AI Magazine Volume 19 Number 3 (1998) (© AAAI)

RoboCup-97

The First Robot World Cup Soccer Games and Conferences

Itsuki Noda, Shóji Suzuki, Hitoshi Matsubara, Minoru Asada, and Hiroaki Kitano

■ RoboCup-97, The First Robot World Cup Soccer Games and Conferences, was held at the Fifteenth International Joint Conference on Artificial Intelligence. There were two leagues: (1) real robot and (2) simulation. Ten teams participated in the real-robot league and 29 teams in the simulation league. Over 150 researchers attended the technical workshop. The world champions are CMUNITED (Carnegie Mellon University) for the small-size league, DREAMTEAM (University of Southern California) and TRACKIES (Osaka University, Japan) for the middle-size league, and AT-HUMBOLDT (Humboldt University) for the simulation league. The Scientific Challenge Award was given to Sean Luke (University of Maryland) for his genetic programming–based simulation team LUKE, and the Engineering Challenge Awards were given to UTTORI UNITED (Utsunomiya University, Toyo University, and Riken, Japan) and RMIT (Royal Melbourne Institute of Technology, Australia) for designing novel omnidirectional driving mechanisms. Over 5000 spectators and 70 international media covered the competition worldwide. RoboCup-98, the Second Robot World Cup Soccer, was held in conjunction with the Third International Conference on Multiagent Systems in Paris, France, in July 1998.

RoboCup is an attempt to promote AI and robotics research by providing a common task, soccer, for evaluation of various theories, algorithms, and agent architectures (Kitano, Asada, et al. 1997). RoboCup-97, the First Robot World Cup Soccer Games and Conferences, was held on 22–28 August 1997 at the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97) (figure 1). It was organized by RoboCup Japanese National Committee and Nihon Keizai Shinbun Inc., and it was sponsored by Namco Limited, Sony Corporation, Nihon Sun Microsystems K.K., and Itochu Techno-Science Corporation. Over 5000 people watched the games, and over 100 international media (such as CNN, ABC, Le Monde, Le Figaro, Der Spigel, The Australian, NHK, and Sky Channels), as well as prominent scientific magazines such as *Science*, covered them. The First RoboCup Workshop was also held (Kitano 1997). This article reports on RoboCup-97.

RoboCup-97 had two leagues: (1) the real-robot league and (2) the simulation league. Aside from the world championship awards, RoboCup created the RoboCup Scientific Challenge Award and the Engineering Challenge Award to be equally prestigious. Detailed information about RoboCup is given at www.robocup.org/RoboCup. In this issue of *AI Magazine,* winners of each league contributed an article describing scientific aspects of their teams.

Real-Robot League

The Real-Robot League, which uses physical robots to play soccer games, consists of several categories. At RoboCup-97, there were two categories for game competition and one for skill competition.

The *small-size league* is a team consisting of five robots and plays on a field that is equivalent to one Ping-Pong table. Each robot is about 15 centimeters (cm) in diameter, or under 180 cm^2, and the maximum length must be less than 18 cm. An orange golf ball is used.

In the *middle-size league,* there are five robots to a team, and each robot must be less than 50 cm in diameter, or 2,000 cm^2. A Federation Internationale de Football Association (FIFA) size-4 Futsal ball is used. The field of play is equivalent to 9 Ping-Pong tables (3 tables by 3 tables).

The *expert-robot league* is for competition

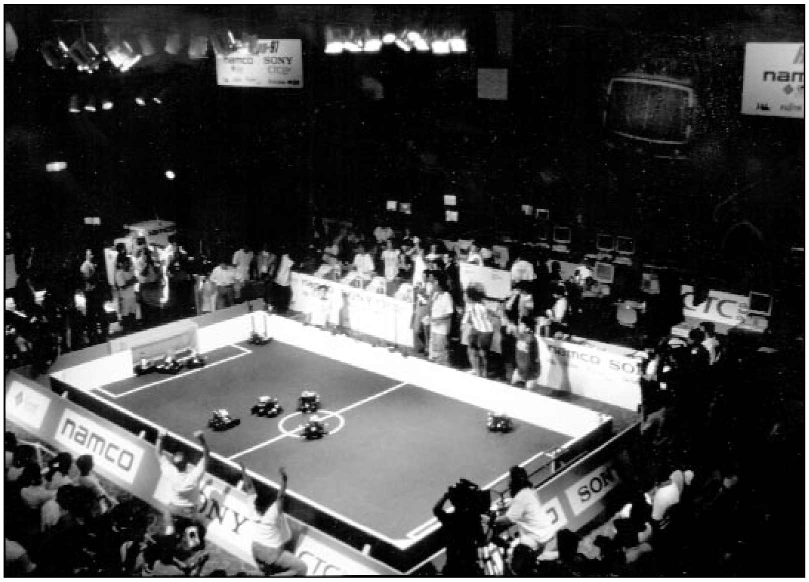


Figure 1. RoboCup-97.

Figure 2. A Match of the Small-Robot League.

between robots having special skills. These skills concentrate on isolated aspects of the game of soccer.

For the real-robot league at RoboCup-97, about 10 teams throughout the world participated in the competition: 4 teams (CMUNITED [Carnegie Mellon University (CMU)], MICROB [Paris-VI, France], ROGI-II [University of Girona, Spain], and NAIST [Nara Advanced Institute of Science and Technology (NAIST), Japan]) in the small-size robot league, 5 teams (DREAMTEAM [Information Sciences Institute–University of Southern California (USC-ISI)], TRACKIES [Osaka University, Japan], ULLANTA [Ullanta Performance Robotics, Royal Melbourne Institute of Technology (RMIT), Australia], and UTTORI UNITED [a joint team of Riken, Toyo University, and Utsunomiya University, Japan]) in the middle-size league, and 2 teams (RMIT [Australia] and Colorado School of Mines) in the expert-robot league.

The Small-Size Robot League

For the small-size robot league, the use of a global vision system is permitted, which enables the team to plot absolute position information for each robot and the ball.

