Cognitive Robotics in a Soccer Game Domain: a Proposal for the E-League Competition

Alejandro J. García, Gerardo I. Simari, Telma Delladio, Diego R. García, Mariano Tucat, Nicolás D. Rotstein, Fernando A. Martín, Sebastián Gottifredi

Grupo de Robótica Cognitiva
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA) [1]
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur. Av. Alem 1253, (8000) Bahía Blanca, Argentina
Tel: ++54 291 4595135 - Fax: ++54 291 4595136
{ajg,gis,td,drg,mt,ndr,fam,sg}@cs.uns.edu.ar

1 Introduction

In this work, we will discuss the design of a team of robots to play soccer at RoboCup E-League [7, 6, 1, 8]. This task is being carried out in the Cognitive Robotics group of the Laboratory of Research and Development in Artificial Intellgence (LIDIA), Department of Computer Science and Engineering, Universidad Nacional del Sur. The RoboCup competition provides a great opportunity to develop a multi-agent system in which we can test and apply new ideas and results. In the following sections, we will briefly describe the league in which we will participate, and our proposal for the implementation of a team.

2 Robocup E-League

The E-League is part of RoboCup Soccer, which provides an entry point for teams that do not have the resources to compete in Small Size or higher leagues. This league's setup is based on the Small Size League, and includes common vision and communication software. Figure 1 shows E-League's support for vision and communication. A video camera overlooks the field, capturing images that are sent to a video server called *Doraemon* [4]. This program will process the images, extracting information about each robot and the ball. This information is condensed in a package with the following format:

0	No. obj.	Frame no.	Time diff.					
1	CamX	CamY	CamZ					
2	Type	Name	Found?	PosX	PosY	PosZ	Orient.	VelX	VelY
⋮	...	...	...	...	...	...	...	...	...
10	Type	Name	Found?	PosX	PosY	PosZ	Orient.	VelX	VelY

[1] Member of the IICyTI (Instituto de Investigación en Ciencia y Tecnología Informática).

Line 0 contains the number of objects being identified by the video server, the current frame number, and the time difference between the last two frames. Line 1 contains the (x, y, z) coordinates of the camera with respect to the field. Afterwards, the following nine lines contain information about each of the robots, and the ball. Each of these lines contains the following information:

- Object type: ball, robot, etc.
- Object name: "*ball*" for the ball, and an identifier for each robot.
- Whether the object was found by the video server or not. In case it wasn't found, the server will predict the information for this frame.
- The (x, y, z) coordinates of the object.
- The object's orientation, in radians.
- The object's velocity with respect to the x and y directions.

This information is then processed by each of the two teams' agents (see Figure 1), which will be running on computers which we will call *PC Shuttles* (one for each team). Communication follows a client/server model. Therefore, the agents are clients which receive the video server's packages by means of the league's local network. In Figure 1, this can be seen in the connection from *Doraemon* to *PC Shuttle*. This completes the "vision path", which carries perceptual information from the environment to the agents.

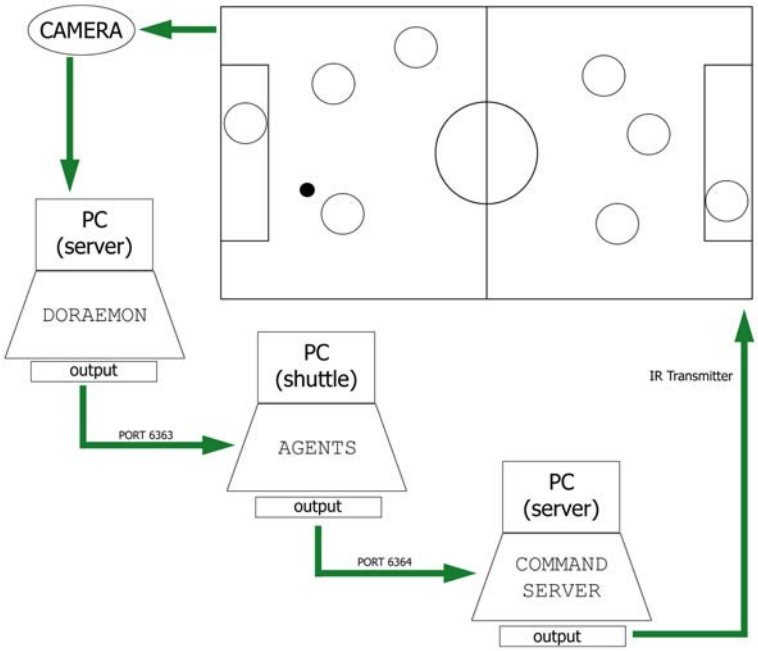

Figure 1: E-League setup

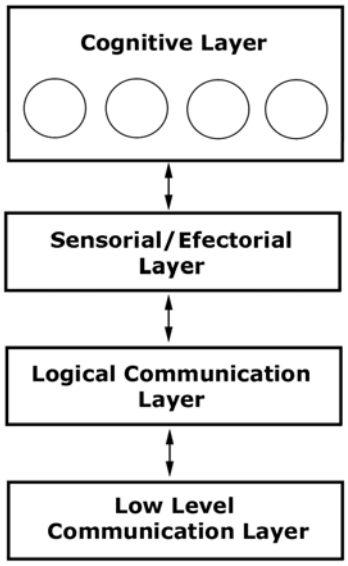

Figure 2: A proposed architecture for the implementation of an E-League team.

