MECATEAM'2007: A COGNITIVE MULTI-AGENT SYSTEM APPROACH FOR A DISTRIBUTED MULTI-ROBOT SYSTEM CONTROL

Augusto Loureiro da Costa*, Orivaldo Vieira Santana Jr.*, Cleber Pinelli*, Timoteo Salles *

*Programa de Pós-graduação em Mecatrônica
Universidade Federal da Bahia
Salvador, Bahia, Brasil

†Departamento de Ciência da Computação
Universidade Federal da Bahia
Salvador, Bahia, Brasil

Emails: augusto.loureiro@ufba.br, orivajr@dcc.ufba.br, cleberpinelli@yahoo.com.br, timoteo@dcc.ufba.br

Abstract— A Cognitive Multi-Agent System approach for multi-robot system distributed control is the main topic in our reserch group project. In this sense the MecaTeam'2007 brings up some implementations issues under this multi-agent system development for Soccer Simulation 2D, like a new implementation for the agent architecture used by the MecaTeam based on a multi-thread approach. This new implementation presented a significative computing load reduction, then the old implementation, and kept the decision level model and the concurrency among these three decision levels. Aditionally a methodology based on Petri Net to model agents knowledge is used in the MecaTeam'2007 implementation.

Keywords— Multi-Agent System, Multi-Robot System, Collective Robotics

1 Intoduction

The Mobile Robots and the Autonomous Agents definitions converge to the the following points: both of them are immerged into a given environment, which they can percept and mainly they can act in this environment. This actions are genereted by a control system, according to their perception to achieve a goal. *The perception system, the action systems and the control system*, together can be called Autonomous Agent. These Autonomous Agents allows the mobile robot to have autonomy in the to sense which action should be performed in the environment.

Autonomous agents have a high degree of self determination, they can decide by themselves, when and under which conditions an action should be performed. This actions can be chosen just like a *stimulus-response* behavior. Agents who behaves in this way are called *Reactive Agents*. On the other hand, *Cognitive Agents*, are more complex, and can hold an actualized environment model, and an automatic reasoning mechanism which allow the agent to choose their desired goals, and coordinated actions to be performed by the agent into the environment looking for this goals satisfaction. The Cognitive Agents also presents some social skill which allow these agent to interact with another agents to joint to an agent society to perform a collective task, called Multi-Agent System.

The Multi-Agent System presented here allow a set eleven mobile robots behaves like a soccer team in the RoboCup 2D Soccer Server Simulation League (Kitano et al., 1997). This Multi-Agent System presents a new implementation for Concurrent Autonomous Agent architecture (Costa and Bittencourt, 1999; **?**), when he three process for the three decision levels concurrent architecture was replaced by a multi-thread approach, and the lowest level from the original architecture, called Reactive level, was replaced for UvaTri-Learn'2003 basic skills. The paper is organized according the following sequence: section 2 presents the agent architecture; section 3 presents the changes brought by the new implementation and briefly presents a methodogy, based on Petry-Nets, used to model the agent behavior and its interaction with the oder agents and help the agent knowledge implementation; finally 4 presents conclusion and futures works.

2 The Concurrent Autonomous Agent

The Concurrent Autonomous Agent is Based on the Generic Model for Cognitive Agents (Bittencourt, 1997), it implements an autonomous agent architecture with three decision levels, *Reactive, Instinctive and Cognitive*, according to a concurrent approach. Each decision level is implemented in a different process: *Interface*, *Coordinator* and *Expert*.

The Reactive level encapsulates the Basic Player skill package released by UvaTrilear 2003 (Boer and Kok, 2001). It contains a collection of reactive behaviors responsible for some agent's individual skills like: intercepting the ball, kicking the ball to a desire position in the field, moving to a desire position, etc. It also provides the agent environment synchronization and a simple

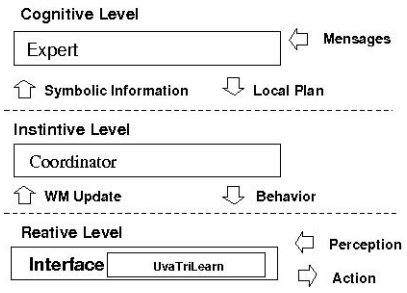

Figure 1: The Concurrent Autonomous Agent.

World Model updated by the perception information sent by the Soccer Server. Every time that a perception is received from the Soccer Server, the reactive level updates the *World Model*, sends a message to the Instinctive level informing the new *World Model* state, and uses the current behavior to perform the necessary skills to handle the agent actions. This is done by a set of commands sent to the Soccer Server to be applied to the robot: *kick, dash, turn, catch, etc.* The current behavior is chosen by the Instinctive level.

