A Distributed Multiagent System for RoboCup

Maritza Bracho de Rodríguez[1], Miguel Angel Castro[2], José Ali Moreno[2]

1 Unidad de Investigación en Inteligencia Artificial, Decanato de Ciencias y Tecnología
Universidad Centroccidental Lisandro Alvarado. Barquisimeto. Venezuela.
mbracho@ucla.edu.ve

2 Laboratorio de Computación Emergente, Facultades de Ciencias e Ingeniería,
Universidad Central de Venezuela. Caracas. Venezuela.
{mcastro, jose}@neurona.ciens.ucv.ve

Abstract. The RoboCup Small Size Robot Football League competition is considered by the robotic and A.I. research community as an ideal platform to design and test distributed multiagent systems. A description of a robotic soccer team in terms of its major hardware and software components is presented. The mobile robots and a commodity cluster are the main hardware components. A hybrid architecture in terms of a stimulus response architecture for the onboard agents, a vertical layered architecture for the offboard agents and a supervisory mechanism to coordinate the agent behaviors in order to achieve a coherent and high performance operation is introduced. The global functionality is produced by the interaction of multiple processes using methods and techniques of emergent computation integrated with the traditional perspective of knowledge representation. Experimental results show that the proposed architecture renders reasonable real time response so as to face the challenges posed in the RoboCup Small-Size Robot League.
Keywords: Distributed Multiagent System, Multiagent Architectures, Robotics.

1 Introduction

The World Cup Robot Soccer, RoboCup, has been proposed as an initiative to promote research in the Artificial Intelligence and Robotic domains [15]. In this way, the research goals consist on the solution of an integrated task that covers a broad range of problems including vision and object recognition, multiagent systems, strategic decision making, path planning, intelligent robot control, motor control, and many more. It can then be stated that RoboCup provides a very useful framework to study physical bodies executing intelligent behaviors, [1]. In particular, the Small Size Robot League, (F180), of the RoboCup competition presents itself as a convenient and affordable platform to design and test distributed multiagent systems. The completion of the main task is approached by the resolution of partial goals such as: (1) Modeling and construction of a robotic architecture with limitations on its onboard processing capacity imposed by the restrictions in the size of the robots. (2) Modeling and development of a global vision system for robust and efficient sensing. (3) Design of algorithms of reduced complexity that may handle real time path and motion planning. (4) Modeling of systems of multiple agents capable of executing soccer behaviors. (5) Design and implementation of efficient communications between the offboard and onboard processes.

Different approaches had been proposed in the literature to construct this type of systems where a team of robots performs a set of cooperative tasks while playing soccer. CMUnited [19] uses an arquitecture that combines high and low level reasoning viewing the system as the combination of the robots, a vision camera connected to centralized interface computer and several client modules as the minds of the robot players. Big Red [12] implements a system that contains four main components: the vision system, the artificial intelligence system, the wireless communications to the robots, and the robots. The AI system broadcasts commands over the wireless network to the robots, which in turn carry out the commands. The vision system determinates the current game state and sends the state to the AI computers for further processing and determination of a strategy to be executed. LuckyStar [14], **-** RoboCup 2001 Small Size Champions **-**, uses a host computer that provides the intelligence for all the robots. It processes the vision data and computes the next move for each robot. The outputs, which are the translation and rotation speed of the robots, are sent via a RF transmitter

In this paper a hybrid control architecture distributed between a team of mobile robots and a small cluster of heterogeneous personal computers is introduced. Every robot is considered as an onboard agent, with stimulus response architecture, performing navigation. There are also offboard agents, with a vertical layered architecture, performing behaviors according to team strategies and generating orders to its respective onboard agent.

The behaviors of the agents are arranged according to their position in space, in this sense, the football field is divided in several functional regions. A coherent and high performance operation of the agents is achieved by coordinating their activities through a supervisory mechanism. The global coherent functionality emerges from the interaction of multiple processes using methods and techniques from emergent computation, [13], [20], integrated with the traditional perspective of knowledge representation.