The "effectoric path" also responds to a client/server model, in which the teams are clients of the Command Server. At every time step, the Command Server gathers commands from each of the teams which are sent to the robots on the field. The teams send commands of the form "`[name]:[msg]\n`", where `name` is the robot's name, and `msg` is the message sent to it. The Command Server has a command buffer for each robot, which is updated as messages arrive. Periodically, the Command Server will send these messages to the robots by means of one or more infra-red transmitters directed towards the field. Each robot is capable of receiving these messages, and obtaining the command that is directed to him.

3 Architecture for Team Implementation

We are currently working on the implementation of a team for competing in Robocup E-League 2004 [3]. The organization of this implementation is being carried out in layers, each of which offers support at a given level of abstraction. The objective of this model is to modularize the solution to this problem, isolating the different aspects that must be solved by means of these layers. For instance, the problem of communicating with the vision server should not be intertwined with the task of reasoning about the best strategy for scoring a goal.

In the following, we will describe our approach to this separation into layers (see Figure 2):

- **Low Level Communication Layer:** This layer contains the low level communication support provided by the league. This includes infra-red transmitters, video camera, network support, and the software that is necessary for processing visual information (Doraemon) and processing commands that will be sent to the field (Command Server).

- **Logical Communication Layer:** Composed by a set of routines that interface the Low Level Communication Layer with the higher layers. They provide the necessary services that enable communication with the servers.
- **Sensorial/Efectorial Layer (SEL):** In this layer, raw visual information is processed and translated into environment *states* that reflect a higher level of abstraction. For example, given a set of coordinates and speed for each of the robots and the ball, this information could be translated into *"robot* `r3` *is in position to kick the ball"*. Another function implemented by this layer involves the translation of actions selected by the agents into packets that will be sent to the Command Server.
- **Cognitive Layer:** This layer models the *behavior* of the team of robots. It contains the implementation of the agents that represent the team. These agents perceive the environment only through the information offered by the SEL. Different alternatives are possible in this layer. For example, one of the choices involves implementing the team as a *society* of agents, or as a single agent that controls all of the robots. In the case of a multi-agent model, decisions concerning communication among the agents must be made (direct communication through messages, blackboards, multi-agent cooperation protocols, etc).

3.1 Robotic Equipment

Our team is composed of four robots, built with Lego Mindstorms^{TM} [5]. The league only imposes restrictions on processor speed and memory size, and this robotic kit falls within these restrictions with a speed of 10MHz and 32KB of RAM. This equipment can be loaded with a variety of firmwares, which determine the programming language that can be used. In this case, we will use BrickOS [2] and the C programmig language for the program that will be running on the robots' processors. This should not be mistaken with the programs that correspond to the *agents*, as we will see in the following section.

3.2 Implementation of the Cognitive Layer

The implementation of the Cognitive Layer is currently under development, and it is composed by logic-based agents that will be implemented using Prolog. This approach has a series of advantages, such as declarative programming, which allows a more or less direct translation from behavior models to the code that will be directly executed by the agents.

In this proposal, each agent is implemented by a Prolog program. Based on the environment information provided by the SEL, these agents will reason and select the next action that will be executed. This decision is communicated to the SEL, which will build a packet that will be received by the Command Server via the Logical Communication Layer. Finally, the Command Server will gather these packets and sends them to the robots on the field.

Another design issue that must be resolved is the *basic set* of actions that will be available to the agents in order to carry out their plans. An important aspect of this decision is the tradeoff

between the *length* of a given basic action and its *reliability*. That is, a long basic action avoids excessive sensing and deliberation, but is more prone to failure (for example, due to slipping, collision, etc). Therefore, we need to decide which actions will be directly implemented in the robots' code (which will be activated by messages), and which actions are left as decisions to be made by the agents as part of their plans.

4 Conclusions

In this work, we have described a proposal for the implementation of a team of robots that will be participating in the RoboCup E-League competition. We regard this as an opportunity to build a multi-agent system in which new research ideas and developments can be tested, such as planning, knowledge representation, argumentation, and multi-agent cooperation mechanisms.

Acknowledgements

We are very grateful to the *Association of Computing Machinery* (ACM) for granting us a scholarship that will allow us to cover most of the costs of participating this year in the competition in Lisbon, and to Dr. Elizabeth Sklar of the University of Columbia in New York, for making this possible. We are also grateful to the Department of Computer Science and Engineering, Universidad Nacional del Sur, for making its space and resources available to us at any time. The project is also partially supported by the *Agencia Nacional de Promoción Científica y Tecnológica* (PICT 13096).

Finally, Gerardo Simari is partially supported by *Comisión de Investigaciones Científicas*, (CIC) Gobierno de la Provincia de Buenos Aires. Telma Delladio is partially supported by *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET).

References

[1] http://agents.cs.columbia.edu/eleague/.

[2] http://brickos.sourceforge.net/.

[3] http://cs.uns.edu.ar/∼gis/robocup-tdp.htm.

[4] http://sourceforge.net/projects/robocup-video.

[5] http://www.legomindstorms.com.

[6] http://www.robocup2004.pt.

[7] http://www.robocup.org.

[8] Anderson, J., Baltes, J., Livingston, D., and Sklar, E. Toward an undergraduate league for robocup. In *Proceedings of the RoboCup Symposium* (2003).