The Instinctive level encapsulates a Knowledge-Based System with a one cycle Inference Engine, a rule base and a fact base. The Fact Base stores the information sent by the Reactive level about the *World Model* state. The Rule base is organized into a finite number of rule sets. Each plan is associated with a different rule set. This rule set contains the knowledge to identify the current environment state and to choose the best behavior to satisfy the current plan. In the case that the chosen behavior is different from the currently active behavior in the Reactive Level, a message is sent to the Reactive Level informing the new chosen behavior. The Instinctive level is also responsible by the generation of symbolic information that is sent to Cognitive level.

The cognitive level also encapsulates a knowledge-based system, but one with a multiple cycle inference engine, that also includes a fact base, a local rule base, a social rule base and a Plan Set base. The local rule base is responsible for handling the symbolic information sent by the Instinctive level, building with it a logical model about the environment. Using this logic model, it chooses a local goal; verifies whether the current plan is still valid; chooses the most appropriate plan and sends it to the instinctive level. The social rules contain the knowledge needed by the agent to take part in a cooperation process. The cognitive level also stores the *Plan Set* that is presented at subsection 2.1.

2.1 Plan Set

Autonomous Agents and Multi-Agent Systems often admit different plans that satisfy the same goal g_i, possibly with different satisfaction degrees. In this situation there are different plans that allow the multi-agent system to perform the same cooperative task. Depending on the available agents and on their respective internal states and skills to assume the tasks that are necessary to satisfy the plan, these plans can present different satisfaction degrees. With the aim of facilitating the choice of the best plan by each autonomous agent, according to the current environment state, the available agents, their respective skills and the internal states of the other agents, a data structure called *Plan Set* was proposed.

- **Definition 1 :** A *Plan Set* P_i is a data structure that contain: an identification string *planset–id*; a goal g_i; and a list of plans $L = \{p_1, p_2, ..., p_w\}$ that can satisfy g_i, ranked according to the optimality criteria where p_1 is the optimal plan to achieve g_i.

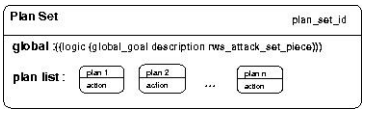

Figure 2: Planset for the global goal right wing side attack set piece.

A *Plan Set* P_i stores in a plan list L all known plans $p_1, p_2, ..., p_w$ to achieve the goal g_i. This goal can be either a local goal or a global goal. For local goals, the plan contains a combination of the agent's behaviors, that once performed successfully by the agent drives the environment to the desired state describe by the goal g_i. On the other hand, for global goals, the plan contains a set of local goals, that once performed successfully by some selected agents in the society that present the needed skills, drive the environment to the desired state describe by the goal g_i.

For a given cooperative task, where the environment assumes the *state* A at instant t_0, the optimal plan p_1 can drive the game from the initial state A to the desired state B in $t_0 + \Delta_1 t$. According to the plan p_1, one player should drive the ball through the field right wing side (task j_1), two others players should move to the main area entrance (tasks j_2 and j_3) and two more players should approach the penalty, (tasks j_4 and j_5). This plan consists of the five cooperative tasks – $J_{p_1} = j_1, j_2, j_3, j_4, j_5$ – represented in figure 3 and should be performed by a set of five players.

The relationship among the necessary tasks $j_1, j_2, ..., j_5$ associated with plan p_i for the goal g_i, that can drive the environment from an initial state $S(t_0)$ at instant t_0 to a desired state $S(t_0 +$

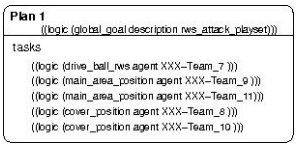

Figure 3: The optimal plan p_1 for g_i.

$\Delta_1 t$) at instant $t_0 + \Delta_1 t$, can be represented by the Petri Net shown at figure 4.

