PolyteCS Team Description
For RoboCup2002

M. Ebrahim. Shiri, Pooyan Fazli, Alireza Davoodi, M. Moin Ayazifar,
Vahid Hashemi and Sara Ramezani

Faculty of Math. and Computer Science
Department of Computer Science

Amirkabir Univ. of Technology (Tehran Polytechnic), 424, Hafez Ave., Tehran 15914,
Iran

http://www.aku.ac.ir/polytecs
email: PolyteCS@aku.ac.ir

Abstract: This paper is to describe the PolyteCS simulation team. PolyteCS is the result of 15 months of research in RoboCup Simulation Research Lab. Our main goal in this project is to develop a multi-agent system based on fast and improved methods of learning and deciding in cooperative planning .We also introduce a new concept to make heterogeneous agents. To achieve this, PolyteCS uses case-based reasoning techniques as its focus.

Keywords: Case-Based Reasoning, Multi-Agent System, Heterogeneous Agent, Graph with Classified Concepts and Relations.

1. Introduction

The Robosoccer simulation is an effective environment to achieve variety of experiences in field of artificial intelligence and multi-agent systems[1].

The main interest behind the PolyteCS's effort in the RoboCup soccer domain is to develop and apply decision and learning techniques in multi-agent systems. In other words we improve an agent–oriented system consists of expert agents. In this domain each agent achieves necessary parameters (*e.g. positioning table, personality and personality inheritance rate*) to choose the best action. Especially we are interested in Case-Based Reasoning (CBR) [2,3], which is a recent approach to problem solving that has got a lot of attention over the last few years [4,5]. We also introduce new memory architecture for case representing and retrieval named GCR .

2. Agent Architecture

PolyteCS's agents are trained to fulfill their special roles. For this purpose, each agent has an exclusive case base different from others' in itself. Our goal is to simulate a real distribution of roles like a real football team.

The PolyteCS's agent is based on architecture with three layers, where the first layer deals with skills and communication with soccer server. The second layer includes agent's case base and C*ase Interpreter Unit (CIU)*. The CIU interprets the cases chosen by decision unit to server commands. The third layer has a decision unit that uses the second layer for its reasoning and decision.

3. GCR Representation in Case Based Structure Agents

Representing different situations of field in a case is the most important part of our agent. In fact the representation problem in PolyteCS is the problem of decision what to store in a case, finding an appropriate structure for describing case contents, and deciding how the case memory should be organized and indexed for effective retrieval and reuse.

For this purpose, we introduce a particular knowledge representation formalism called *Graph with Classified Concepts and Relations (GCR)*, which is an extended model of the conceptual graphs [7] to represent the cases and their solutions.

In [6] we discussed some interesting capabilities of the GCR. We argued that it makes possible to express the relations between the solutions and cases, to express the modifications that are necessary to adopt the previous case to the current situation and goals.

Further, our memory architecture is also structured using this formalism, i.e. within this memory organization, the categories are inter-linked via GCR, which contains the features and intermediate states. (e.g. similar cases).

The memory organization is thus embedded in a network structure of *categories, cases,* and *index pointers.* Each case is associated with a category and each index value (indices) points to a single case or the associated category. At least one of the cases in a category is representative (we refer to them as *typical* cases).

Finding a case that matches the current situation is done however by combining the input features of case into a pointer to the case (or category that shares most of the features). If these features match with the description of its typical cases then these cases are returned, else the indices under that category are then traversed in order to find the case that contains most of the additional features.

Storing a new case is performed in the same way, with the additional process of dynamically creating new node containing the description of the case and jointing it to the category using the GCR formalism. The referent part of the established relation will then represent the similarity between the typical case and the stored one. The new case is stored thus, by searching for a matching case, and by establishing the appropriate feature connection. If the feature of the new case matches the feature of an existing one, the new case may not be retained.

Because robosimulation is a dynamic real-time environment, the privilege of this method is that the agent in restricted time can fast retrieve the best case or cases to find the best solution for decided situation.