The organization of the paper is as follows: in the second section, an overview of the system architecture is presented, with a brief description of the vision system, the robotic architecture and the communication system. In the third section, the distributed multiagent software of the system is presented. In the fourth section, agent behaviors are explained. In the fifth section experimental results are discussed and in the final section the conclusions and further work are described.

2 System Architecture

The proposed architecture is based on a small heterogeneous cluster, a commodity cluster, that consists of a small set of personal computers, interconnected by a local area network, working as a local area multiprocessor. Some examples of these clusters are Beowulf, HPVM Project, Berkeley NOW, [3], [5], [11], [16].

This integrated computer resources provides a distributed environment creating the illusion of only one parallel machine in which the system is perceived as single unified system. The details of the local area network and the individual machines are not directly visible.

Figure 1 shows a diagram of the hardware system with its main components: (1) A local area multiprocessor integrated by a local area network of personal computers, PCs. (2) A global vision system composed by a digital camera connected to a vision server via its parallel port. (3) The workspace consisting on a football field built in accordance with the rules of the RoboCup small size robot league. (4) A three robots team, performing roles of forward, midfielder and defender.

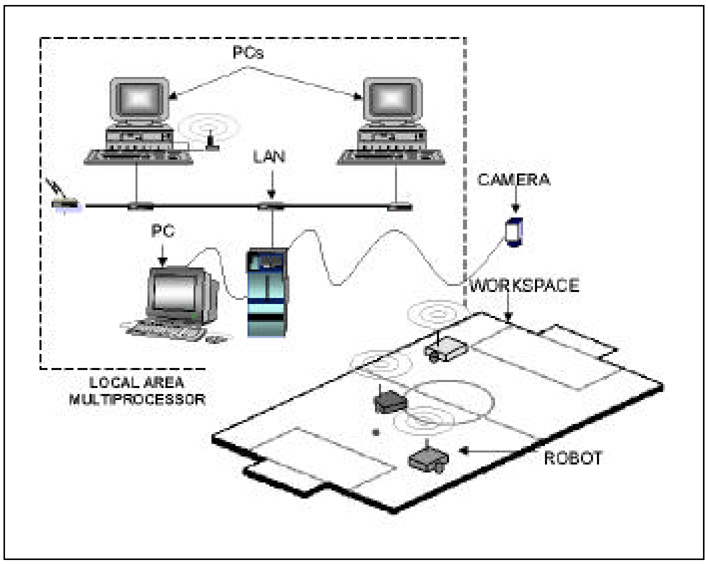

Fig. 1. System Overview

2.1 Vision System

The global vision system is the sensorial source which allows the recognition and segmentation of the objects in the workspace and hence the global positioning of the robots and the ball in the playing field. It consists of a Connectix Quick Cam digital camera, placed 2.5 m. directly above the workspace connected to a PC by its parallel port. The camera is operated in 32-bit mode with a resolution of 160x320 pixels, and a frame rate of approximately 5.6 frames per second.

2.2 Robotic Architecture

UCV Robot [7], - shown in Figure 2 -, is a small radio controlled mobile robot, built in two layers, using low cost hardware parts. Its dimensions are width: 12 cm., length:

12 cm. and height: 10 cm. The lower platform holds two wheels of 6.4 cm diameter, in wheelchair position and two servomotors, type FMA Direct S3501AM, one for each of them. The wheels are powered independently for both steering and driving. The motors are controlled by pulse width modulation using a switching device, - to power on and off rapidly -, and a standard scheme to indicate to the motor controller which speed is to be set. This standard uses a variable width pulse, repeated periodically to specify the position of the servo. The control pulse is defined by a width that represents "stop" and by a width-change that yields "full travel". A typical value for the stop width on this type of servo is 1.50 ms, with a maximum of 2.0 ms and a minimum of 1.0 ms for full travel in opposite directions. The servo requires the reception of the control pulse approximately every 20 ms or 50 times a second. An incremental shaft encoder is placed into the servo motor gear to count approximately 400 pulses per wheel revolution. Every 100 ms the proportional motor speed pulses are compared to the pulses accumulated by the shaft encoder. In case of differences an adjustment is made to motor speed.

