ISocRob 2007
Team Description Paper

Hugo Costelha[1], Nelson Ramos[2], João Estilita[3],
João Santos[4], Matteo Tajana[5], João Torres[6], Tiago Antunes[7], and Pedro Lima[8]

Institute for Systems and Robotics, Instituto Superior Técnico,
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
{[1]`hcostelha`,[8]`pal`}`@isr.ist.utl.pt`
{[2]`nmsra`,[4]`jcssa`}`@mega.ist.utl.pt`
{[3]`j.estilita`,[5]`matteo.tajana`,[6]`jgdtorres`,[7]`tmbcma`}`@gmail.com`
`http://socrob.isr.ist.utl.pt`

Abstract. The ISocRob team participated in RoboCup for the first time in 1998. Since then, its supporting research project (SocRob) has developed work on several different fields, such as software and hardware architectures, computer vision, navigation, behavior modeling, planning and multi-robots coordination. In recent years we developed, among others, new omnidirectional robots, a new multi-robot middleware, sensor fusion algorithms, and relational behaviors models based on discrete event systems. Some of the novelties we plan to take to RobotCup 2007 include a new 3D estimation algorithm based on a single omnidirectional camera, a common architecture for the Four-Legged and Middle-Size league and improved multi-robot behaviors, such as a Fuzzy decision-making based goalkeeper.

1 Introduction

The ISocRob team and its regular participation in the Middle-Size League (MSL) of RoboCup since 1998 are the competition side of the SocRob project[1], a research endeavour of the Intelligent Systems Laboratory of the Institute for Systems and Robotics at *Instituto Superior Técnico* (ISR/IST), Technical University of Lisbon, which started in 1997. The project goal is to develop a novel approach to the design of a population of cooperative robots based on concepts borrowed from Systems Theory [1] and Distributed Artificial Intelligence [2]. Besides the MSL team, ISocRob has participated in RoboCup Soccer Simulation League in 2003 and 2004, and in the RoboCup Four-Legged League in 2006.

This paper describes the current status of the SocRob project research progresses and endeavours, as well as the status of the ISocRob team. We focus on the past addressed research, but also on current related research not yet (fully) applied to the actual robots.

---

[1] The acronym of the project stands both for **Society of Robots** and Soccer Robots, the case study where we are testing our population of robots.

Fig. 1. ISocRob omnidirectional robots

The remaining of the paper is divided in two sections: Section 2 describes the details of the robots hardware and software architecture. Section 3 addresses the on-going and some past research developed under the project, as well as new research challenges.

2 Hardware and Software

2.1 Hardware

ISocRob's team has been composed, from 2000-2004, of four Nomadic Super Scout II robots. In 2005 and 2006, ISR/IST developed 5 new omnidirectional robots, in a joint venture with the Portuguese SMEs IdMind (responsible for the electronic hardware) and ServiLog (responsible for the mechanical hardware). Both the mechanical and electronic hardware are now finished although, given the nature of the project, improvements are always being done. The full robotic team is shown in Fig. 1. We are currently undergoing changes on the old Nomadic team in order to bring them back to a competition capable state.

The omnidirectional robots have most of their processing power concentrated on a NEC Versa FS900 laptop, with an INTEL Centrino at 1.6 GHz, 512MB of RAM and a 30GB disk. The laptop includes a CD-ROM, wireless 802.11b, 3 USB 2.0 ports, and 1 mini-firewire port, as well as a spare Li-Ion battery for extra autonomy.

Three MAXON DC motors model RE35/118776, nominal voltage 15 V (operated at 12 V) and 91:6 (or 15.1(6):1) gear ratio (ref. MAXON 203116) support each robot tripod kinematic structure. The maximum velocity achievable with such motors is estimated to be 4m/s, leading to a maximum translational speed of the robot around 3.5m/s and a maximum rotational speed of 20rad/s (1146 degree/s). The motors are controlled by master-slave PIC based controllers developed by IdMind and us, connected to the motor through an H bridge motor driver VNH3SP30, including an RS232 connection and a programmable PID controller. The omnidirectional wheels were specially designed to reduce the robot vibration, by a suitable design of the small rolling drums placed along the border of the main wheel.

