Adaptation to Multi-Agent Environment by External Advice

Hidehisa Akiyama, Daisuke Katagami, Yoshiaki Yasumura, Katsumi Nitta
Tokyo Institute of Technology
4259, Nagatsuta-cho, Midori-ku, Yokohama-shi, Kanagawa, 226–8502, JAPAN
Email: {akiyama,katagami,yasumura,nitta}@ntt.dis.titech.ac.jp

***Abstract*— In multi-agent environment, it is necessary for each agent to improve his cooperative action dynamically according to the situation. However, in the environment where agents' sensory capabilities are restricted, it is difficult to perform exact situation judgment and to generate the effective cooperative action. In such an environment, the supervisor who can overlook the whole and can give advice to agents plays an important role. We use the RoboCup Soccer Simulator as an experiment environment. In the simulator, we can use a coach agent as a supervisor. First of all, we aimed to design the agent who can accept and utilize advice effectively.**

In this paper, we propose a strategy framework that adopts the supervisor's advice to each agent effectively. This framework parameterizes team strategies and tactics, and provides some of rules about the parameterizing of tactics. In this framework, team strategies include not only basic tactics but also additional tactics. Additional tactics is a subset of one strategy and can be converted to differential parameters to basic tactics. This feature enables us to express the additional tactics by the Standard Coach Language used in the simulator, and to store the set of local tactics consistently. A coach agent can use the stored tactics in an official game to improve the team performance as a supervisor. And also, this framework enables human to become a supervisor. Because of this, we can store the effective tactics using human trainer's instruction.

I. Introduction

In multi-agent environment, it is necessary for each agent to improve his cooperative tactics dynamically according to the situation. However, in the environment where agents' sensory and communication capabilities are restricted, it is very difficult for each agent to perform exact situation judgment and intention understanding, and to generate effective cooperative actions. Even if it is human, it will be difficult to cope with these problems. In many cases, a supervisor which manages a team strategy and gives some advice messages in real time exists when the team consists of multiple persons.

We noticed the role of such a supervisor and aimed to adapt a team to the environment and improve a team performance by supervisor's advice. As an experimental environment, we use the RoboCup Soccer Simulator [4] that is used in the RoboCup Soccer Simulation League. In previous researches, the researches about a coach agent that gives the advice have been done actively [1], [12], while the researches about the agents that receive the advice hardly exist [14]. So, it is hard to say that advice is used effectively. Then, first of all, we aimed to design the agent who can accept and utilize advice effectively. In this paper, we propose a team strategy framework to receive supervisor's advice and reflect it to the cooperative action effectively. And we also describe the application to acquire the effective tactics.

The setup of the paper is as follow. In Section II we introduce the RoboCup Soccer Simulator environment and previous researches about the simulated soccer. In Section III we explain the concept of our framework, and in Section IV we described how to use our framework. In Section V we introduce our application and in Section VI we conclude and discuss about future works.

II. RoboCup Soccer Simulation

A. Agents in the Simulator

We can use two different kind of agents, a player and a coach, in the RoboCup Soccer Simulator. All agents are run by different process and have a sensor independently.

Player agents' vision sensor is severely restricted. They can observe the environment locally and can get only a part of objects' information on the soccer field. And also, the information that they received is very noisy and ambiguous. So, it is very difficult for player agents to recognize the field state correctly and to correct cooperative actions and improve a team performance by each player's judgment.

On the other hand, a coach agent has global vision. A coach agent can get a complete positional information of all objects on the soccer field. And also, a coach agent can send advice to player agents during a game. So, A coach agent can take a role of supervisor.

B. Advice from a Coach Agent

The Standard Coach Language (called CLang) [5] can express the advice concerned with soccer and has already been built into the simulator. In an official game, almost all advice message from coach agent must be expressed by CLang. Because of this, we use CLang, but extend it if necessary.

