs RozsdasFC	Enabled	21	0
HELIOS_base vs DeadlyTeam	Enabled	31	0
DeadlyTeam vs WrightEagle	Enabled	0	42
DeadlyTeam vs WrightEagle	Disabled	0	48
RForestFC++ vs WrightEagle	Disabled	0	50
DeadlyTeam vs Dainamite	Disabled	0	23

base team of the famous HELIOS2010 [17] and HELIOS2011 [18], which can be found at URL http://sourceforge.jp/projects/rctools/downloads/55186/agent2d-3.1.1.tar.gz/. The WrightEagle [19], the HELIOS2011 and the Dainamite [20] participated in the Competition RoboCup2011 Soccer Simulation 2D in Istanbul, Turkey, where the Chinese WrightEagle team won this championship, the Japanese team HELIOS2011 reached the second place and the Dainamite team from Germany catched the 14-th place. The WrightEagle, HELIOS2011 and Dainamite teams can be downloaded at URL http://www.socsim.robocup.org/files/2D/binary/RoboCup2011/.

All rcg files of the matches of aforementioned tables can be downloaded at the URL http://robocup.inf.unideb.hu/~andras/RoboCup/RCG/.

B. Teaching Experiences

We have been using (Atan [16]) Java and (rcssserver [2]/sample client [3]) C++ based RCSS client agents [4], [5], [21] in the education of programming for two years in Software Information Technology and Engineering Information Technology BSc regular courses at the University of Debrecen. The use of the ”lighter RCSS” has started this year and we have very good experience with it. The lighter student teams are proved better than the last year's teams.

The development of ”lighter teams” requires around 1-3 months. Considering that the developers was first year BSc students, it is a short period of time. But we have no sufficient experience for making a precise global comparison, because we build RoboCup teams only in the context of the investigation of soccer simulations.

C. Summary and Further Work

Regarding the mentioned industrial simulation project, we have no plans to use the classical nor the ”lighter RCSS” environment. Partially based on the experience of this work, we can decide that the simulator of Football Avatar will be developed from scratch. In the future, we plan to develop a football avatar to ”lighter RCSS” agent conversion software. In addition, it is notable that we are using RCSS data to test mathematical models of sport science [12].

ACKNOWLEDGMENT

The publication was supported by the TÁMOP-4.2.2.C-11/1/KONV-2012-0001 project. The project has been supported by the European Union, co-financed by the European Social Fund.

REFERENCES

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, “Robocup: The robot world cup initiative,” in *Proceedings of the first international conference on Autonomous agents*, ser. AGENTS '97. ACM, 1997, pp. 340–347. [Online]. Available: http://doi.acm.org/10.1145/267658.267738

[2] The RoboCup Soccer Simulator Maintenance Committee, “The robocup soccer simulator,” 2001-2012. [Online]. Available: http://sourceforge.net/projects/sserver/

[3] M. Chen, K. Dorer, E. Foroughi, F. Heintz, Z. Huang, S. Kapetanakis, K. Kostiadis, J. Kummeneje, J. Murray, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang, and X. Yin, “Users manual robocup soccer server for soccer server version 7.07 and later,” 2003. [Online]. Available: https://sourceforge.net/projects/sserver/files/rcssmanual/

[4] N. Bátfai, *Programozó Páternoszter újratöltve: C, C++, Java, Python és AspectJ esettanulmányok*. author's edition, 2012. [Online]. Available: http://www.inf.unideb.hu/~nbatfai/konyvek/PROP/prop.book.xml.pdf

[5] ——, *Paternoster of Programmers Reloaded: C, C++, Java, Python and AspectJ Case Studies*. author's edition, 2012. [Online]. Available: http://www.inf.unideb.hu/~nbatfai/konyvek/POPR/popr.book.xml.pdf

[6] ——, “Football(er) simulation markup language,” 2010-2012. [Online]. Available: http://sourceforge.net/projects/footballerml/

[7] ——, “Footballer and football simulation markup language and related simulation software development,” *Journal of Computer Science and Control Systems*, vol. 3, no. 1, pp. 13–18, 2010.

[8] ——, “The socceral force,” *CoRR*, vol. abs/1004.2003, 2010.

[9] N. Bátfai, E. Bátfai, and I. Pšenáková, “Jávácska one: Open source mobile games to revolutionize education of programming,” *Journal of Computer Science and Control Systems*, vol. 3, no. 2, pp. 5–10, 2010.