The kicker, based on an electro-mechanical design, has a control board designed by the Portuguese CAMBADA team to control the charge/discharge cycles of the capacitor, connected to a pair of coils that induce the movement on a metal piston. The kicker control board allows selecting different applied forces during use. Recently we developed and added a new mechanism to the kicker, allowing to select between high or low kicks. This selection mechanism, plus the force control, enables independent height and force control, providing a wide range of possible kicks, which can go from kicking to the upper corner of the goal, to a pass to a nearby teammate.

To ease the ball control and reception, a front drum acts both passively when a ball is received by the robot, by damping its motion, and actively, by rolling the drum at constant speed over the ball to keep it locked with the robot.

Each robot is additionally endowed with the following sensors:

- 1 AVT Marlin F-033C firewire camera. The camera is part of an omnidirectional catadioptric vision sensor, similar to the one used in the old robots [3]. The mirror was re-designed to obtain a bird's eye-view ranging from 250 to 7000mm away from the robot center. The mirror radius is 50mm, and the lens image plan is located 600mm above the ground. The bottom point of the mirror is located approximately 120mm above the image plan.
- 16 sonars (SRF04 RangeFinder) disposed in a ring around the robot.
- 1 500 CPR encoder per motor for motor control and odometry.
- 1 AnalogDevices rate-gyro XRS300EB to improve orientation determination.
- 1 Sharp infrared sensors to detect the ball when it is between the robot fingers.

To power the electronics and motors, 2 packs of 9Ah NiMH batteries per robot are now used, with a significant gain in autonomy/weight ratio. IdMind has developed special chargers for these batteries, which allow charging the robots in maximum 3 hours, with the batteries in place, as well as running the batteries from DC current with a cable.

"Plug-and-play" connections of most peripherals to the laptop, where decision-making, guidance and navigation algorithms are running, were used. The lower level electronics is mostly based in PIC microcontrollers, which manage the interface between the laptop and the several available devices (sensors and actuators).

2.2 Software

Version 1.0 of MeRMaID (Multiple-Robot Middleware for Intelligent Decision-making), the new software architecture of the SocRob project, depicted in Fig. 2, was recently released [4]. This architecture keeps the fundamental principles of the functional architecture the group has been following since 1999 [5]. Nevertheless, it was completely re-designed to be re-written in C++, as well as to improve the matching between software modules and the main functional concepts. We are currently working on finishing MeRMaID 2.0, which will provide a more clear API and increase the OS independence, allowing the architecture to run on the Four-Legged League team and also ease its use on the old Nomadic team. The MeRMaID middleware is based on the *Active Object* pattern. Active Objects are objects that decouple method execution from method invocation in order to simplify synchronized access to an object that resides in its own thread of control. These objects retain their own execution context and execution flow. All processing is done within this context.