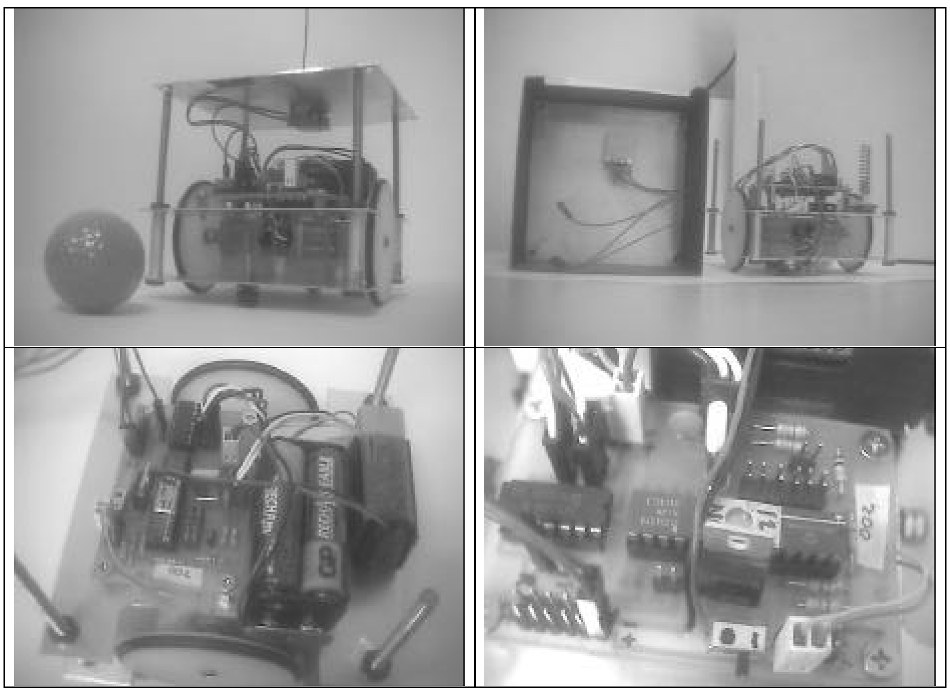

Fig. 2. Different Views of UCV Robot

The second platform of the robot holds a communication and CPU board, a voltage regulator and the battery pack. The board is equipped with a Microchip pic16C73A processor. The robot communicates with the main computer by a radio transceiver device, RMIV Glolab Inc., at 418 MHz with a transfer rate of 4800 bps. The microcontroller on the CPU board commands the servos and manages the communications.

2.3 Communications

The communications are executed via a radio transceiver connected by an interface to the serial port of one of the computers, in the LAN. The transmitted radio signal with the commands to the robots is received by the robot radio transceiver. The communication CPU board contains a micro controller that processes and executes data verification. The protocol communication uses a master slave model, in which only the master can initiate transactions. Each robot, a slave, has a unique address and responds taking the action requested in the command. The protocol establishes the format for the command by placing into the transmission bytes, the robot address, a function code defining the requested action, any data to be sent, and an error-checking field. The function code in the command tells the addressed robot what kind of action to perform. The data bytes contain additional information to perform the function. The error-checking field provides a method for the robot to validate the integrity of the message contents.

3 Distributed Multiagent System

The system is completely autonomous and free of human intervention during a game set. Its main components are: the vision system software executed in a node that serves several client processes, the control supervisor and the agent processes executed in other cluster nodes and the motion commands performed by the robots. All data is served between the processes running in the cluster nodes through message passing interface, MPI, - “LAM” version, University of Notre Dame, [17] -, and the commands are sent to robots by radio transmissions.