Figure 2 shows a game setup of the small-size league. This league is played on a field equivalent to a Ping-Pong table, which is 152.5 cm by 274.0 cm. This size was selected because a Ping-Pong table is a low-cost standardized material that can be purchased throughout the world. We initially defined a field as 1/100 of the FIFA world cup field, which is 120 meters (m) by 90 m, but researchers would have had to build everything from scratch. Considering the amount of work that has to be done in building robots, field construction really presents no major effort, but it seemed important that necessary materials be widely available. Availability of low-cost materials around the world is particularly important because RoboCup is widely used for educational purposes.

For RoboCup-97, three teams (CMUNITED, ROGI-II, MICROB) used global vision and differential drive (robot moved by two wheels, and robot's direction changed by difference in rotation speed of two wheels), and NAIST used an on-board camera, two driving wheels, and one steering wheel. These two approaches are on the two extreme ends of the design spectrum. In the future, we expect teams that use a hybrid approach with global vision or distributed vision and an on-board camera and other sensor systems. The small-size league is particularly suited for experiments on multiagent autonomous systems in a sensor-rich environment.

Global vision is permitted in this league for two reasons: First, it is rather difficult for this size robot to have an on-board camera system. If it did, it would inevitably be expensive so that many underfunded research groups would not be able to participate in the initiative. Second, the use of global vision allows us to investigate issues of distributed vision systems expected to cooperate with each other for video surveillance, monitoring, and guidance.

Actually, three of the four RoboCup-97 teams adopted the global vision system, and only one team, NAIST, adopted an on-board vision system for each of its two robots. External views of robots from each team are shown

Versus	Team 1	Team 2	Team 3	Team 4	Points	Goal Diff.	Goals Scored	Rank
Team 1, MICROB	—	0–0	1–3	0–0	2	–2	1	3
Team 2, ROGI-II	0–0	—	0–2	0–1	1	–3	0	4
Team 3, CMUNITED	3–1	2–0	—	5–0	6	9	10	1
Team 4, NAIST	0–0	1–0	0–5	—	3	–4	1	2

Table 1. Result of the Small-Size League Preliminary.

MICROB: Universite Paris VI, France; ROGI-II: University of Girona, Spain; CMUNITED: Carnegie Mellon University; NAIST: Nara Institute of Science and Technology, Japan.

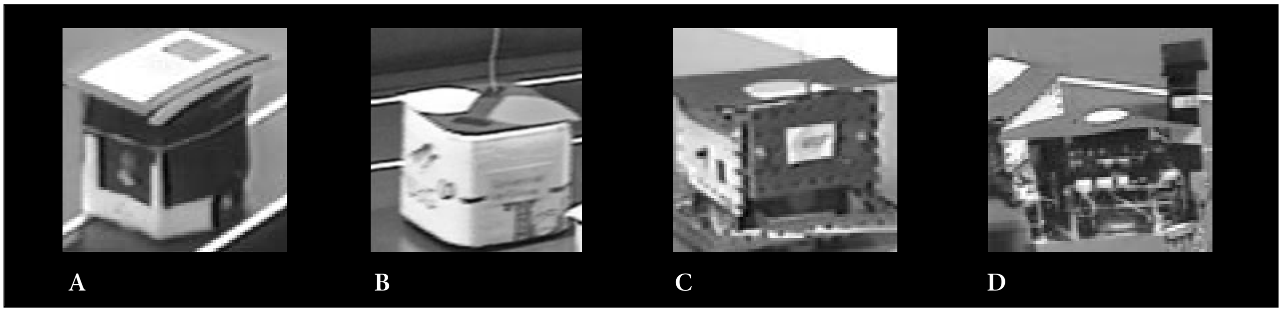


Figure 3. Participating Robots. A. MICROB. B. ROGI-II. C. CMUNITED. D. NAIST.

in figure 3. Six games were scheduled for the preliminary rounds so that every team could have games with all other teams. However, because of a conflict in radio frequency for robot control, there was no game between MICROB and ROGI-II. Table 1 shows the results of the preliminary games for the small-size league.

According to the ranks in the preliminary round, CMUNITED and NAIST went to the finals. The Small-Size League World Championship was awarded to CMUNITED from Carnegie Mellon University, which won over NAIST by 3–0.

The Middle-Size League

The middle-size league is played on a field equivalent to nine Ping-Pong tables (3 tables by 3 tables), which is 4.575 m by 8.220 m. In this league, a FIFA size-4 Futsal ball is used. Figure 4 shows the game setup for the middle-size league. This year, we had five teams in the middle-size league, each of which has its own features in several aspects:

Figure 4. A Match of the Middle-Size Robot League.

First was TRACKIES from Osaka University, Japan (Asada Lab.). Its features include remote brain systems, nonholonomic vehicles, four attackers, and one goalee with an omnidirectional vision system. It was able to learn techniques to obtain basic skills, such as shooting and avoiding.

Second was RMIT RAIDERS from RMIT, The Department of Computer Systems Engineering, Research Center, Australia. Its features include a special mechanical design for omnidirectional motion of the round-shape robots and a global vision system that was used to control the four robots.

Third was the SPIRIT OF BOLIVIA from Ullanta Performance Robotics. Three robot platforms provided by Real World Interface Corp. had vision, sonar, and bumper sensors.

