Wright Eagle 2-D Simulation Team

Chen Xiaoping, Fan Changjie, Wu Feng, Wang Jiliang, Cai Jingnan

AI Center of Department of Computer Science and Technology,
University of Science and Technology of China,
Hefei, China

cjfan@mail.ustc.edu.cn WWW home page: http://mas.ai.ustc.edu.cn

Abstract. This paper will give you a brief description of the Wright Eagle 2-D Simulation team. Our long-term goal is to build a robot soccer team where the decision making part is useful for all the multi-agent system. So we do not only pay much attention to the high-level decision making models, but also the low-level implementation details. The article below will describe our approaches of these two aspects in detail.

1.Low-level Design

Player Identification System

Under the current server, usually it is very hard to get full information identification through vision alone. We cannot determine the uniform number of a player, or even which team he belongs to. These non-determinate factors can bring huge trouble in later high-level decision processes. So we expect the problem be solved early in low-level processes.

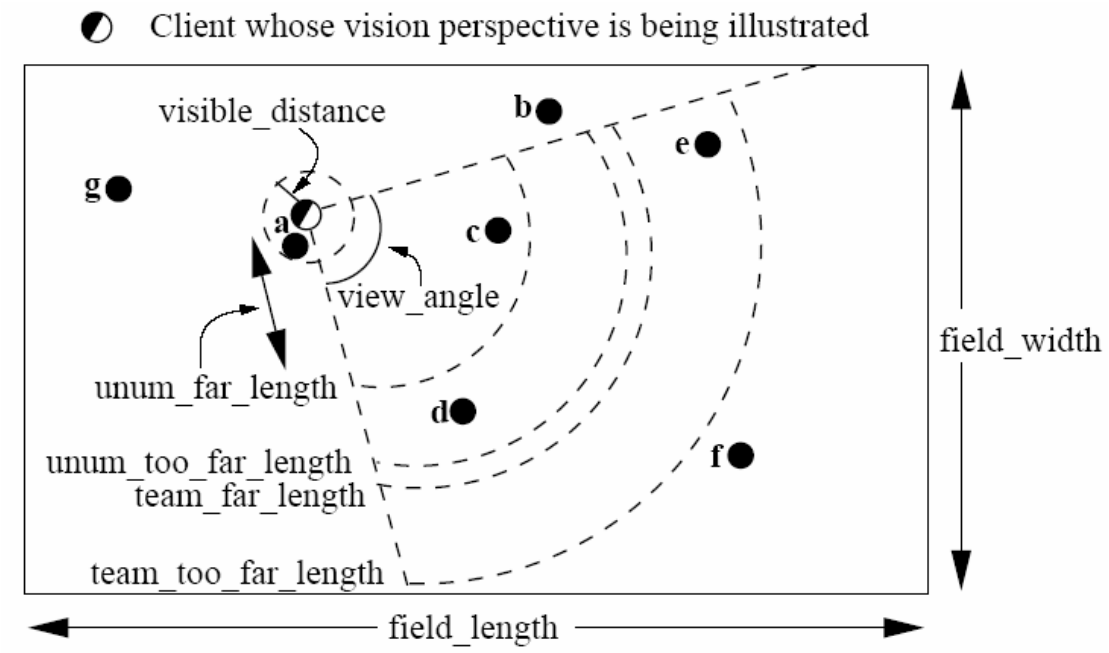

Figure 1. The vision sensor model of an individual agent in the soccer server.

The past way to deal with the problem is to match the nearest one to the numberless or teamless player who has been seen in history. When there are more than one likely results of matching, it is left to high-level processes without identification.

But the following facts should be noted:

1. Only those history players who are capable to move into the visual range should be considered in this matching.

2. When a player d whose history information is determined meets only one unknown player in his possible range of motion, we just cannot make sure they should be matched. But conversely, an unknown player meets only one player whose history information is determined, we may be confident to match them as one.

By making a round of match using the 2nd rule we can greatly reduce the number of cross-matching in the later process of evaluation by distance. In experiment it can do the work well in one millisecond. Following is the chart of the algorithm.

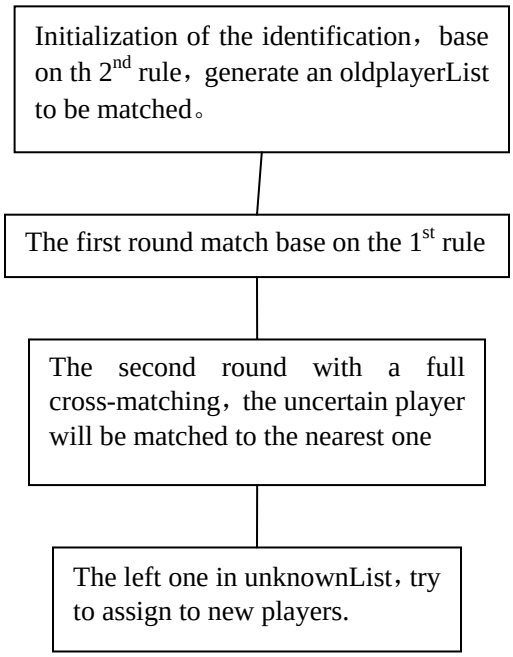

Figure 2. The briefly flow chart of the algorithm.