3.1 Functional Distributed Multiagent Architecture

In this distributed approach there are several active agents. To achieve a coherent and high performance operation these agents are coordinated by a supervisory mechanism. This supervisor keeps and receives complete information about the current game status and the positions of each of the agents. Based on this information, it evaluates the state of the world and the state of the agents and establishes the strategy to follow according to the results of the evaluation and to preprogrammed scenarios.

This strategy is sent to the offboard agents where a process interprets it in function of the role assigned to the agent and traduces the strategy in terms of a basic behavior to be performed. Examples of these are intercept the ball, follow the ball or another player, pass the ball, and move to.

This interpretation is provided to a path planner, inside the agent, that applies a cellular automata dynamics to generate free collision paths between two points, [4], [6]. To compute paths, this planner works on a finite collection of non intercepting square cells, a cellular space representing the workspace, where each object or goal is a particular set of cells and the free space is the set of cells not belonging to any object. A path is a finite discrete sequence of cells. In phase one, the planner inputs the discrete cellular space and places a predefined template of cells over the goal position, to direct the movement of the robot. In phase two, the configuration resulting in the prior phase is used to evolve a dynamic that computes the Manhattan distance between the initial and goal positions yielding a cellular automata with the following possible states: (0) free space, (1) obstacles, -other robots -, (2) initial position, (3) goal position, (4) Manhattan distance 1 to the goal, ..., (3+*l*) Manhattan distance *l* to the goal. To calculate the path the steepest descend of the function is followed going backwards from the goal to the start positions. The sequence of cells that defines the path and the initial orientation of the robot are used to produce a list of commands for the robot. This list is transmitted to the onboard agent via the communications administrator.

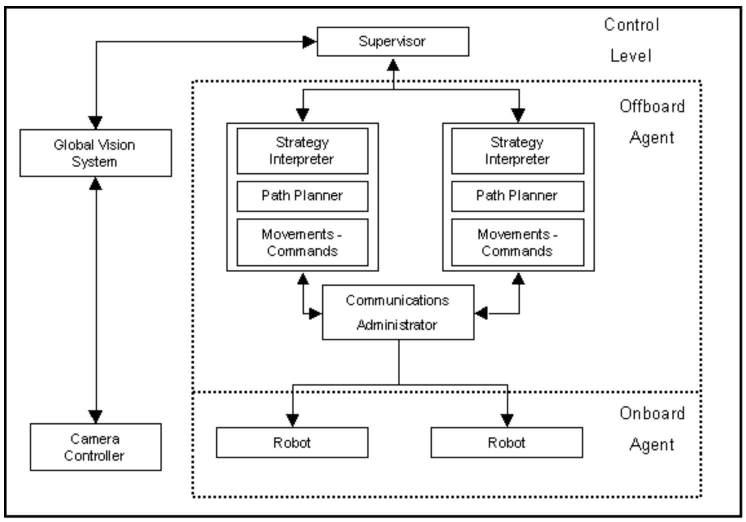

Fig. 3. Distributed Multiagent Arquitecture

Each robot contains an onboard agent with a stimulus response arquitecture, [8], as shown in Figure 4.

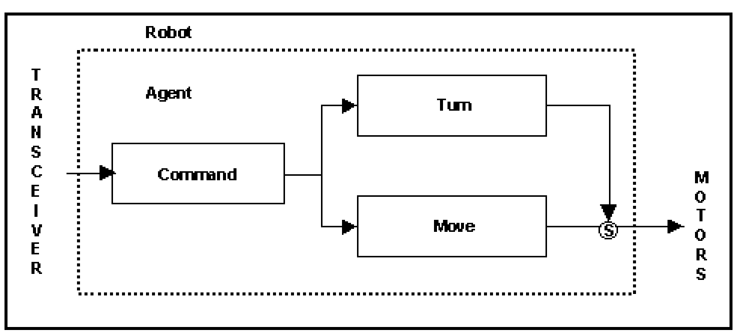

Fig. 4. Stimulus Response Arquitecture

The stimuli are the commands received via radio and the response is the highest active behavior of a fixed priority network. This embedded agent performs low level navigation tasks. The commands received by de radio transceivers are of the type: Turn (an angle), Move (a distance). These commands are interpreted by the onboard CPU, which sends the orders to the corresponding actuators.