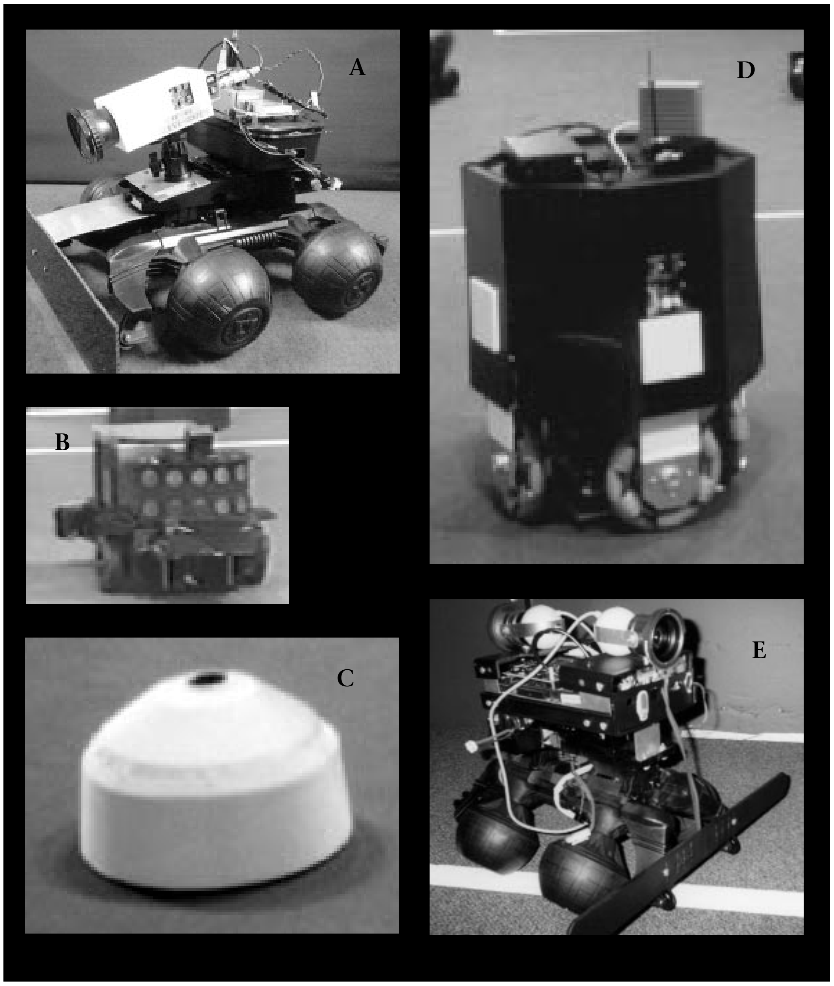


Figure 5. Participating Robots.

A. TRACKIES. B. SPIRIT OF BOLIVIA. C. RAIDERS. D. UTTORI UNITED. E. DREAMTEAM.

Fourth was UTTORI UNITED from Utsunomiya University, Toyo University, and the Institute of Physical and Chemical Research (Riken), Japan. The special mechanical design of omnidirectional motions was different from the RMIT team. Explicit communication was through an infrared, completely on-board system.

Fifth was DREAMTEAM from USC-ISI. A completely on-board system, DREAMTEAM used the same body as TRACKIES.

External views of attacker robots from each team are shown in figure 5.

Figure 6 shows visual images from four of the five TRACKIES robots. It should be noted that one of the robots uses an omnivision camera system that can view 360 degrees. Typically, a goalee robot is equipped with such a camera system because it must respond quickly to opponents' shots from any direction. This setup focuses on a situation where all robots have their own sensing systems and are not backed up by any global sensing systems.

Five games were held in the preliminary rounds (table 2), and the middle-size league final resulted in a draw (0–0) between the DREAMTEAM and TRACKIES. Their score in the preliminary round was also a 2–2 draw. The committee decided, in this case, to award the world championship to both teams.

The Engineering Challenge Award was given to both UTTORI UNITED and the RMIT RAIDERS for designing novel omnidirectional driving mechanisms (figure 7). These teams designed new robot-driving mechanisms that used special wheels (UTTORI UNITED) and balls (RMIT RAIDERS) to enable their respective robots to move to any direction without rotation. Such mechanisms significantly improve a robot's maneuverability, and their potential impact is far reaching.

The Expert-Robot League

Two teams entered the Expert-Robot League: WOLF, Colorado School of Mines, and RAIDERS, RMIT. WOLF used a robot platform by Nomadic Corp. with a stereo vision system and tried to chase a red ball. However, the vision system did not work well; therefore, WOLF did not make it. RAIDERS demonstrated its omnidirectional motion mechanism that was specially designed for RoboCup-97.

The committee decided that no prize was to be awarded in the expert league because neither team made its expected achievements.

Future Issues

Because RoboCup-97 was the first event of the RoboCup initiative, there were many new issues to face. One of the most serious problems was the radio link between robot bodies and their brains. All four teams in the small-size league, and two more teams in the middle-size league, adopted the remote brain system, and they suffered from serious radio noise in the competition site. TRACKIES suffered from noise in both transmitting a video image to the host computer and sending a motion command through the radio control unit. Therefore, the possibility of normal communication for the five robots was low. The environmental setup to prevent such a problem should be realized at future RoboCups.

The second problem was rules during a game. Many robots frequently lost the red ball because of radio noise or some problem in image processing, and often the game was stacked. Therefore, the judge changed the ball position according to a rule fixed at the competition site by all team representatives and restarted the game. However, the positions of the ball made it easy for some robots and more

Versus	Team 1	Team 2	Team 3	Team 4	Team 5	Points	Goal Diff.	Goals Scored	Rank
Team 1, TRACKIES	—	1–0	*	*	2–2	3	1	3	2
Team 2, RAIDERS	0–1	—	0–0	*	*	1	–1	0	4
Team 3, ULLANTA	*	0–0	—	0–0	*	2	0	0	3
Team 4, UTTORI UNITED	*	*	0–0	—	0–4	1	–4	0	5
Team 5, DREAMTEAM	2–2	*	*	4–0	—	3	4	6	1

* = games that were not held because of time restrictions.

Table 2. Results of the Middle-Size League Preliminary.

difficult for others to make a goal. Some systematic way should be developed to avoid such trivial problems.

Simulation League

The Simulation League is a tournament in which teams of 11 software agents play soccer in a virtual field. Participants can use any kind of programming language to build the agents. The only restriction is that they never use central control mechanisms to control a team of agents.

Rules

In the simulation league, each team must build 1 to 11 player programs. Of course, they can use the same program for all players. Each player program connects with SOCCER SERVER as a client using UDP-IP,[1] which simulates movements of objects (players and ball) on the soccer field (figure 8). The player program controls only one player. It receives visual and verbal sensor information from the server and sends control commands (turn, dash, kick, and say) to the server. Sensor information tells only partial situations of the field from the player's viewpoint, so the player program makes decisions using this partial and incomplete information. Only one command is accepted in one simulation step, so that the player program selects the best action at the moment. Limited verbal communication is also available that the players can use to communicate with each other to decide on team strategy. SOCCER SERVER 3.28 was the official simulator.

The programs can be written using any kind of programming language. The restriction is to not share any information among players except by verbal communication by way of the server. For example, player programs never use shared memory or direct communication. For more detail, please refer to the descriptions of the regulations on the SOCCER SERVER home page at //ci.etl.go.jp/~noda/soccer/server or www.robocup.org.