[10] G. J. Chaitin, *Meta Math! The Quest for Omega*, 2004. [Online]. Available: http://arxiv.org/abs/math/0404335

[11] D. R. Brillinger, “A potential function approach to the flow of play in soccer,” *Journal of Quantitative Analysis in Sports*, vol. 3, no. 1, 2007. [Online]. Available: http://www.stat.berkeley.edu/~brill/Papers/jqas.pdf

[12] N. Bátfai and Gy. Terdik, "The application of the data of RoboCup 2D Soccer Simulation League to test several sport science results (submitted manuscript)," 2012. [Online]. Available: http://robocup.inf.unideb.hu/fersml/assr/

[13] S. Marian, D. Luca, B. Sarac, and O. Cotarlea, "Oxsy 2010 team description," 2010. [Online]. Available: http://www.socsim.robocup.org/files/2D/tdp/RoboCup2010/2D_TDP_Oxsy.pdf

[14] H. Akiyama and H. Shimora, "Helios_base - agent2d-3.1.1," 2012. [Online]. Available: http://en.sourceforge.jp/projects/rctools/releases/?package_id=4887

[15] M. Hawasly and S. Ramamoorthy, "Team description paper for robocup 2011 2d soccer simulation league," 2011. [Online]. Available: http://wcms.inf.ed.ac.uk/ipab/robocup/research/TDP-Edinferno2D.pdf

[16] N. James and D. Wood, "Atan - robocup 2d soccer server java interface," 2012. [Online]. Available: https://github.com/robocup-atan/atan/

[17] H. Akiyama and H. Shimora, "Helios2010 team description," 2010. [Online]. Available: http://www.socsim.robocup.org/files/2D/tdp/RoboCup2010/2D_TDP_HELIOS.pdf

[18] H. Akiyama, H. Shimora, T. Nakashima, Y. Narimoto, and T. Okayama, "Helios2011 team description," 2011. [Online]. Available: http://www.socsim.robocup.org/files/2D/tdp/

[19] A. Bai, G. Lu, H. Zhang, and X. Chen, "Wrighteagle 2d soccer simulation team description 2011," 2011. [Online]. Available: http://ai.ustc.edu.cn/en/robocup/2D/tdps/WrightEagle2011_2D_Soccer_Simulation_Team_Description_Paper.pdf

[20] H. Endert, T. Karbe, J. Krahmann, F. Trollmann, and N. Kuhnen, "The dainamite 2008 team description," 2008. [Online]. Available: http://www.socsim.robocup.org/files/2D/tdp/RoboCup2008/2D_TDP_DAInamite.pdf

[21] N. Bátfai, *Mesterséges intelligencia a gyakorlatban: bevezetés a robotfoci programozásba*. author's edition, 2012. [Online]. Available: http://www.inf.unideb.hu/∼nbatfai/konyvek/MIRC/mirc.book.xml.pdf

Norbert Bátfai is working as an assistant professor in Faculty of Informatics at the University of Debrecen, Hungary. He received his M.Sc. (summa cum laude) in Computer Science at 1998 from the Kossuth Lajos University (KLTE), Debrecen, Hungary. In 1999, he won the first prize in the Java Programming Contest organized by Hungarian Java Alliance: Sun, IBM, Oracle, Novell and IQSoft. In 2004, his company won the first prize in the Hungarian Mobile Java Developer Contest organized by Sun Hungary and Nokia Hungary. In 2008, the Hungarian Chief Information Officers' Association has selected him as an IT trainer of the year. He received the Ph.D degree in 2011. He won the Pollák-Virág award from Scientific Association for Infocommunications Hungary in 2012.

András Mamenyák is mayor in Engineering Information Technology BSc at the University of Debrecen, Hungary. He is a part of the World Football - Modelling and Visualizing Research Group and the Football Avatar Project at the University of Debrecen. He took part in the Romanian national physics competition final and in the Vermes Miklós International Physics competition final. He also took part several times in the Hungarian Gordius national mathematics competition final.

János Komzsik is mayor in Engineering Information Technology BSc at the University of Debrecen, Hungary . He is a member of the World Football - Modelling and Visualizing Research Group at the University of Debrecen. He is also a member of the Football Avatar Project.

József Zákány is mayor in Engineering Information Techonolgy BSc at University of Debrecen, Hungary. He is a part of the World Football - Modelling and Visualizing Research Group and the Football Avatar Project at the University of Debrecen.

Csaba Székelyhídi is mayor in Software Information Technology, BSc at the University of Debrecen, Hungary. He is a member of World Football Modelling and Visualizing Research Group. In 2011 he graduated from Mihály Fazekas High School of Debrecen, Hungary.

Roland Dóczi is mayor in Engineering Information Techonolgy BSc at University of Debrecen, Hungary. He is a part of the World Football - Modelling and Visualizing Research Group and the Football Avatar Project at the University of Debrecen.

Márton Ispány received his Ph.D. degree from the Kossuth Lajos University, Hungary. He is an associate professor at the Department of Information Technology at the University of Debrecen, Hungary. Dr. Ispány's research areas include integer valued time series analysis, branching processes, and data mining.

György Terdik received his Ph.D. degree from the Kossuth Lajos University, Hungary. He is a member of the editorial board of the quarterly Publicationes Mathematicae Debrecen. He is a professor and Head of the Department of Information Technology at the University of Debrecen, Hungary. Dr. Terdik's research areas include nonlinear, non-Gaussian time series analysis, Lévy processes, and high speed network modeling.