3.2 Global Vision System

Since the RoboCup rules specify well-defined colors for different objects in the field, the identification of the pixel colors lead to a simple and straightforward manner for the classification and recognition of the objects in the scene. To produce the color segmentation several methods have been presented in the literature [9], [10], [18]. In our system, the RGB color space is partitioned using linear color thresholds and every single pixel is classified by whether or not it is contained within the subcube that is delimited by the thresholds. A two-operation test is performed to determine if a pixel belongs to a subcube, one for the three channels RGB against the minimum threshold and the other one against the maximum threshold. This test is based on the fact that the subtraction of two bit fields will result in a unit to borrow, if the first operand is smaller than the second one, [2], [4], [7].

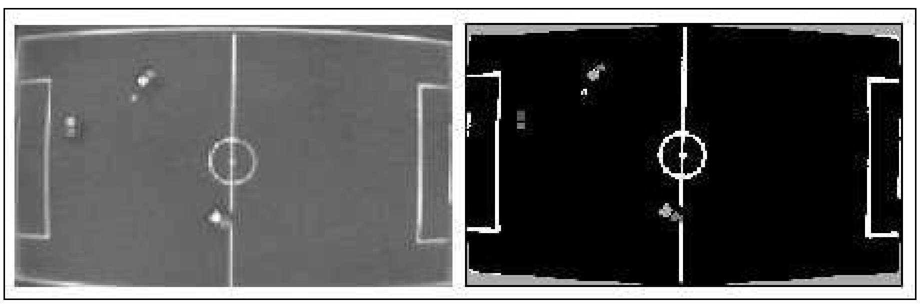

Fig. 5. Left: Input image. Right: Image after color detection.

The input of the algorithm is a 160x120-pixel resolution image taken by the digital camera above the playing field, - Left in Figure 5 -. A pixel color classification is performed, the pixel information is represented on a cellular automata as a discrete rectangular pattern with four possible states: (0) free, (1) ball, (2) Color 1, (3) Color 2, (4) Color 3, (5) Color 4. Then, a filter algorithm is applied in order to produce the segmentation of the colors. As long as the colors are delimited static and mobile objects are identified and positioned in a specific coordinate system. This processed cellular automata pattern is supplied to the client process in the form of the cell state values. The global vision system communicates to other process in the whole system using a client server model: the client requests a processed image and the server replies with the last computed cellular automata pattern, in this way a discrete version of the workspace is provided.

In a practical setting the robustness of this method allows the recognition of a five-color classes domain. Considering scenes similar to those occurring in the RoboCup, robot soccer cup tournament, detected colors can be used to identify the field, the ball and the dual color identification of the robots. The field background and white field lines do not classify in any of the color classes and are hence identified by default.

The scene, - Right Figure 5 -, shows three types of robots differentiated in colors, a ball, goals A and B, the field and the field lines. This information is supplied to client computers in the form of a pattern of cell state values using a cellular automata representation.

4. Agent Behaviors

The behavior of the agents depends on their position in the workspace, for this reason the football field is divided in several functional regions. In this approach every robot tries to position itself in the area of the field defined according to its assigned role. The agents may perform one of the following roles: (1) Defender, with the primal mission of guarding the front of its own goal following protection paths, covering with its body the small area defined by the angle that forms the ball with the goal borders. If possible, it should move the ball towards the midfield. (2) MidFielder, which has the possibility to play different roles. The midfield defender role if an opponent robot has the ball. In this case it should try to block the ball and the attacking robot. The midfield forward role if it has the ball. In this case it should try to pass the ball to the forward or it should lead the ball to the opponent goal. (3) Forward, - or attacker -, which tries to take control of the ball and to lead it to the opponent goal if the ball is in its zone of influence. If the ball is not in its zone, it should keep track of the ball.