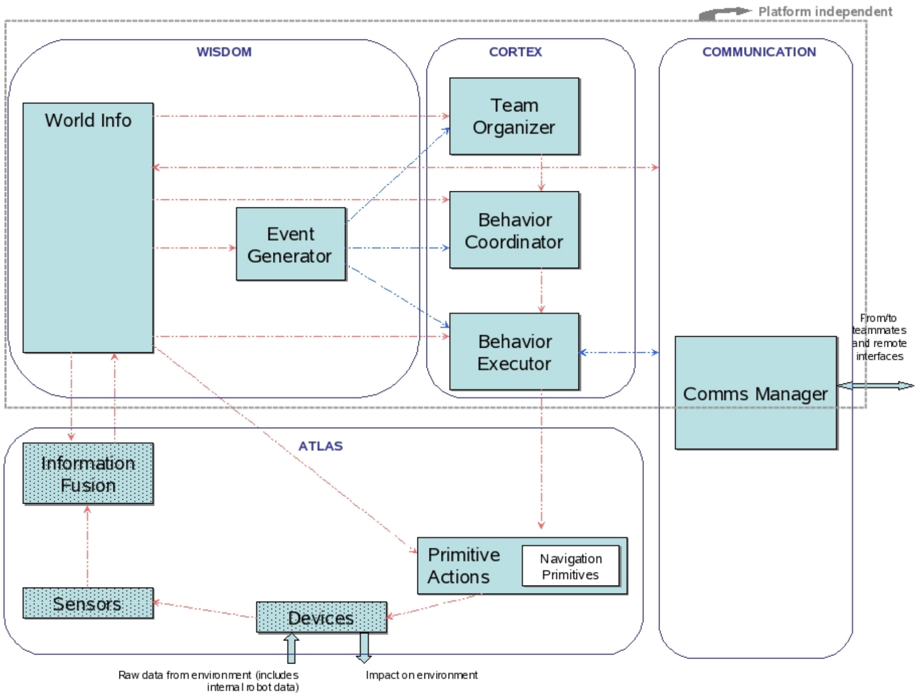

Fig. 2. Block diagram of the MeRMaID functional architecture. The ▢ block denotes main modules, ▬ denotes active objects, ▨ denotes multiple active objects, and ▭ denotes general objects (in object-oriented programming conventions). The ⇢ indicates data flow, while ⇢ indicates events flow.

The main modules available on MeRMaID (see Fig. 2) are *Atlas*, *Communication*, *Wisdom* and *Cortex*. A brief description of each of these modules follows.

The Atlas module is where the interaction with the world occurs. All direct sensing and acting activity is performed within Atlas, perceiving and producing effects on world.

Communication with other robots, external interfaces and the referee box takes place through the Communication module. Other modules connect to the Communication model to access the content of the exchanged messages.

The Wisdom module is where the relevant information about the world, such as robot postures, ball position or current score, is kept and managed. This information can be obtained either by sensor information or messages received from teammates or the referee box (through the Communication module).

The Cortex is where the decision process takes place, based on information retrieved from the Wisdom module. The Cortex connects to the Communication module to communicate with other robots, in order to perform cooperative behaviors.

3 Addressed Research and Future Challenges

From the very beginning of the project, one main concern has been the development of behavior coordination and modelling methods which support our integrated view to the design of a multi-robot population, not necessarily for playing robot soccer. Behaviors can be classified into three distinct groups according to the relations among robots they imply [6]:

Organizational Those which are related to the teams organization, that is, the assignment of roles to players.

Relational Those involving two or more teammates, such as performing a pass or coordinately defending the goal.

Individual Those which are executed by one single robot.

Behaviors are externally displayed and emerge from the application of certain operators. From the operators standpoint, the architecture has three levels, clearly mapped in the software architecture of Fig. 2:

Team Organisation where, based on the current world model, a strategy is established, including a goal for the team. This level considers issues such as modelling the opponents behavior to plan a new strategy. Strategies simply consist of enabling a given subset of the behaviors at each robot by assigning a *role*, e.g., if a player has the role of defender, it will only be authorized to run defensive behaviors. Both *data* and *events* can be used at this level to make decisions. Data corresponds to information retrieved from sensors, which was post-processed using various algorithms (e.g. Kalman, Monte Carlo). Events are defined as occurring when a change of (logical conditions over) predicate values (from `TRUE` to `FALSE` or `FALSE` to `TRUE`) takes place, e.g., event `lost_ball` occurs when predicate `has(ball)` value changes from `TRUE` to `FALSE`.

Behavior Coordination where switching among behaviors, both individual and relational, occurs so as to coordinate behavior/task execution at each robot towards achieving the team goal, effectively establishing the team *tactics*. In the past years we have used finite state automata and logic based decisions and, more recently, we implemented a Petri net [7] based framework and Fuzzy decision-making [8]. Both data and events can be used to select behaviors.