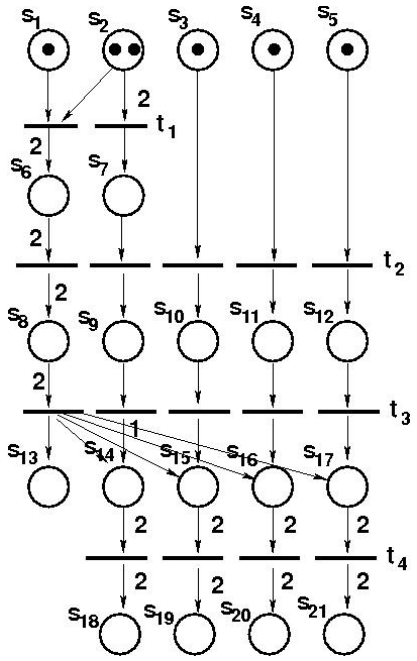

Figure 4: A Petri Net for the optimal plan p_1.

The arcs labeled with number 2 have an associated weight of 2, the other arcs, that are not labeled, have an associated weight of 1, the default value. The places s_1, s_2, s_3, s_4, s_5 represent the scenario depicted by *State A* : s_1 represents the robot, $Player_7$, located at the attack right midfield position waiting for the ball passing; s_2 the robot, $Player_8$, located at the defense center-right midfield position with ball control; s_3 the robot, $Player_9$, located at the attack center midfield position and waiting for the ball control under $Player_7$; s_4 the robot, $Player_{10}$, located at the defense center-left midfield position and waiting for the ball control under $Player_7$; s_5 the robot, $Player_{11}$, located at the attack left midfield position and waiting for the ball control under $Player_7$.

One token is associated with each agent and another token is associated with the ball. The places s_1, s_3, s_4, s_5 initially hold one token each and place s_2 holds two tokens because at the initial state this robot has the ball control. Transition t_1 represents a ball passing from $Player_8$ to $Player_7$. The places s_6 and s_7 represent the $Player_7$ and $Player_8$ that are waiting for the ball control under $Player_7$. At transition t_2, $Player_7$ assumes the ball control. After transition t_2, places $s_8, s_9, s_{10}, s_{11}, s_{12}$ become active, they represent: s_8 $Player_7$ driving the ball through the right wing side until close to the penalty area; s_9 and s_{11} $Player_8$ and $Player_{10}$ move to the penalty area entrance; s_{10} and s_{12} $Player_9$ and $Player_{11}$ move to the penalty area head.

After transition t_3, $Player_7$ crosses the ball to the main area head. This transition can activate one of the following places $s_{14}, s_{15}, s_{16}, s_{17}$. In other words, one of the players $Player_9$, $Player_{11}$, $Player_8$, $Player_{10}$, the one that gets the ball control, kick it to the goal firing transition t_4. Then one of the places $s_{18}, s_{19}, s_{20}, s_{21}$ will be the Petri Net final place. The Petri Net shown in figure 4 expresses the concurrence and dependence relationships among the tasks involved in plan p_i. This Petri Net can also be used to guide the generation of the respective rule sets that compose the instinctive level and are used to execute the plan p_i.

On the other extreme, there is plan p_4 that also allows the goal g_i to be achieved but using only two cooperative tasks. According to plan p_4, one player drives the ball through the field right hand side (task j_1) and another player moves to the penalty area entrance (task j_2, see figure 5).

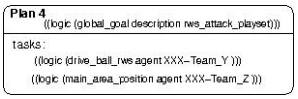

Figure 5: The critical plan p_4 for g_i.

Two others alternative plans, p_2 and p_3, which allow the agent society to achieve intermediate state C in $t_0 + \Delta_2 t$ and state D in $t_0 + \Delta_3 t$ can also satisfy the goal g_i.

The plans p_1, p_2, p_3 and p_4 present different satisfaction degrees for goal g_i. The optimal situation happens when the five agents are available to execute all the tasks in plan p_1, called optimal plan. This plan drives the game from *state A* to *state B* in $t_0 + \Delta_1 t$. On the other hand, the worst situation happens when the agents are able to execute just the two tasks in plan p_4. This plan drives the game to *state E* in $t_0 + \Delta_4 t$.

In other words a $PlanSet$ P_i is a data struture that holds all the plans known by an agent to attain a given goal. This goal can be either a local goal or a global goal. For local goals, the cognitive level uses its rule base and the logical World Model to choose which plan $p_i \in P_i$ will become the active one. Each plan contains a combination of agent's behaviors that once successfully associated by the Instinctive level with the environment states drives it to the goal g_i. For each plan p_i there is an encapsuleted rule set that associates the current environment state $S_i(t)$ to

the agent's behaviors specified by the plan p_i. For global goals, the plans contain tasks that can be associated to agents local goals and once all the local goals in p_i are achieved, the global goal is satisfied. The decision about which plan $p_i \in P_i$ would become active and which agent will assume the necessary task $j_i \in J_{p_i}$ is done through agent society interactions.