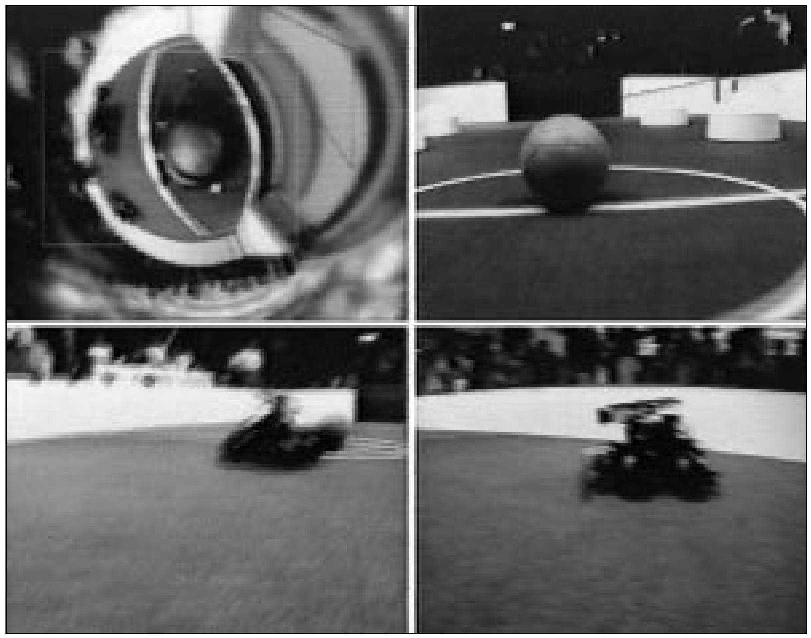


Figure 6. Visual Images from Four Robots in the Field.

Participants

Twenty-nine teams from 10 countries participated in the simulation league. Table 3 lists the teams.

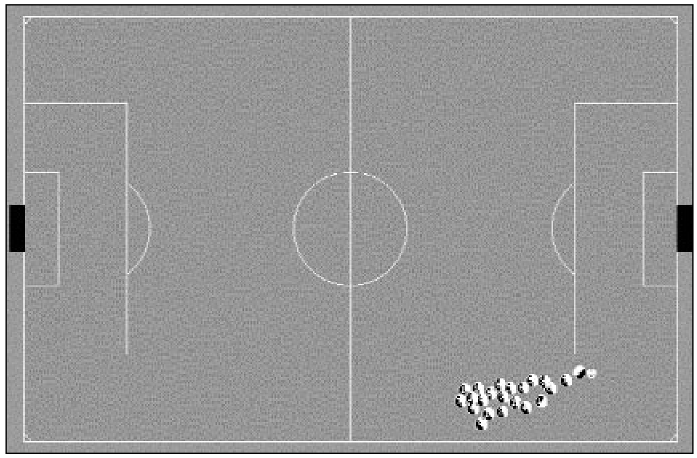
The competition was carried out in two stages: In the first stage, 29 teams were divided into 8 groups. Each group consisted of 3 or 4 teams. In a group, a team had a match with every other team in the same group (round-robin system). The first and second places in each group qualified to go to the second stage. The second stage was a single-elimination system of 16 teams.

We had three days for test runs before the formal competition. During these days, every team did final tuneups and had test matches with each other. With these test matches, participants exchanged information about their teams and discussed their technology.

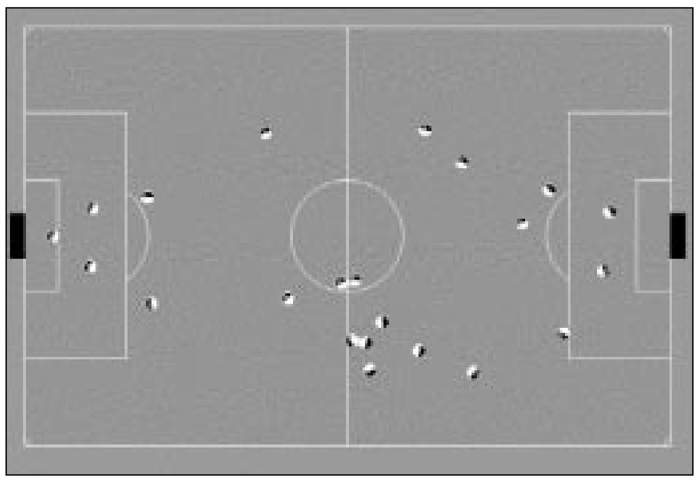
Coevolving Soccer Softbots

The University of Maryland's RoboCup simulator team consisted entirely of computer-evolved players developed with *genetic programming* (Koza 1992), a branch of evolutionary computation that uses natural selection to optimize over the space of computer algorithms. Unlike other entrants that fashioned good softbot teams from a battery of relatively well-understood robotics techniques, our goal was to see if it was even possible to use evolutionary computation to develop high-level soccer behaviors that were competitive with the human-crafted strategies of other teams. Although evolutionary computation has been successful in many fields, evolving a computer algorithm has proven challenging, especially in a domain such as robot soccer.

Our approach was to evolve a population of teams of Lisp *s*-expression algorithms, evaluating each team by attaching its algorithms to robot players and trying them out in the simulator. Early experiments tested individual players, but ultimately, the final runs pitted whole teams against each other using coevolution. After evaluation, a team's fitness assessment was based on its success relative to its opponent. This fitness score determined which teams would be selected to interbreed and form the next generation of algorithms.



"Everyone Go after the Ball": A Popular but Troublingly Suboptimal Early Team Strategy.



Soccer Teams Eventually Learn to Disperse Themselves throughout the Field.

The RoboCup soccer simulator makes evolutionary computation extremely difficult. The simulator gives noisy data, limited sensor information, and complex dynamics. Most problematic is that the simulator runs in real time; even at full throttle, games can take many seconds or even minutes. Unfortunately, evolving a team of 11 soccer players can require hundreds of thousands of evaluations; so, in the worst case, a single soccer evolution run could take a year or more to complete.