Behavior Execution where behaviors are executed. Since a behavior is formed by linked *primitive actions*, this modules selects primitive actions within the executing behavior, according to data and events provided by the *Wisdom* module. A Primitive action is the atomic element of a behavior, which can not be further decomposed, usually consisting of some calculations (e.g., determination of the desired posture) plus a call to a guidance algorithm or the direct activation of an actuator (e.g., the kicker). If the behavior being executed is a relational one, there can be explicit or implicit communication. If the communications is implicit, it means it is done through the sensors (e.g. vision), if the communication is explicit, it is done through the *Communications Manager*. Currently, each behavior is defined either by a finite state automaton or a Petri net. In the finite state automaton, each state corresponds to the execution of a primitive action and transitions are associated with events. In the Petri net case, places can represent execution of primitive actions, predicates, or just short-term memory. Transitions are used to represent events, test predicate values and combine actions.

A Discrete Event Systems [1] based approach has been followed to the modelling of behaviors and their coordination. Individual behaviors are modelled by finite state automata (FSA), and some work has been done on optimal task planning by composing primitive actions into FSA, given the uncertainty associated to the action effects and the goal of minimizing the time to a goal [9]. Due to their capability to model concurrency, Petri nets are specially suited for modelling relational behaviors, as they can capture the concurrent nature of exchanging messages between teammates while each of them is executing their own primitive actions. Petri net models have been developed based on concepts borrowed from Joint Commitment Theory [10][11].

Sensor fusion methods for world modelling were also introduced by our team in [12]. The goal is to maintain and update over time information on the relevant objects, such as ball position and velocity, teammates pose and velocity, opponents pose and velocity, or position of the goals with respect to the robot. Such information is obtained by each robot from the observations of its internal sensors and then fused among all the team robots, using a Bayesian approach to sensor fusion.

Metric navigation has been one of our major research topics. Using an omnidirectional catadioptric vision system which preserves the Euclidean norm for ground-plane views, we developed a method to estimate the robots posture on the field from single shot images from the omnidirectional vision system. This estimate, plus odometry and rate-gyro readings are used on a Monte-Carlo or Kalman Filter based self-localization [13].

In the field of computer vision, we are working on estimating the 3D position of the ball on the field using a single omnidirectional camera, by using the camera plus mirror model and a model of the ball. We are also using this work to improve the detection of teammates and opponents.

An alternative distributed decision-making architecture supported on a logical approach to modelling dynamical systems, which is based on situation calculus [14], was also introduced by the team in [15]. This architecture includes two main modules: i) a *basic logic decision unit*, and ii) an *advanced logic decision unit*. Both run in parallel; the former intends to quickly suggest, using simple logical decision rules, the next behavior to be executed, whereas the latter uses more sophisticated reasoning tools (e.g., situation calculus) capable of planning, learning and decision-making, both for individual and cooperative (teamwork) situations. This configures an hybrid architecture where the basic (reactive) unit only controls the robot if the advanced (deliberative) unit takes too long to make a decision, assuming a situation urgency evaluation. A partial implementation of this architecture, the basic logic decision unit, was already performed using Prolog. In 2003, this approach was used to select behaviors at the Behavior Coordination level. Its modelling convenience allowed the quick development of different roles for field players (attacker vs defender), as well as *dynamic role change* between field players (defenders switch with attackers, depending on who is in a better position to get the ball).

One recurrent discussion among the team concerns the usage of self-localization. Currently we have the option to use relative or global measurements according to the context. For single robot actions, like grabbing the ball or dribbling to the goal, is usually preferable to use relative measurements, e.g., distance to ball or goals, while when considering multiple robots it is usefull to use global coordinates, at least for relational decision making. A typical case where global coordinates are useful is when distributing the robots by field zones, or when deciding to pass to zones where we know that there is a well positioned teammate and less opponent robots.

As mentioned previously, ongoing work concerns using the MeRMaID middleware on the Four-Legged League as well. Given the similarity between the scenario on both leagues, having the same architecture will further allow the use of many common behaviors and planning algorithms at the Cortex level. Overall development time will be decreased and the various algorithms will have an increased testing time. This work also paves the way to easily build and manage teams of heterogeneous robots. The work currently under development to improve the hardware capabilities of our old Nomadic robots will increase the number and heterogeneity of the robots used in this project, specially considering the different mobility characteristics of the Nomadic robots.