The robots must posses some basic skills to execute cooperative and competitive behaviors. These skills are standard to all the robots, independently of the role that it is playing. and are supported by basic behaviors defined at the agents, as: Move the Ball, Intercept the Ball, Follow the Ball, Follow a Robot, Go to Position.

Finally, each robot contains an embedded agent that performs low level navigation tasks. The commands to the robots are the input to stimulus response machines that execute the reactive behaviors of the embedded agent.

5. Experimentation

This distributed multiagent system is working in a commodity small cluster, with heterogeneous nodes, some PCs of 233 MHz and one PC of 550 MHz. The experiments have been carried out using UCV robots chasing a ball on a football playing field. The actual execution time of the whole system is approximately 2.5 frames/second and it depends on several factors: (1) The frame rate of the camera and the data communication mode by the parallel port limits the efficiency in the performance of the global vision system. While the process that interprets the color classifications, filters, segments the objects and produces the cellular automata representation is capable of processing 18 frames/second of 160x120 pixels, the camera only has a frame rate of 5.6 frames/second. Hence, the system waits for the camera in its process of acquiring the image and transferring the data. (2) The off board agent process is capable of path planning and motion planning at a reasonable 30 frames/second. In spite of this, the supervisory mechanism is a bottleneck in the efficiency of the multiagent system. Its cause lays mainly in the organization of the communication protocol with the agents and in the interpretation of their strategy. This has as consequence a large waiting time for the agents. Further work on this mechanism is being made in order to increase its efficiency. (3) Finally, although the robots show a reasonable performance, they require improvements in speed and motion control.

6. Conclusions

This paper summarizes the ongoing research done by our group in the field of RoboCup and Multiagent system. It presents a distributed multiagent system for a robotic soccer team whose functionality integrates methods and techniques of traditional artificial intelligence with emergent computation.

The principal attributes of this system are: (1) A team of a small radio controlled mobile robots, constructed using low cost hardware parts. These robots allow the execution and test of diverse robotic algorithms, in a real world environment. (2) A cellular automata approach for image processing that provides robust and very efficient sensing. (3) A cellular automata approach for workspace representation, path and motion planning that renders very satisfactory real time performance. (4) A hybrid control arquitecture consisting of onboard agents performing navigation and offboard agents performing behaviors and generating orders to its respective onboard agent that shows satisfactory performance. (5) A small commodity cluster employing a message passing libraries that brings advantages as scalability, **-** allowing the aggregation of new components **-**, flexibility, **-** admitting different configurations for the arrangement of nodes **-**, and a low cost **-** high effectiveness relation.

Future work includes the incorporation of a digital camera with increased frame rate, the extension of the system to handle a team of five robots, the coupling of the system to these new components, the homogenizing of the nodes to the same family of processor, as well as the tuning of the overall system.

References

1. Asada, M., Stone, P., Kitano, H., Drogoul, A., Duhaut, D., Veloso, M., Asama, H., Suzuki, S. (1997). The RoboCup Physical Agent Challenge: Goals and Protocols for Phase I. Lectures Notes in Computer Science, Lectures Notes in Artificial Intelligence, 1395: 42-61. Berlin: Springer-Verlag.
2. Baltes, J. (1999). Practical Camera and Colour Calibration for Large Rooms. Lectures Notes in Computer Science, Lectures Notes in Artificial Intelligence, 1856: 128-135, Berlin: Springer-Verlag.
3. Becker, D., Sterling, T., Savarese, D., Dorband, J., Ranawake, U., Packer, C. (1995). Beowulf: a Parallel Workstation for Scientific Computation. Proceedings of the 1995