3 The Multi-Agent Systems for Robots Soccer Simulation

The Expert-Coop++ library (Costa et al., 2003) was chosen to implement the multi-agent system that controls the MecaTeam'2007 eleven robot.

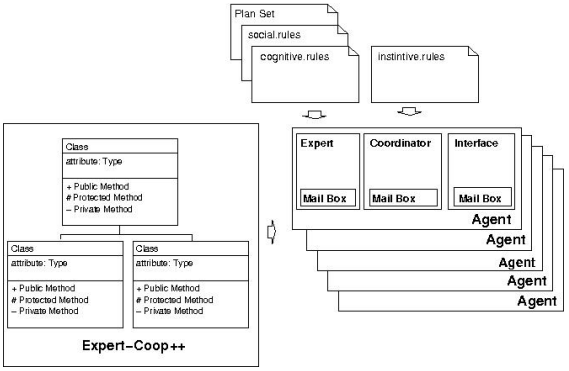

Figure 6: Agents implementation

Expert-Coop++ is a modular object-oriented environment, designed to help the implementation of Cognitive Multi-Agent Systems, under best effort soft real-time requirements. The Expert-Coop++ environment consists of a library of C++ classes that supports the creation of agents according to a predefined agent architecture, the *Concurrent Autonomous Agent* (Costa and Bittencourt, 1999). The Expert-Coop++ environment introduces a new data structure called *Plan Set* presented in subsection 2.1. The environment also offers three different cooperation strategies: Dynamic Social Knowledge (Costa and Bittencourt, 2000), Contract Net Protocol (Smith, 1980), and a slight variation of Coalition Based On Dependence (Ito and Sichman, 2000). For the multi-agent system implementation described in this paper, no agent comunication have been used nether agent cooperations strategies.

4 Conclusion and Future Works

The MecaTeam'2007 presents a multi-agent system approach for a multi-robot system distributed control uther the RoboCup Soccer Simulation League. his implementation brings ups a new approach for the Concurrent Autonomous Agent implementation, based on multi-thread approach who allows a significant reduction of the MecaTeam computing load. Also a mothodology form agent knowledge and Multi-Agent System behavior based on Petry-Net was presented. This methodology helps the agent Knowledge implementation decreasing the complexity to implement the rules bases used by the agents.

References

Bittencourt, G. (1997). In the quest of the missing link, *Proceedings of IJCAI 15, Nagoya, Japan, August 23-29*, Morgan Kaufmann (ISBN 1-55860-480-4), pp. 310–315.

Boer, R. d. and Kok, J. R. (2001). *The incremental development of a synthetic multi-agent system: The uva trilearn 2001 robotic soccer simulation team.*, Master's thesis, University of Amsterdam, Netherlands.

Costa, A. L. d. and Bittencourt, G. (1999). From a concurrent architecture to a concurrent autonomous agents architecture., *IJCAI'99, Third International Workshop in RoboCup* pp. 85–90. Springer, Lecture Notes in Artificial Inteligence.

Costa, A. L. d. and Bittencourt, G. (2000). Dynamic social knowledge: A comparative evaluation., *Intenational Join Conference IBREAMIA'2000 / SBIA'2000.* pp. 176–185. Springer, Lecture Notes in Artificial Inteligence, vol. 1952 - Best Paper Track Award.

Costa, A. L. d., Bittencourt, G., GonÃ§alves, E. M. N. and Silva, L. R. (2003). Expert-coop++: Ambiente para desenvolvimento de sistemas multiagente., *IV ENIA Encontro Nacional de Inteligência Artificial* pp. 597–606. XXIII Congresso da Sociedade Brasileira de Computação.

Ito, M. and Sichman, J. (2000). Dependence based coalition and contract net: A comparative analysis., *Intenational Join Conference IBREAMIA'2000 / SBIA'2000.* pp. 106–115. Spring-Verlag, Lecture Notes in Artificial Inteligence, vol. 1952.

Kitano, H., Tambe, M., Stone, P., Veloso, M., Coradeschi, S., Osawa, E., Matsubara, H., Noda, I. and Asada, M. (1997). The robocup synthetic agent challenge, 97, *International Joint Conference on Artificial Intelligence (IJCAI97)* . Nagoya, Japan.

Smith, R. (1980). The contract net protocol:high-level communication and control in a distributed problem solving, *IEEE Transactions on Computers* **29**(12): 1104–1113.