To keep the number of evaluations to a minimum, we severely limited the population size, which demanded special customizations to prevent the population from converging to a suboptimal strategy. We also cut down the number of evolved algorithms by grouping players into squads, with one algorithm to a squad, or using one single algorithm for the entire team (Luke and Spector 1996). We performed runs for both strategies; by the time of the competition, the single-team strategies had better fitness. To further speed up runs, evaluations were run in parallel on an ALPHA supercomputer cluster.

Because we had only one shot to evolve teams, we cannot make rigorous scientific claims about population development. Nonetheless, an admittedly anecdotal observation is still interesting. After a hesitant start, most early teams soon began to learn the worrisome suboptimal kiddy soccer strategy: Everyone go after the ball, and kick it to the goal (top figure). Thankfully, eventually players learned to hang back and protect the goal and, ultimately, disperse through the field to provide better coverage (bottom figure).

By the end of the final runs, the computer had produced teams that, as appropriate, passed to teammates, blocked the ball, protected different parts of the field, and tried to stay open. Our submission was surprisingly successful, beating its first two hand-coded competitors before succumbing. Hopefully this and other experiments will show that evolutionary computation is ready for a number of problems that previously have only been the purview of human ability.

– Sean Luke

Results

The champion of the RoboCup-97 simulation was AT HUMBOLDT. The runnerup was ANDHILL, the third-place winner was ISIS, and the fourth-place winner was CMUNITED.

Most of the matches in the first round were one-sided games. In a typical case, the score was 23–0. The main reason for such a score was that there was a big difference in players' individual skills. In the next RoboCup, such differences would become small because most of the RoboCup-97 teams agreed to make their programs public so that the know-how of a good technique could be shared.

The tactics of defense were not as good as those of offense for most teams. To defend an opponent attack effectively, a team must have a tactical model of opponent offense, but because RoboCup-97 was the first competition, no team knew what kind of tactics were possible. Now, we can build opponent models by watching replays of the matches. With such models, defense will become more effective in the next RoboCup.

Both finalists were in the same group (group D) in the first round. During the first match for group D, each team had already recognized another team as a rival in the final; so, ANDHILL used its weaker program for the first match. Moreover, ANDHILL tried to change its formation for the second half of the final match. Its efforts improved the performance of its team, so that the score of the second half was closer than the first half. However, ANDHILL was defeated by AT HUMBOLDT.

Of course, these strategies (changing tactics between matches and during a match) were made by human choice. It might be challenging to realize these strategies using a program. The process of each match was recorded in a log file.

Features of Teams

There were three types of approach to building player programs: (1) agent programming, (2) multiagent system, and (3) learning.

Agent Programming Because situations in the field simulated by SOCCER SERVER change dynamically, each player should be programmed as an agent in a dynamic environment. A main issue in this programming is "real time–ness." A player should perform a suitable play for the current situation as soon as possible. For example, when the ball is in front of its own goal and the opponent player is coming, then the goalie should kick the ball away as soon as possible before searching for a teammate that is the best receiver for the pass.

Many participants used reactive systems to

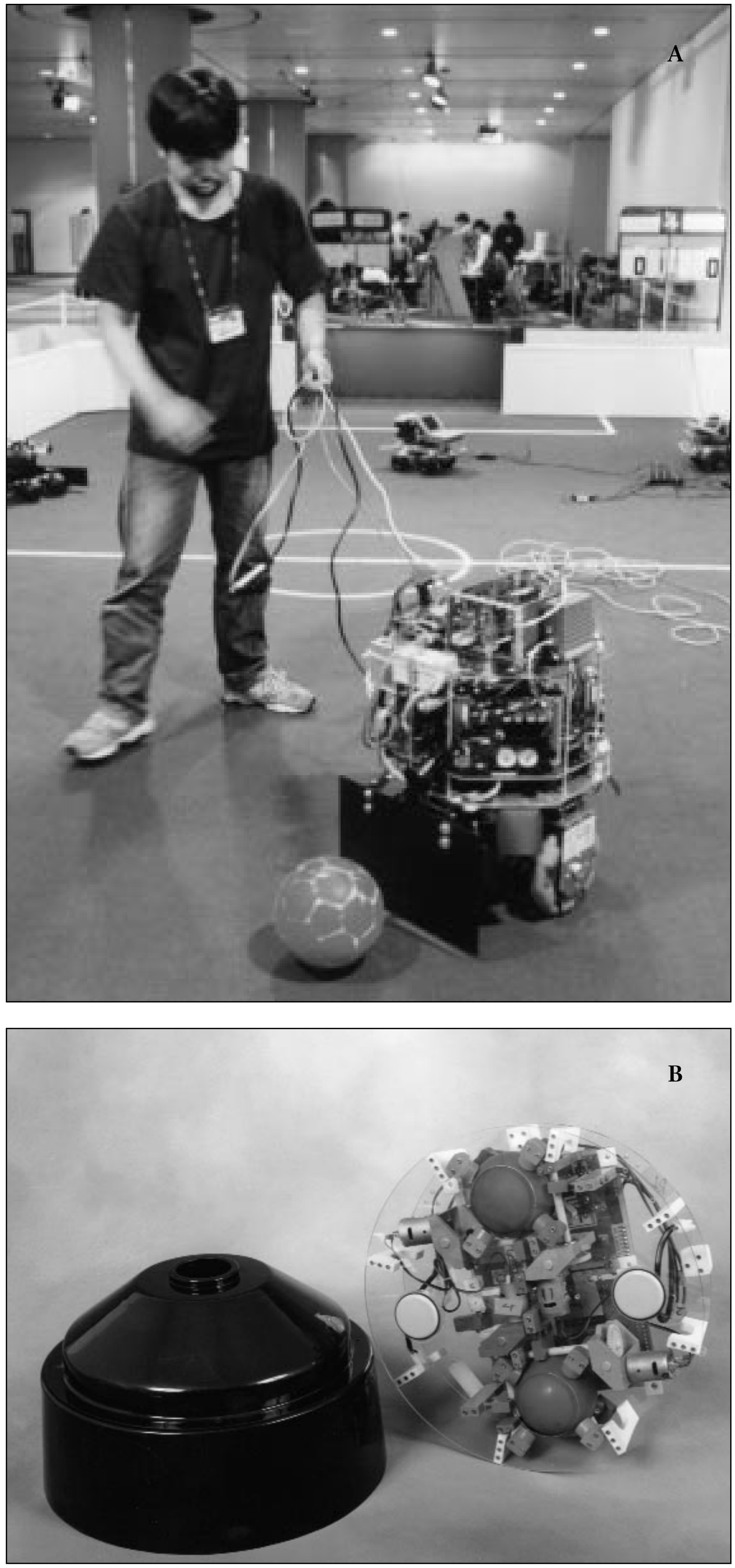


Figure 7. The Omnidirectional Mechanism.
A. UTTORI UNITED. B. TRACKIES.