Other research challenges we are currently investigating concern using reinforcement learning to adjust parameters of our discrete event task models, stochastic satisfiability methods to improve the efficiency of searching for optimal plans using discrete event models, modelling the behavior of opponents using FSA, so as to use this in game-theory based decision optimization [16], and formation control methods to dynamically distribute the team robots across the field.

Acknowledgments. This work was partially supported by Fundação para a Ciência e a Tecnologia (ISR/IST plurianual funding) through the POS_Conhecimento Program that includes FEDER funds. Further support was given by the project “SocRob” (POSI/ROBO/43900/2002) of the Fundação para a Ciência e Tecnologia and POSI, in the frame of the QCA, and by the Infaimon and BASE2 companies.

References

1. Cassandras, C.G., Lafortune, S.: Introduction to Discrete Event Systems. Kluwer Academic Publishers (1999)
2. Ferber, J.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley Longman Publishing Co., Inc. (1999)
3. Lima, P.U., Bonarini, A., Machado, C., Marchese, F.M., Marques, C.F., Ribeiro, F., Sorrenti, D.G.: Omni-directional catadioptric vision for soccer robots. Robotics and Autonomous Systems **36**(2-3) (2001) 87–102
4. Lima, P.U., Ramos, N., Barbosa, M., Costelha, H.F.: MeRMaID - Multiple-Robot Middleware for Intelligent Decision-making. Technical Report RT-701-07 (2007) ISR-IST Internal Report.
5. Lima, P.U., Ventura, R.M., Aparício, P., Custódio, L.: A Functional Architecture for a Team of Fully Autonomous Cooperative Robots. In: Proceedings of RobCup-99: Robot Soccer World Cup III, Springer-Verlag (1999) 378–389
6. Drogoul, A., Collinot, A.: Applying an Agent-Oriented Methodology to the Design of Artificial Organizations: A Case Study in Robotic Soccer. Autonomous Agents and Multi-Agent Systems **1**(1) (1998) 113–129
7. Girault, C., Valk, R.: Petri Nets for Systems Engineering. Springer (2003)
8. Ramos, N., Lima, P.U.: Robot behavior coordination based on fuzzy decision-making. In: ROBÓTICA 2006 - 6th Portuguese Robotics Festival. (2006)
9. Damas, B.D., Lima, P.U.: Stochastic Discrete Event Model of a Multi-Robot Team Playing an Adversarial Game. In: Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles. (2004)
10. Cohen, P.R., Levesque, H.J.: Teamwork. Noûs **25**(4) (1991) 487–512
11. van der Vecht, B., Lima, P.U.: Formulation and Implementation of Relational Behaviours for Multi-robot Cooperative Systems. In: Proceedings of RoboCup-2004: Robot Soccer World Cup VIII, Springer-Verlag (2005) 516–523
12. Marcelino, P., Lima, P.U.: Bayesian Sensor Fusion for Cooperative Object Localization and World Modelling. In: Proceedings of 8th Conference on Intelligent Autonomous Systems. (2004)
13. Cabecinhas, D., Nascimento, J., Ferreira, J., Rosa, P., Lima, P.U.: Self-localization based on kalman filter and monte carlo fusion of odometry and visual information. In: Proceedings of ROBÓTICA 2006 - 6th Portuguese Robotics Festival. (2006)
14. Reiter, R.: Knowledge in Action : logical foundations for specifying and implementing dynamical systems. MIT Press (2001)
15. Pires, V., Arroz, M., Custódio, L.: Logic Based Hybrid Decision System for a Multi-robot Team. In: Proceedings of the 8th Conference on Intelligent Autonomous Systems. (2004)
16. Neto, G.F., Lima, P.U.: Minimax Value Iteration Applied to Robotic Soccer. In: Proceedings of the IEEE ICRA 2005 Workshop on Cooperative Robotics. (2005)