Caos Online Coach 2006 Team Description

José Antonio Iglesias, Agapito Ledezma and Araceli Sanchis.

Universidad Carlos III de Madrid
Avda. de la Universidad, 30, 28911, Leganés (Madrid), Spain
{jiglesia, ledezma, masm}@inf.uc3m.es

Abstract. This paper describes the main features of the *Caos Coach 2006 Simulation Team*. This Coach focuses on the challenge of the opponent modelling using sequential events of the players, from observations of their main features. Also, it is able to translate observations of a dynamic and complex environment into a time-serie of recognized events. Finally, our coach implements a mechanism to compare different time-series.

1 Introduction

The main goal of the RoboCup Coach Competition is to create an agent (coach) which provides advices to other agents about how to act. About the Coaching problem raised in Coach Competition, Riley et al. [1] present a general description of the coaching problem. This description is the first step in understanding advice-based relationships between automated agents. One of the reasons to consider a coach role in a team of autonomous agents is that a coach role provides a method of oversight for the agents and can aid in the creation of agents with adjustable autonomy [2]. Also, Riley et al. [3] justify that coaching can help teams to improve in simulated robotic soccer domain.

The *RoboCup Coach Competition* changed recently in order to emphasize opponent-modelling approaches. The main goal of this competition is to compare two team behaviours, but in one of them a play pattern[1] (way of playing soccer) has been activated but not in the other one. This is a sub-league for automated coaches which are able to work with a variety of teams through the use of the standard coaching language. This coach can only support its team by giving messages to its player in a standard coach language called *CLang*, which was developed by members of the simulated soccer community [4].

The competition focus is on opponent modelling and online adaptation. The coaches must work both by analyzing logs of previous games and adapting while a game is being played.

[1] In this paper we use the term pattern as a contraction of play pattern

2 Coach Architecture

In this competition, the coach is given a number of game logs and it must model them in order to detect the used patterns in online mode and report them. Hence, the coaches should be looking for the qualitative differences among the pattern log file and the corresponding no-pattern log file to recognize the pattern correctly.

Therefore, this competition requires two phases:

1. **Offline analysis**: Is the first phase of each round and the coaches analyzes the log files of the patterns which will be used during the round.
2. **Online Detection**: The task of the coach in this phase is to detect the pattern(s) activated in the opponent team.

The following two sections describe how *Caos Coach* works on these phases.

3 Offline Analysis

The goal of this section is to extract and store in a useful way the important features from log files. These features must be relevant to classify the opponent team behaviour. *Caos Coach* divides this part into four different sub-parts. Also, in order to obtain the main differences between the pattern and no-pattern log files, these four sub-parts are carry out in the two log files.

The overview structure of the off-line analysis is shown in the Figure 1.

3.1 Features Extraction

In this first stage, the most relevant data of the log files are needed. Hence, the following features are extracted:

1. *Cycle:* Number that enables arranges the events.
2. *Ball Position:* Ball's position is stored as *x and y* axis in a coordinate system.
3. *Teammate and Opponent Positions:* Player's position is stored as *x and y* axis.

3.2 Event Recognition

Once the main features have been obtained, our coach has to infer what events have occurred. It considers mainly, who is the ball possessor every cycle and, according with Kuhlmann et al. work [5], it classifies the next events:

DribbleX (T), HoldX (T), GoalX (T), PassXtoY (T), FoulX (T), StealX (T), MissedShotX (T), InterceptedPassXtoY (T), Clear (T)

Where *X* represents the player who execute the action and *T* represents the team of this player. The event *Clear (T)* represents the events which cannot be classified as any of the other categories.