ISIS: An Explicit Model of Teamwork at RoboCup–97

ISIS (ISI synthetic) won the third-place prize in the RoboCup-97 Simulation League tournament. ISIS was also the top U.S. team. Although ISIS's performance in the tournament was initially marked by lopsided wins, its later games were exciting and close; it twice won in overtime. In terms of research accomplishments, ISIS illustrated the reuse of STEAM, a general model of teamwork (Tambe 1997), that both reduced its development time and improved teamwork flexibility.

ISIS's development was driven by the three research challenges emphasized by the RoboCup simulation league: (1) teamwork, (2) multiagent learning, and (3) agent and team modeling. With respect to teamwork, our previous work was based on the development of pilot-agent teams for real-world combat simulations. For this work, providing individual agents with preplanned, domain-specific coordination knowledge led to teamwork inflexibility. Furthermore, the coordination knowledge was not reusable. STEAM, a general, explicit model of teamwork, was developed to alleviate these difficulties. STEAM requires that individual team members explicitly represent its team's goals, plans, and mutual beliefs. It then enables team members to autonomously reason about coordination and communication in teamwork, providing improved flexibility. Given its domain independence, it also enables reuse across domains—here, RoboCup provided a challenging test domain given its substantial dissimilarity to the original domain. However, a promising 35 percent of STEAM code was reused for RoboCup. Indeed, all the current communication among ISIS agents is driven by STEAM's general-purpose reasoning about teamwork. For example, midfielders communicate with each other about an approaching threat so they can coordinate their defense. With possible improvement in STEAM reuse in the future, such coordination might improve as well.

ISIS also took initial steps toward addressing the challenge of multiagent learning. Using C4.5, ISIS players learned offline to choose an intelligent kicking direction, avoiding areas of concentration for opponent players. Further aspects of multiagent learning, as well as arenas of agent and team modeling (particularly to recognize opponents' strategies), are under active investigation.

ISIS agents were developed as a two-level architecture: The lower level, developed in C, processes input and rapidly computes recommendations for directions to turn (to intercept the ball) or possible directions to kick the ball (for example, kicking direction computed by C4.5 rules mentioned previously or kicking direction to clear the ball). However, the lower level does not make any decisions. Instead, all the decision making rests with the higher level, implemented in the SOAR integrated AI architecture, which takes into account the recommendations made by the lower level. STEAM's teamwork reasoning is currently also implemented in SOAR and has led to enhancements to the SOAR architecture. For example, explicit team operators, an enhancement of SOAR's individual operators, are used for explicit representation of a team's goal and plans.

Some key weaknesses of ISIS players stemmed from a somewhat inappropriate interaction with the RoboCup simulator: The simulator version used in RoboCup-97 allowed agents to take as many as 3 actions (1 action every 100 milliseconds [ms]) before sending them a sensor update (1 update every 300 ms). This action-to-sensor update ratio required that agents continually make predictions. Unfortunately, with weak predictive capabilities, ISIS agents could not always quickly locate and intercept the ball or maintain awareness of positions of teammates and opponents. However, the RoboCup simulator will evolve for RoboCup-98 toward more humanlike play.

We hope to continue working on ISIS in preparation for RoboCup-98 and meet the research challenges outlined for the simulation league. More information about ISIS, including the code, is available at www.isi.edu/soar/tambe/socteam.html.

– Milind Tambe, Jafar Adibi, Yaser Al-Onaizan, Ali Erdem, Gal A. Kaminka, Stacy C. Marsella, Ion Muslea, and Marcello Tallis

Group A: LAI (Université Carlos III De Madrid), FC MELLON (CMU), RM KNIGHTS (RMIT), ICHIMURA (Kinki University, Japan).

Group B: RIEKKI (University of Oulu, Finland), CMUNITED (CMU), HEADLESS CHICKENS (RMIT), NIT-STONES (Nagoya Institute of Technology, Japan).

Group C: MICROB (Université de Paris VI), BALCH (Georgia Institute of Technology), PROJECT MAGI (Aoyama University, Japan), OHTA (Tokyo Institute of Technology, Japan).

Group D: AT HUMBOLDT (Humboldt University, Germany), TEAM SICILY (Stanford University), KASUGA-BITO (Chubu University, Japan), ANDHILL (Tokyo Institute of Technology, Japan).

Group E: PAGELLO (University of Padua, Italy), HAARLEM (Chukyo University, Japan), ORIENT (Toyo University, Japan).

Group F: UBC DYNAMO (University of British Columbia, Canada), LUKE (University of Maryland), OGALETS (University of Tokyo, Japan), TUT (Toyohashi University of Technology, Japan).

Group G: CHRISTENSEN (Charlmers University of Technology, Sweden), TEAM GAMMA (ETL, Japan), KOSUE (Kinki University, Japan).

Group H: ISIS (USC-ISI), GARBAGE COLLECTORS (private, Japan), I&W (Waseda University, Japan).

Table 3. RoboCup-97 Team List.

realize this behavior. These reactive systems are based on, or inspired by, Brooks's subsumption architecture. They also used planning systems to make high-level play decisions such as pass-receiver selection. For example, AT HUMBOLDT, PAGELLO, RICKKI, and LAI used the combination of reactive systems and planning systems.

Multiagent System Because soccer is a team game, a couple of teams were programmed based on multiagent systems to describe team plays, for example, ISIS, RM KNIGHTS, FC MELLON, and TEAM SICILY. The following issues had to be dealt with: how to define team plays and decompose them into roles, how to assign the roles to players, and how and what to communicate effectively among players.

ISIS was the most successful team using the multiagent system. It built its programs based on STEAM, which it developed to describe teamwork in a party of helicopters. Remarkably, 35 percent of the code is shared between the soccer players and the party of helicopters even though these domains are quite different from each other (see sidebar).

