Caspian 2004 Coach Description

Mohammad Nejad Sedaghati, Sina Iravanian, Bardia AghaBeigi,
Arash Alikhani,Mohammad Saeed Tabatabaee, Nina Gholami, Mehrdad Senobari,
Mohammad Jafar Abdi, Ehsan Rafiee, Omid Mehdi Zadeh, Sommayeh Zahedian,
Leila Pakravan Nejad, Hamid Hamraaz, Shams Feyzabaadi, Reza Seyed Khamooshi,
Mohammad Reza Kangavari

mnsedaghat@yahoo.com

Intelligent Systems Lab, Computer Engineering Department
Iran University of Science and Technology, Tehran, Iran

http://caspian.iust.ac.ir

Abstract. The RoboCup Coach Competition mainly deals with one autonomous agent providing advice to another autonomous agent about how to act. The main concern in the design of coach agent is to develop team strategy using CLang rules. Our team has developed an Integrated Visual Environment called Strategy Builder which facilitates team strategy development. The main emphasis of Caspian Simulation Team is developing coachable agents which respond to even complicated CLang rules in reasonable time. To address this challenge we take advantage of a knowledge-base in Horn Normal Form which has a noticeable impact on the agent's general performance. Moreover hierarchical rule classification is introduced which greatly reduces the number of conditions that should be verified in each step.

1. Introduction

The RoboCup Coach Competition mainly deals with one autonomous agent providing advice to another autonomous agent about how to act [1], thus dividing the domain into two (interrelated) research topics: one developing a coach agent, and the other, developing a coachable team. These research topics bring out the following challenges:

1. Mapping strategies into CLang
2. Advising coachable agents
3. Integrating advisable action selection mechanism in player agents

Strategy Builder is an Integrated Visual Environment for developing high-level strategies. The main contribution of Strategy Builder is its easy-to-use interface and automatic compilation of the visually designed strategies into CLang. The primary results show that, generating CLang with a visual language not only facilitates coaching research, but it also enables a human coach expert to implement his knowledge into a coach agent.

There is an initial set of rules, which is the result of visually designed strategies in Strategy Builder, and processing opponent log files, then our coach selects the most suitable set of rules to start the game. During the game the coach may switch between the currently on rules, and another set of available off rules, thus adapting its coached agents to the current game situation.

The coachable agents, on the other side, must have a good understanding of advices, and should perform a wise reaction on critical situations (e.g. when there is a rule conflict). The main emphasis of Caspian RoboCup Soccer Simulation team is on developing a coachable team. The Caspian Coachable Team provides support for the 2004 RoboCup Competition rules, and the standards of communication between coachable agents.

In this paper we first introduce Strategy Builder in more detail, then discuss how our coach advises the coached teammates before the game, and adapts the players during the game. Then we focus on some implementation details of the Caspian Coachable Team. Finally the potential improvements to the current approach are given in Conclusion and Future Work section.

2. Strategy Builder: A New Visual Coaching Environment

The standard coach language (CLang) was developed to enable coach agents to work with teams from different research groups. It presents a clear semantics which prevents misinterpretation from both the player and the coach. But coach developers found it difficult to use, and it leads to a slow coach development procedure [2]. That may be because CLang does not include the concepts of tactics and most of such high-level concepts (mainly with the goal of keeping the language simple). By the way, there are some expert coaches who have valuable knowledge of coaching but do not have the knowledge to translate the tactics into CLang.

Strategy Builder is a research product introduced by our group, Caspian RoboCup Soccer Simulation team. It is a visual environment for designing soccer team strategies. This environment provides a set of icons for designing strategies. After a strategy is designed, a visual compiler may be invoked to translate the strategy into a corresponding set of rules in CLang. Hence, it is observed that our environment, allows the user to interact with a visual abstraction of the coaching process. The motivation behind the design of our coaching environment is to provide the soccer simulation coach developers and human coach experts with, simple yet powerful interface to share their knowledge at some level and to turn Strategy Builder into a rapid strategy development environment. From the coach developers' point of view, it can be an easy to use environment which relieves a coach of having to handle strategy construction using low level CLang rules.

2.1. Visual Support for Rapid Strategy Development

Strategy Builder is a graphical icon-based environment in which each player is represented with a unique icon. Also each action has a graphical representation of its own including icons and different types of arrows. In an attempt to simulate the way in which a human coach expresses strategies, we created a whiteboard which enables strategy planners to draw their target plan by simple drags and drops.
The strategy shown in figure 1 specifies how players should get the ball out of their half and move it towards the opponent penalty area, using the pass routs denoted by the arrows. The designed plan can be saved for future use, in a textual representation called SBLang.