4. Roles and Personalities in PolyteCS's Heterogeneous agents

4.1. Personalities and Cases

For each player we consider a main role and some secondary roles where each role can be implemented by personalities based on different parameters such as stamina and so on. The main role, based on the case base stored in each agent, having default personality, which is named *Primary Case Base (PCB).* But often it is necessary that an agent changes its role due to the game situation or changing the team's strategy, etc., and gets an appropriate secondary role. To perform this task, the agent connects to *Universal Case Base (UCB)*, which includes all case bases of different personalities, indexed via roles, and demands the case base of the related role. Each personality related to a specific role, has a dynamic territory. In this territory the player based on different parameters can choose its proper personality to play its role.

In fact our agents do not change their roles and personalities only based on geometrical parameters but also consider some others such as team situation.

4.2. Personality and Inheritance

One of the important points of a team is the capability of players to adapt themselves to the environment. For this purpose we use the inheritance mechanism by assuming some *Virtual Reference Points (VRP)* inside the field.

Each VRP has its own properties which agents can inherit them.

The set of VRPs and their properties, describes team state, until the next event occurs.

$$\text{Team state}=\{(VRP_1, \text{properties}_1), \ldots , (VRP_n, \text{properties}_n)\}$$

A state would change when an event occurs. For example when ball possession or game result changes. Then the VRPs and their properties change according to the event has been occurred.

In this while agents calculate some features such as: team gravity center, the length and width of the team, distribution of players along the length and width of the field, ball possession, and event_ID. According to these features agents choose the best case and refresh the VRPs and properties. When an agent player closes to a VRP, inherits its properties, which affect agent's personality.

We use a characteristic function to determine the inheritance rate from a VRP, some function arguments are, distance from VRP and movement direction of the agent. This function is now developing.

5. Positioning

We use the following approach to dynamically estimate the positions of players in the field. In our team each player keeps the position of all players of the two teams in a table. A control parameter corresponding to the maximum possibility of error for each of these positions is also stored in this table. In a periodic time we refresh each player's positioning table by seeing the position of only a few of the players and estimating the position of the other players using current team strategy.

Each player adjusts its neck_angle and view_angle so that it can see the most players, considering their priority (this priority is calculated using parameters such as error control).

6.Conclusion and Future Works

Our online-coach is in its preliminary stages .We plan to develop it in near future. We also plan to improve our online-learning with focus on opponent modeling.

PolyteCS Members would like to thank CMU99 and F.C.Portugal simulation teams for making available their low-level source codes.

References

[1] Peter Stone, Manuela Veloso, Multiagent Systems: A Survey From Machine Learning Perspective. *Autonomous Robots, Volume 8, Number 3, July 2000*.

[2] M. Lenz, A. Hubner, and M. Kunze. Textual CBR. In M. Lenz, H.-D. Burkhard,B. Bartsch-Sporl, and S. Web editors, Case-Based Reasoning Technology –From Foundations to Applications. *LNAI 1400, Berlin, 1998. Springer Verleg*

[3] Alireza Davoodi, Case Based Reasoning Application. *Third Student Conference of Electrical Engineering, Tehran, Iran, March 2002 (accepted).*

[4] Wendler, J., Lenz, M.: CBR for Dynamic Situation Assessment in an Agent-Oriented Setting. In D. Aha and J. J Daniels, editors, *Proc. AAAI-98 Workshop On Case Based Reasoning Integrations, Madison, USA, 1998*

[5] H-D Burkhard, Helmut Myritz, Gerd Sander, Thomas Meinert. AT Humboldt In RoboCup-99

[6] M. E. Shiri, Graph-Structured Representation for Case-Based Systems. *First International Conference on Graphs, and Computational Algorithms, Tehran, Iran, 2000.*

[7] Sowa, J.F. *Principles of Semantic Network*: Exploration in the Representation of Knowledge. Morgan Kaufmann Publisher Inc., San Mateo CA, 1991.