Generally, The algorithm will stop in the first three steps. With a result that we didn't encounter an wrong match till now.

2.High-level Design

Decision Evaluation Mechanism

We regard the whole decision process as a POMDP model, in which the probabilities of observation are transformed to the probabilities of state-transition of the next stage. Thus the model is simplified to an MDP model.

States **S**: In each state the ball is in the control of some player. When the ball is free, we directly predict the possible states of the ball under control, based on the interception model which is presently quite mature.

Action **A**: The atom actions prescribed by the server are not used. Instead a behavior module is introduced with domain knowledge. Examples of action that we defined are Pass, Dribble, Shoot Position and etc. Each action has its own formal model, to compute the possible state and the probability of state transition.

Reward **R**: The basic idea is to discriminate the several main scenarios, appointing different goals to each. The highest reward is achieved when the goal is reached, otherwise a small reward will be given according to state point sensitivity. A negative reward is given when failure.

A basic decision process is:

Analyze the situation, make sure the current state and then confirm the goal state.

The action generator can generate a mass of actions as candidate. Each action has its reachable states and the transformation probability. The value of the states achieved is computed by approximately method iteratively.

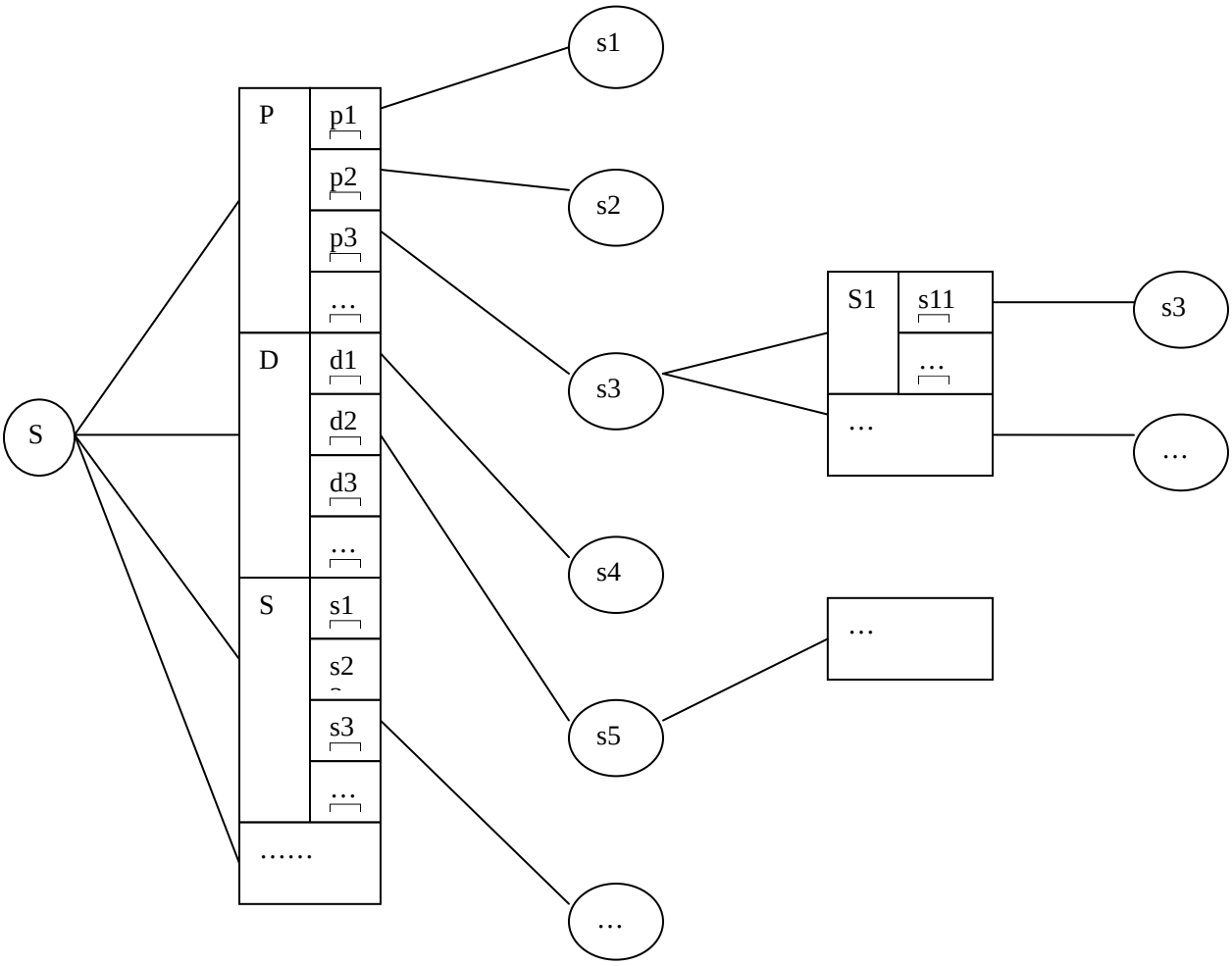

Figure 3: A sketch of a simple decision: The rectangles are action generators and the circles are states. P represents Pass. D represents Dribble and S represents Shoot.