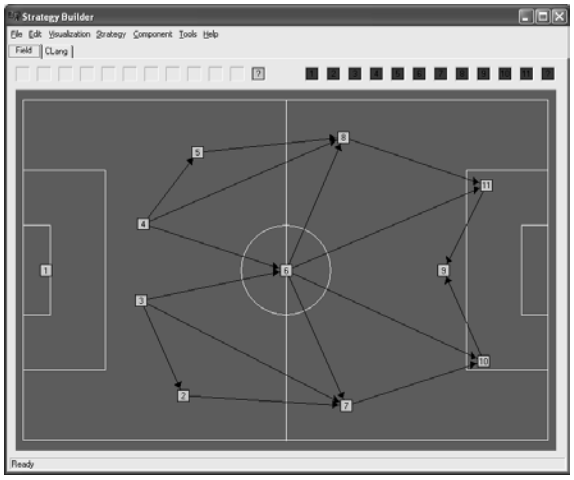


Figure 1. A simple strategy designed by Strategy Builder

2.2. Textual Representation: SBLang

SBLang is our proposed XML format language embedded into Strategy Builder to enable storing and reusing the visually designed strategies. The eXtensible Markup Language (XML) is a standardized notation for documents and other data [3]. An XML document contains, in addition to its content, information about its structure, which is achieved through tagged elements. Noticeably, XML is recognized as a standard format for exchanging data, and the number of applications using XML is rapidly growing.

Listing 1 shows the overall structure of our proposed XML format for defining soccer team strategies. In this XML structure, a team strategy consists of several sub-strategies, each designed for a particular game situation. The team strategy specifications start with the <STRATEGY> tag, followed by a sequence of <SUB_STRATEGY> tags which represent the sub-strategies, and ends with a corresponding terminating tag </STRATEGY>.

Listing 1. the overall structure of SBLang

```
<STRATEGY>

    <SUB_STRATEGY name = "Corner1">

        <COND value = "(playm ck_our)"/>
        <REGION_LIST>…</REGION_LIST>
        <OBJ_POINT_LIST>…</OBJ_POINT_LIST>
        <ACTION_LIST>…</ACTION_LIST>

    </SUB_STRATEGY>

</STRATEGY>
```

Each sub-strategy is distinguished by a tag named <SUB_STRATEGY>. The "name" attribute of this tag identifies the sub-strategy, which can be used in further references. Each sub-strategy is designed for a specific situation which is represented in the <COND> tag by an attribute named "value".

There is a list of user-defined regions which can be referred to in a particular sub-strategy. These regions are listed within <REGION_LIST> and </REGION_LIST> tags. For the players involved in a sub-strategy, the <OBJ_POINT_LIST> is considered which contains information related to each player and his assigned region. (We consider our players to be an object referred to using his assigned region, hence the name "Object Point"). The actions that players are supposed to do are defined within the <ACTION_LIST> tag.

2.3. Support for Reusability: the Built-in Strategies

A team strategy is made up of several sub-strategies to be used for different game situations. Some examples are, defense sub-strategy, attack sub-strategy and set play sub-strategies. Various strategies can be developed for a single game situation. We have worked toward the goal of presenting sub-strategy components which can be combined together to form the whole team strategy. For this purpose, a library is considered in Strategy Builder which contains built-in strategies (figure 2).

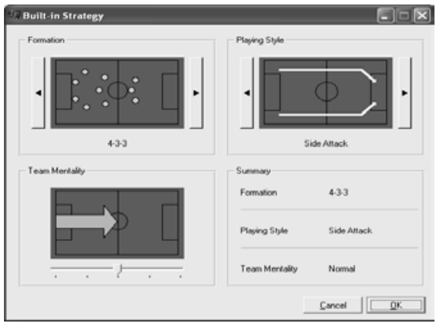
A built-in sub-strategy is defined through the following parameters:

1. **Formation:** describes the distribution of the players in the field. It includes predefined common soccer formations (4-3-3, 4-4-2, 3-4-3, etc).
2. **Playing Style:** is concerned with the type of attack and defense used. Attacking style is mostly concerned with the type of actions performed by the players to attack [4]. These actions may be mainly short passes and dribbles or they may be long passes by the players. Defense type is mainly concerned with the type of marking of the players.
3. **Team Mentality** (offensive, normal, defensive): In a team with an offensive mentality players are more concerned about attacking by advancing more into the opponent half. On the other side of the extreme, in a team with a defensive mentality, players remain on defending positions in their half, trying to prevent opponent's attacks.

In addition to the built-in strategies, the user's designed strategies can be saved in the SBLang format to be used in future designs as a user-defined sub-strategy.