International Conference on Parallel Processing, (ICPP), 1:11-14. Beowulf Papers: http://www.beowulf.org/papers/papers.html

4. Behring, C., Bracho, M., Castro, M., Moreno, J. A. (2000). An Algorithm for Robot Path Planning with Cellular Automata. Proceedings of the Fourth International Conference on Cellular Automata for Research and Industry. ACRI2000 in Theoretical and Practical Issues on Cellular Automata: 11-19. Berling: Springer-Verlag
5. Beowulf at NASA: Goddard Space Flight Center , Earth and Space Sciences Project, NASA. (1994). http://beowulf.gsfc.nasa.gov/
6. Bracho, M., Moreno, J. A. (2000). Heuristic Algorithm for Robot Path Planning Based on Real Space Renormalization. Lecture Notes in Computer Science, Lectures Notes in Artificial Intelligence, 1952: 379-388. Berlin: Springer-Verlag
7. Bracho, M., Castro, M., Moreno, J. A. (2001). A Robotic Arquitecture for RoboCup. Proceedings of IX Conferencia de la Asociación Española para la Inteligencia Artificial – IV Jornadas de Transferencia Tecnológica de Inteligencia Artificial, 1: 675-684.
8. Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation, 2(1):14-23.
9. Brown, T., Koplowitz, J. (1979). The Weighted Nearest Neighbor Rule For Class Dependent Sample Sizes. IEEE Transactions on Information Theory: 617-619.
10. Bruce, J., Balch, T., Veloso, M. (1999). Fast Color Image Segmentation Using Commodity Hardware. Technical Report, Eigth AAAI Mobile Robot Competition. WS-99-15:47-56.
11. Culler, D., Arpaci-Dusseau, A., Arpaci-Dusseau, R., Chun, B., Lunetta, S., Mainwaring, A. (1997). Parallel Computing on the Berkeley NOW. Proceedings of th 9th Joint Symposium on Parallel Processing, Japan. http://now.cs.berkeley.edu/Papers2/index.html
12. D'Andrea, R., Lee, J., Hoffman, A., Samad-Yahaja, A., Cremean, L., Karpati, T. (1999). Big Red: The Cornell Small League Robot Soccer Team. Lectures Notes in Computer Science, Lectures Notes in Artificial Intelligence, 1856: 657-660. Berlin: Springer-Verlag.
13. Forrest, S. (1990). Emergent Computation: Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks. Physica D, Nonlinear Phenomena; 42: 1-11.
14. Kiat, B. N., Yee, L.W. (2002) LuckyStar2002. Ngee Ann Polytechnic. Alpha Centre & Electronic and Computer Engineering Departments. Singapore. http://www.np.edu.sg/~nbk
15. Kitano, H., Asada M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H. (1997). RoboCup a Challenge Problem for AI and Robotics. Lectures Notes in Computer Science, Lectures Notes in Artificial Intelligence, 1395: 1-19. Berlin: Springer-Verlag.
16. Moura, L., Buyya, R. Parallel Programming Models and Paradigms. Rajkumar Buyya (Eds). (1999). High Performance Cluster Computing: Programming and Applications; Vol 2. Upper Saddle Rive, New Jersey: Prentice Hall.
17. LAM – MPI Parallel Computing. (2002). http://www.lam-mpi.org/
18. Sanchez, J. G., Bracho, M., Moreno, J. A. (2000). Object Recognition an Tracking in a Digital Image. Proceedings of the International Joint Conference IBERAMIA-SBIA'2000: 264-273.
19. Veloso, M., Stone, P., Han, K., Achim, S. (1997). CMUnited-97 Small Robot Team. Lectures Notes in Computer Science, Lectures Notes in Artificial Intelligence, 1395: 242-256. Berlin: Springer-Verlag.
20. Wolfram, S. (1986). Approaches to Complexity Engineering. Physica D; 22: 385-399.