Machine Learning Machine learning was also a major technique for programming player clients. FC MELLON, ANDHILL, and BALCH used machine learning. Machine learning can be used to improve a player's skill of handling the ball, for example, approaching a moving ball or kicking with suitable power; to organize the formation of players; and to acquire suitable conditions for a certain play, for example, which teammate to pass to.

Peter Stone and Manuela Veloso of FC MELLON used a couple of learning techniques for various levels of play: neural network learning to improve basic skill and decision tree learning for high-level collaboration.

Genetic Programming The most impressive technology used in the simulation league was genetic programming. LUKE used genetic programming to improve the behavior of its players. At first, it prepared pieces of program such as (kick POWER DIR), (turn POWER), (plus X Y), and (if COND THEN ELSE). It combined these pieces randomly and built programs to control players, then it made teams of the players and made them compete with each other. Using the result of the competition, it selected teams that got a good score and generated new programs with the genetic programming method from the selected teams. It repeated this cycle and evolved player programs. Of course, this evolution was done before the RoboCup competition, and LUKE participated in the competition using programs in the final generation.

LUKE also reported interesting changes in behavior during the evolution. In the first generation of the programs, most of the players did not see the ball. After some alterations, all players came to chase the ball. Then, some players came to defend their own goals after more alterations. Finally, all players on a team spread on the field and passed the ball smoothly. Interestingly, these changes are similar to developments over the history of human soccer.

The Scientific Challenge Award was given to LUKE for demonstrating the utility of this genetic programming approach (see sidebar).

Computer versus Human

After the final match, an extra exhibition match took place between AT HUMBOLDT, the champion team, and a human team.[2] The human team consisted of 11 men selected from participants. Each man controlled a player from the console by a simple interface.

AT HUMBOLDT won 8–1. However, this result does not means that the computer is strong enough. Because it was the first time for most people on the human team to control players directly using the interface, AT HUMBOLDT got goals one-sidedly, so that the score of the first half was 6–0. The human team improved in controlling players during the second half. As a result, the score of the second half was 2–1. AT HUMBOLDT was stronger than the human team, but it was close. Every member of the human team said that humans would win the next time. There are a couple of reasons why humans' skills improved so quickly: First, the ability of humans to learn skills to control players is high. Their skill improved after only 5 minutes. Second, humans are good at modeling opponent tactics. AT HUMBOLDT (as well as other computer teams) has few tactics to achieve goals; so, it was easy to cut passes. Third, the human team used communication effectively. It used voices for communicating to exchange the plan of defense and offense.

Future

Overall, RoboCup-97 was very successful (Kitano 1998). It made clear scientific issues necessary in developing multiple robots to work collaboratively in the real world and helped us identify technical challenges in bringing multiple robots to an out-of-lab setup. Results of the simulator league clearly demonstrate the strength of the AI approach over hard-coded programming. It was encouraging for the AI community that for a game with this level of complexity, the AI-based approach proved to be more effective than