2.4. Compilation

The SBLang representation of a designed strategy is compiled into CLang during the compilation process. For this purpose an SBLang compiler is embedded into Strategy Builder. Some tags are barely directives and do not compile into a corresponding CLang. Some tags have an equivalent CLang statement (e.g. the regions), but some compile into more than one CLang statements. For example the <pass> tag in SBLang is considered for defining a pass rule. Its elements define the passer and receiver and their respective regions. But this tag will be compiled into two CLang rules. One rule would be the pass rule itself, and the other a positioning for the pass receiver. The two rules together coordinate these two players in order to have a successful pass.



Formation 4-3-3

Playing Style Side Attack

Team Mentality Normal

Figure 3. A built-in sub-strategy

3. Caspian Coach

There is an initial set of rules, which is the result of visually designed strategies in Strategy Builder, and processing opponent log files then our coach selects the most suitable set of rules to start the game. Strategy Builder separates rules of each strategy with one line of comment (denoted by the '#' character) describing that special strategy. It is actually the protocol between Strategy Builder and the Caspian Coach. The comment is used by the coach to identify the type of the strategy which is now represented in CLang.

Our coach analyzes the opponent's log files, and based on this analysis selects the most suitable strategy to start the game with. For example if the opponent team doesn’t act well in the wings, the coach selects Side-Attack as its team's playing style. At the beginning of the simulation, the coach says the initial set of rules. During the game the coach may switch between the currently on rules, and another set of available off rules, thus adapting its coached agents to the current game situation.

During the game, the coach observes the game and can analyze whether the currently active strategy is still doing effectively. If not, it says the agents to turn off the currently active set of rules, and activate another set of rules. The result is that the agents get adapted to the changes made in the opponent team's playing style and the current game situation.

4. Caspian Coachable Team

The Caspian Coachable Team is another research product for the 2004 RoboCup Coach Competition. Great efforts have been made to produce agents of good understanding and high performance. The most time-consuming task of a coachable agent is verifying conditions of all the active rules to decide which one has got fired, and then performing the wisest action possible if more than one rule has got fired. The whole process must be done in only one simulation cycle, which is equal to 100 ms. Therefore efficiency has become an important concern in the design of coachable agents. To address this challenge, several approaches have been considered. One of the approaches is providing the agents with a knowledge-base in Horn Normal Form

[5]. Another approach is providing a hierarchical structure for the set of rules in order to decrease the amount of rules to be verified in each cycle.

4.1. A Knowledge-Base in Horn Normal Form

In general, verifying whether an arbitrary condition is satisfied requires heavy process (mainly because of different possible combination of logical operators). In our approach to increase the performance, we reform all conditions into several Horn Sentences, using basic logic rules. Thus, it is enough to verify the sequence of atomic conditions to the point that a false condition is encountered. The remaining atomic conditions are ignored, thus it results in a better performance.

4.2. Rule Classification

Although verifying conditions in Horn Sentences has a great impact on the agent's general performance, but the obtained performance is not good enough in a system under time pressure. This is because of the high amount of sentences which are needed to be examined. In order to reduce the number sentences that should be verified in each step, we present a hierarchical classification of the set of rules, on the basis of the current state of the game. We define the game state as the team which owns the ball.

5. Conclusion and Future Work

Coaching research is mainly concerned with the following challenges:

1. Mapping strategies into CLang
2. Advising coachable agents
3. Integrating advisable action-selection mechanism in player agents

To address the first challenge, our team has developed an Integrated Visual Environment called Strategy Builder which facilitates team strategy development. This tool automatically translates the visually designed strategies into CLang. This leads to a rapid development of the team strategy which is used by the coach agent. Coach agent makes use of the rules generated by Strategy Builder to form an initial set of rules, which are used to apply suitable strategies during the game. The drawback of the current approach is that the rules are hard coded to the coach agent and the coach is only able to activate or deactivate the set of rules. The coachable agent is another research product of Caspian Simulation Team. Our coachable agents have a good understanding of even complicated CLang rules. Moreover the agent acts in a reasonable speed which is the result of utilizing a knowledge-base in Horn Normal Form, and Rule Classification.

References

1. Riley, P., Veloso, M.: Coaching Advice and Adaptation. In Proceeding of RoboCup Symposium 2003
2. Manuela Veloso, in robocup-sim@robocup.biglist.com, 4 July 2002. http://www.robocup.biglist.com/robocup-sim/archive/
3. Bray, T., Paoli, J., Sperberg-McQueen, C. M., editors: Extensible Markup Language (XML) 1.0, (1998), http://www.w3.org/TR/REC-xml.

4. Paulo Reis, L., Lau, N.: COACH UNILANG – A Standard Language for Coaching a (Robo)Soccer Team. In Birk, Coradeschi, Tadokoro, editors, RoboCup 2001: Robot Soccer World Cup V. Springer, Berlin, (2002).
5. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall International, Inc. 1995. Page 270.