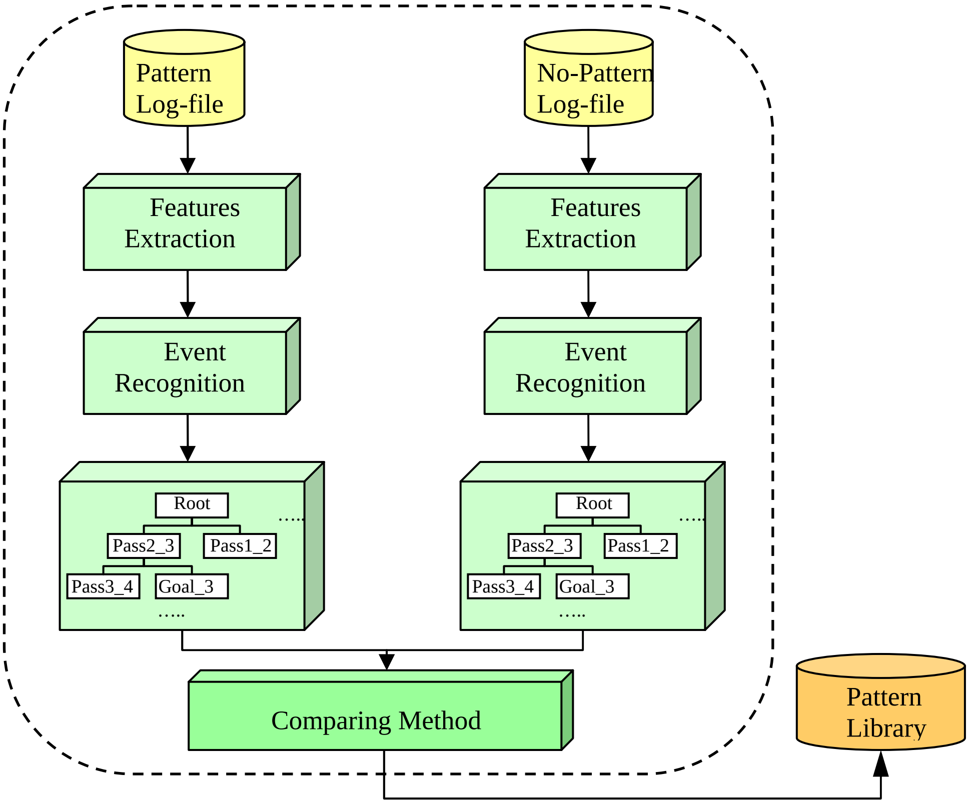

Fig. 1. Off-line Analysis. Overview Structure.

In this phase, always there is some uncertainty inherent in the results because there are events very hardly to identify, even if it is done by a soccer expert.

3.3. Building a trie

In order to get the applied pattern, it must be considered that a pattern describes a simple behaviour that a team performs. As this pattern must be recognizable for the coaches, our Coach considers that the repeating events or behaviour sequences could be related to the activated pattern. Because of this supposition, a trie structure data [6] is proposed to store the results of the above section, as it is used by Kaminka et al. [7] to learn the sequential coordinated behavior of teams.

As we consider two log files (pattern and no-pattern), two tries are built (*pattern-trie and no-pattern-trie*). After building these tries, it is necessary to evaluate dependences. Although there are few methods for discovering significance of sequences and sub-sequences, our coach uses a statistical dependency test [8]. The main idea of this procedure is the proposal that the appearance of repetitive sequences may indicate a pattern. To evaluate the relation between an event and its previous events sequence, one of the most popular statistics is used: Chi-square test [9].

3.4 Comparing Method

In order to compare the two games, *Caos Coach* implements a method to find the important and reliable differences. The input of this process is the result of the previous procedure.

In this procedure, our coach compares two different tries which represent the behaviour of the team in the two games. The result of the comparison is a pattern description as similar as possible to the pattern followed by the pattern-trie. The term *pattern-trie* is used to refer to the trie obtained from the game which followed a pattern, and *no-pattern-trie* terms the trie for the game that did not follow the pattern.

3.4.1 Pattern Description

A pattern defines recurring events to a recurring prefixes. Also, a pattern could consist of a set of simple behaviours. Because of this reason, our pattern description consists of a set of sub-patterns. Our sub-pattern is defined as the possibility measured by a chi-square test value (*chi-sq*) that an event (*event*) occurs after a sequence of events (*prefix*).

Let *OurPDescription* = {p1, p2, ...} be the set of all sub-patterns *sub-*p_i. A sub-pattern is defined as: *sub-*p_i = (*event, prefix, chi-sq*).

3.4.2 Comparing Method

In the comparing algorithm implemented by our coach, a threshold value has to be established for accept or reject the chi-square test hypothesis in the events.

1. If the event of a node is represented in both tries at the same level and their prefix is the same:
   - Chi-square value of both nodes is compared and only if the difference is bigger than the threshold, the node of the *pattern-trie* is stored as a sub-pattern in the pattern description. It means that a different behaviour in the trie has been found (and it is classified as a sub-pattern).
2. If the event and prefix of a node are represented only in the *pattern_trie:*
   - If the chi-square value of the event is bigger than the threshold value, the node is stored as a sub-pattern in the pattern description.

The result of this off-line analysis is a patterns description that is stored in a pattern file.

4 Online Detection

The second phase of our coach architecture is the *On-line Detection*. In this phase, *Caos Coach* connects, in online mode, to the RoboCup Soccer Server. This server sends global *see messages*. The information of these messages is very similar to the information obtained from the log-files. Hence, in order to analyze the behavior of the opponent, the same method used in the off-line analysis phase can be applied.

The information received by our coach from the RoboCup Soccer Server is matched with the patterns obtained from the first phase. Depending of the result of this match process, the pattern is or not recognized.

The basic structure of this online Detection is shown in figure 2.