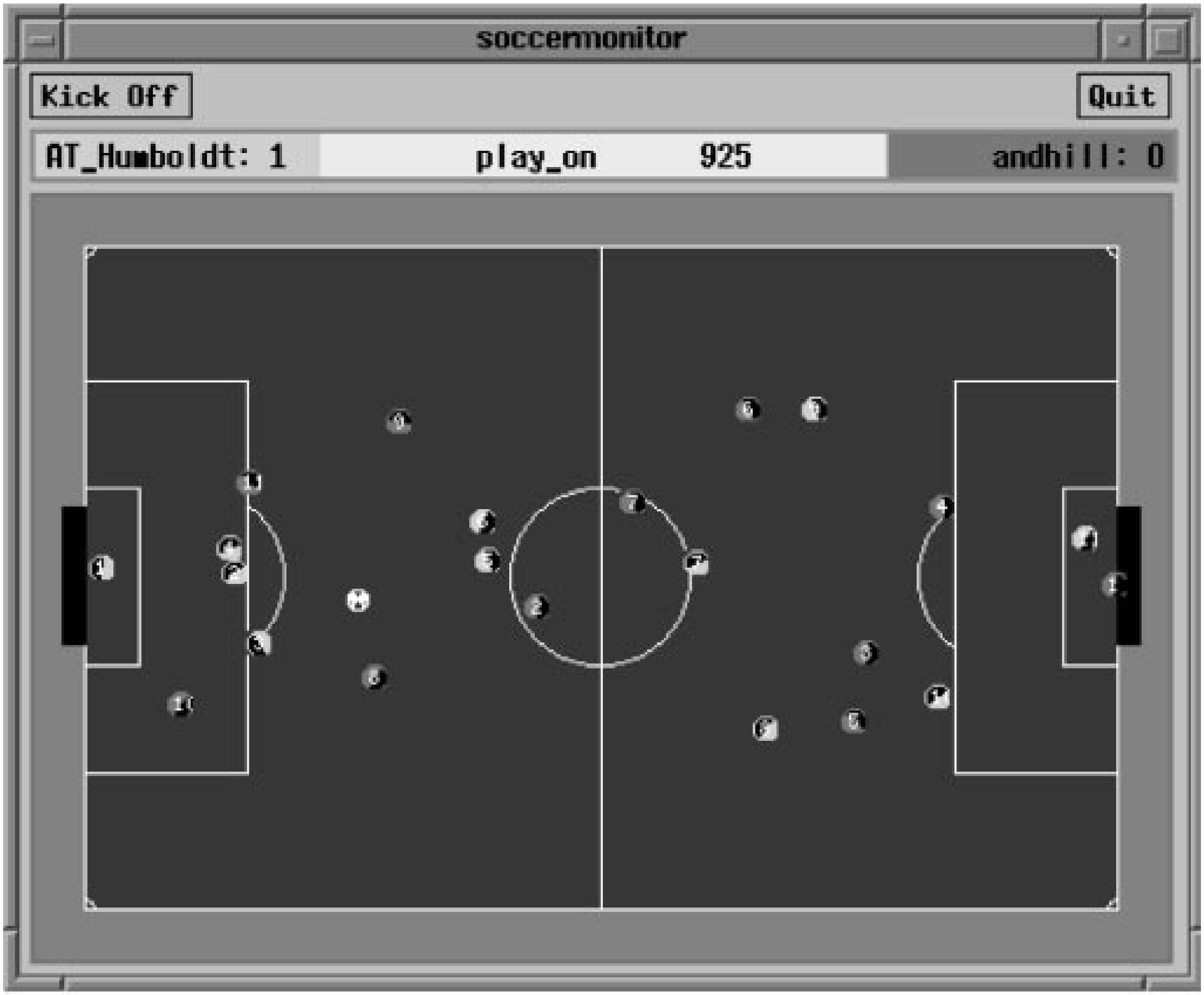


Figure 8. SOCCER SERVER *(A Scene of the Final RoboCup-97Match).*

hard-coded hacking. RoboCup will continue in the future until a robot soccer team beats the human World Cup Soccer champion team, and RoboCup offers a set of short-term (two-year) challenge tasks to promote research activities among RoboCup participants (Kitano, Tambe, et al. 1997). RoboCup-98, the Second Robot World Cup Soccer, was held in July 1998 during the Third International Conference on Multiagent Systems (ICMAS-98) in Paris, France. RoboCup-99 will be held during the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99) in Stockholm, Sweden. We hope RoboCup offers a good target for twenty-first century AI.

Notes

1.ci.etl.go.jp/noda/soccer/server.html or www.robocup.org.

2. This exhibition was organized by Sean Luke and Tucker Balch.

References

Kitano, H., ed. 1998. *RoboCup-97: Robot Soccer World Cup I.* Berlin: Springer-Verlag.

Kitano, H., ed. 1997. Proceedings of the First International Workshop on RoboCup, IJCAI-97, 22–28 August, Nagoya, Japan.

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Osawa, E.; and Matsubara, H. 1997. RoboCup: A Challenge Problem for AI. *AI Magazine* 18(1): 73–85.

Kitano, H.; Tambe, M.; Stone, P.; Veloso, M.; Coradeschi, S.; Osawa, E.; Matsubara, H.; Noda, I.; and Asada, M. 1997. The RoboCup Synthetic Challenge Ô97. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 24–29. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Koza, John R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, Mass.: The MIT Press.

Luke, S., and Spector, L. 1996. Evolving Teamwork and Coordination with Genetic Programming. In *Proceedings of the First Annual Conference on Genetic Programming (GP-96),* eds. J. R. Koza et al., 150–156. Cambridge, Mass.: The MIT Press.

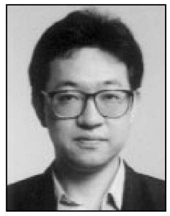
Tambe, M. 1997. Toward Flexible Teamwork. *Journal of Artificial Intelligence Research* 7:83–124.

Itsuki Noda is a senior researcher at the Electrotechnical Laboratories, Ministry of International Trade and Industry, Japan. He received his Ph.D. in technology from the Kyoto University in 1995. His current research interests include neural networks, machine learning, and multiagent systems. He is a developer of SOCCER SERVER used in RoboCup.

Shóji Suzuki is a research assistant at Osaka University. He received his Ph.D. in 1993 from Tsukuba University, Japan. His interest is mobile robots.

Hitoshi Matsubara is a senior researcher at the Electrotechnical Laboratories, Ministry of International Trade and Industry, Japan. He received his Ph.D. in computer science from the University of Tokyo in 1986. His current research interests include multiagent systems and game programming.

Minoru Asada received his Ph.D. from Osaka University in 1982. He is currently a professor in the Department of Adaptive Machine Systems at Osaka University. He is interested in robot learning.



Hiroaki Kitano is a senior researcher at Sony Computer Science Laboratory as well as visiting professor at the National Institute of Genetics and the Tokyo Medical and Dental University. He received his B.A. in physics in 1984 from International Christian University and then joined NEC Corporation. From 1988 to 1993, he was a visiting researcher at the Center for Machine Translation at Carnegie Mellon University. Kitano received his Ph.D. from Kyoto University in 1991 for his work on a spoken-language translation system on massively parallel machines, which led to the Computers and Thought Award from the International Joint Conferences on Artificial Intelligence in 1993. Kitano joined Sony Computer Science Laboratory in 1993, where he works on RoboCup and systems biology. He is president of the RoboCup Federation.

Milind Tambe is a senior research scientist at University of Southern California Information Sciences Institute (USC/ISI) and a research assistant professor with the USC Computer Science Department. He completed his Ph.D. at Carnegie Mellon University in 1991. His research interests are in the area of multiagents, particularly in collaboration, teamwork, agent modeling, and negotiations.

Ali Erdem is a graduate research assistant at the Information Sciences Institute, University of Southern California (USC), and a Ph.D. candidate in the USC Computer Science Department. He received his M.Sc. in computer science from USC in 1991 and his B.Sc. in electrical engineering from the Bogazici University in 1990. His research interests include knowledge-based software engineering, intelligent tutoring systems, and machine translation.

Gal A. Kaminka is a Ph.D. student in the Computer Science Department and Information Sciences Institute at the University of Southern California. His research interests include plan-execution monitoring and robustness, agents that are able to reason about themselves and others, and uses of fuzzy set theory in AI.

Stacy Marsella is a researcher at the Information Sciences Institute (ISI), University of Southern California. He is a member of ISI's Center for Advanced Research in Technology for Education. His work spans several areas of intelligent agent research, including synthetic agent teams as well as pedagogical agents for assessing human and synthetic agent teams. He received his Ph.D. from Rutgers University.

Ion Muslea is a Ph.D. candidate in the Computer Science Department and the Information Sciences Institute at the University of Southern California. He received his B.S. from the Technical University of Cluj-Napoca (Romania) and his M.S. from West Virginia University. His research interests include machine learning, information integration, and information extraction.

Marcelo Tallis is a Ph.D. candidate in the Computer Science Department and Information Sciences Institute at the University of Southern California. His research interests include knowledge-based systems, knowledge acquisition, and machine learning.

Sean Luke is a graduate research assistant at the Department of Computer Science at the University of Maryland at College Park. His interests include genetic programming, evolving cooperative agent behavior, neural development, and web-based knowledge representation systems. His e-mail address is seanl@cs.umd.edu.