The current algorithm can usually achieve 3 step of the iteration. Namely, we can generate a cooperation which it made up with two Pass and one Shoot for example. Because the ball controller and his teammate use the same algorithm to make a programming, when their visible information are approximate, the optimal decisions are nearly same. As a result, a cooperation with a Pass and Position behavior is easy be achieved.

3.New Defense Architecture

In the soccer competition, defensive strategy often requests to keep some, but aggressive strategy then otherwise. Therefore the defense part and the attack one are quite different. In order to make our team more aggressive, the main loop of our top level decision making is: we evaluate the situation of current game first, if any aggressive strategy is available, then do it, otherwise we enter the defense part.

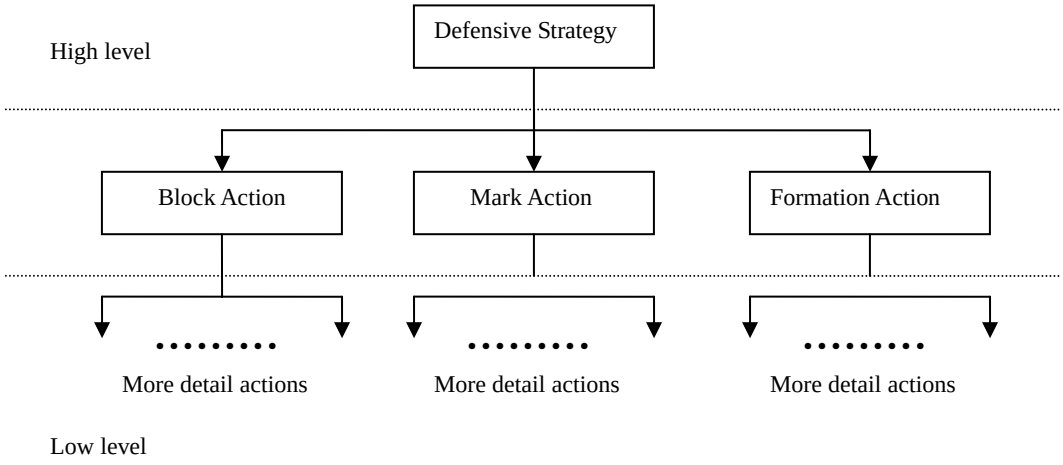

Figure 4: The old layer architecture

The old defense architecture of our team was a strictly hierarchical one. The defensive strategy layer was responsible for deciding which action (block, mark or formation) would be chose, and then we could evaluate each action more precisely. The main idea behind this architecture was to repeatedly divide the defense tactic into subtasks.

This architecture was successful, but after several years of changes and amendments it was difficult to see clearly which action would be better without some detail. And the control flow was strictly top-down, e.g. it was impossible for an action in detail to change another better strategy.

Therefore we decided to alter the decision-making process, using behaviors inspired by behavior-based robot architectures. Our new architecture is still up to a certain degree a hierarchical one, but there are no restrictions like subdivisions into layers and strictly top-down control flow. Typical instance of Figure 2 is: the part of Defensive Strategy generates an approximate optimal task, and each behavior checks this task then gives their results and at last the evaluation part chooses the best one as the final action.

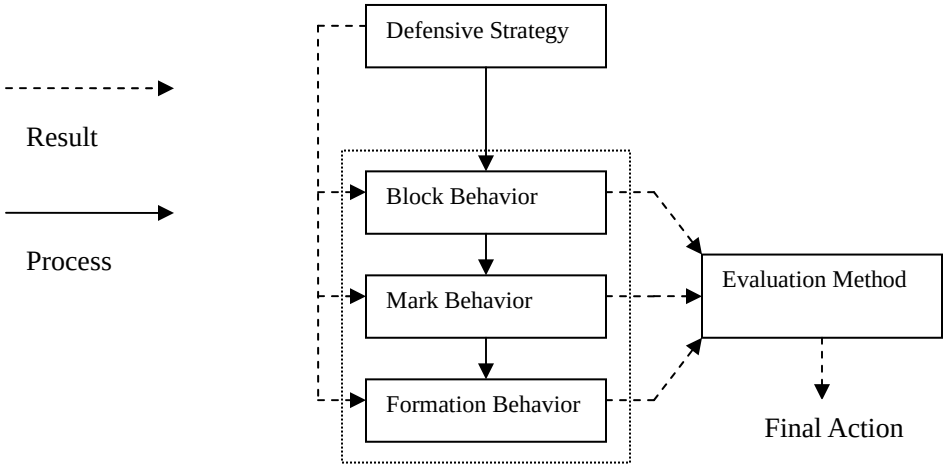

Figure 5:The Behavior Architecture