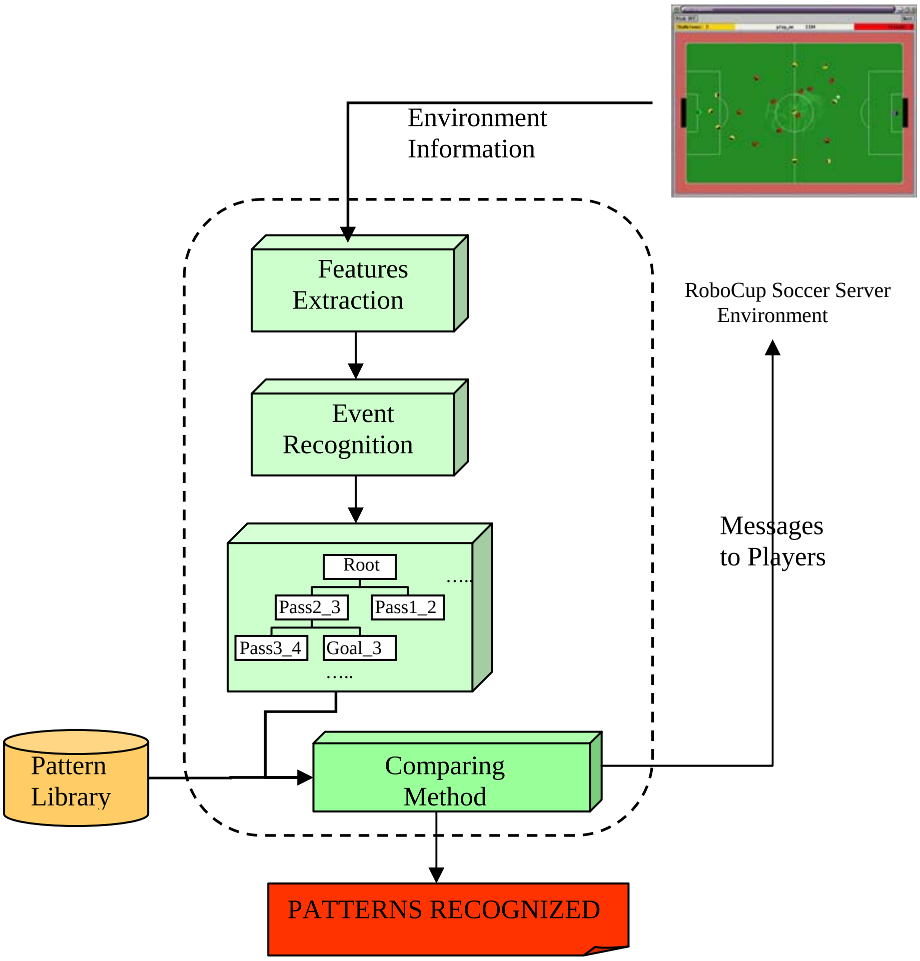

Fig. 2. On-line Detection. Overview Structure.

5 Conclusion and Future Work

The comparing method applied by our coach, works successfully when the pattern followed by a team is related to the players' actions. A simple method is used and the result is very satisfactory.

The different field regions in which the action occurs has not been represented, so if the pattern followed by the team is related to this aspect, our proposed method would not be viable. Also, if the pattern is related to actions that occur when the player is not the ball owner, this method, as well, would not be viable.

This is the first participation for our *Caos Coach Team*, but we plan to improve adding many other different aspects in this first Team.

References

1. P. Riley, M. Veloso & G. Kaminka. Towards any-team coaching in adversarial domains. In Onn Shehory, Thomas R. Ioerger, Julita Vassileva, and John Yen, editors, Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Sytemsi (AAMAS), Bologna, Italy, 2002.
2. P. Scerri, D. Pynadath, & M. Tambe. Adjustable autonomy in real-world multi-agent environments. In Agents-2001, 2001.
3. P. Riley, M. Veloso & G. Kaminka. An Empirical Study of Coaching. In H. Asama, T. Arai, T.Fukada, and T. Hasegawa, editors. Distributed Autonomous Robotic Systems 5, pp 215-224, Springer-Verlag, 2002.
4. M. Chen, E. Foroughi, F. Heintz, Z. Huang, S. Kapetanakis, K. Kostiadis, J. Kummeneje, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang & X. Yin. Soccerserver Manual v7. The RoboCup Federation, 2001.
5. G. Kuhlmann, P. Stone & J. Lallinger. The Champion UT Austin Villa 2003 Simulator Online Coach Team. In Daniel Polani, Brett Browning, Andrea Bonarini, and Kazuo Yoshida, editors. *RoboCup-2003: Robot Soccer World Cup VII*, Springer Verlag, Berlin, 2004
6. Fredkin, E. *Trie Memory***.** Communication of the ACM**.** Vol. 3:9 (Sep 1960). pp. 490-499.
7. G. A. Kaminka, M. Fidanboylu, A. Chang & M. Veloso. Learning the Sequential Coordinated Behavior of Teams from Observations. In Proceedings of the RoboCup-2002 Symposium, Fukuoka, Japan, June 2002
8. Z. Huang, Y.Yang and X.Chen. An approach to plan recognition and retrieval for multi-agent systems. *Proc of AORC*, Sydney, Australia. Jan 2003.
9. Rohlf, F. J. and R. R. Sokal. 1995. Statistical Tables, 3rd ed. W. H. Freeman and Company, New York, NY.