CLang expresses the advice by production rule structure. Player agents that receive the advice must be able to understand the described rules, verify the described conditions and perform the described actions. And also, CLang is designed so that we can express a flexible advice message. Because soccer is a complex game, it is almost impossible to prepare the advice message for the dynamically occurred situation in advance. Therefore, CLang has a certain amount of extendibility. For example, we can define the original words that contains complex conditions or actions and can use them after. This enables us to use a more complex advice easily. Figure 2 shows a very simple example of advice message described by CLang. This example defines the action rules for a certain defensive situation and activates that.

```
( define
   (definerule goal_defence
      ((and (bowner opp {X}) (bpos (rec (pt -52.0 -30.0) (pt -35.0 30.0)
         (do our {2} (home ((pt -36.0 -10.0) + ((pt ball) * (pt 0.1 0.2)))
         (do our {3 4} (mark {9}))
         (dont our {3 4} (offside)))
   )
)
(rule (on goal_defence))
```

Fig. 1. Example of CLang Advice

A coach agent that sends advice must observe and analyze a game dynamically. In order to prevent too much interference to a game, the number of times of advice is restricted. So, a coach agent must send an effective advice within a few times.

C. Usual Strategies on the Simulated Soccer

In this paper, we call the cooperative action rules performed by one or more player(s) "Tactics", and call the rule set that contains several tactics and specifies a team characteristic "Strategy". Strategy is shared by all player agents and a coach agent belonging to the same team. During a game, all agents in the same team refer to the same strategy and decide their action based on it. A team can have several strategy, but only one strategy can be referred simultaneously.

Now, in the soccer simulation league, almost all teams adopt a simple strategy framework. This framework contains only one tactics parameter that affects their basic action pattern, team formation, and so on. At the special situation, player agents that adopt such a strategy must decide their action using built-in action rules. In this case, if we want to change the team strategy, we must exchange all tactics parameters or directly rewrite the agent program. This means that it is difficult to modify or add a tactics flexibly and the adaptability of the team becomes low. Thus, it is hard to say that advice is used effectively.

In next section, we propose and describe a new team strategy framework that enables us to correct tactics by external advice.

III. Strategy Framework

A. Overview

Our strategy framework manages the whole team strategy and enables us to correct tactics consistently. In this framework, a strategy and tactics are described by specific format. This framework has one or more strategies, and has the interface protocol that can exchange the strategy itself and correct tactics from outside. Here, this interface protocol is CLang.

Player agents belonging to the same team have the copy of all known strategies. Player agents refer to the one of those strategies during a game. Each player follows the referred strategy, and performs decision making according to the situation with the tactics included in that strategy. When player agents receive advice, they perform the exchange of strategy or the correction of tactics.

We assume the existence of superviser when we use this framework. A supervisor analyzes the environment, and gives advice to player agents. A player agent must be able to understand the specified strategy format and interpret the advice from the supervisor. And also, a player agent must exchange the strategy or correct tactics based on the advice. These operations are common to all player agents and included in strategy framework.

In order to apply the tactics according to the situation, supervisor must generate suitable advice. However, it is difficult to acquire the effective tactics during a game. It is indispensable to verify the validity of tactics by the prior test. So, we have to acquire the additional tactics before the game. And also, the acquired tactics must be convertible for the format that can be described by CLang.

We use CLang as the interface protocol to a strategy. This is because advice is permitted only to a coach agent in the official game of RoboCup. But, supervisor is not restricted to a coach agent. At the training before a game, supervisor other than a coach agent may give advice. So, if it is at the training time, a special trainer agent or human trainer can give advice to player agents. We assume that we store the effective strategy and tactics by prior training. The task that a coach agent performs is only choosing a suitable strategy or suitable tactics.

B. Decision Making by Player Agents

The kind of actions that player agents can perform is one of the components of the strategy framework. These actions must be implemented in advance. In CLang, some actions are already classified and they can be used in advice message immediately. Our classification of actions also follows CLang.

However, because CLang is expressed by production rule structure, if player agents accept the advice simply, a possibility that the conditions of the existing rule will be overwritten and fault will occur is very high. It is desirable for player agents to accept CLang gently.

In order to cope with this problem, we adopt the technique that evaluates action options based on a fixed evaluation function when player agents perform the decision-making [6]. In this technique, the priority of each action option is parameterized. So, it is possible to change the feature of decision-making little by little. This enables us to keep or change the feature of strategy flexibly.

The tactics in this paper have the parameter set of the priority of each action. We add some parameters further and express more detailed tactics. Usually, player agents make decision by referring to this parameter set. When they receive the advice from supervisor, parameters are modified for the first time.

The evaluation function and the rule of parameterizing are common to all player agents and included in strategy framework.

C. Parameterizing the Tactics

Until now, some researchers realized the characterization of cooperative action using parameterized strategy [10]. But these methods have only strategical flexibility, not tactical. We extend these methodology. In our framework, the operating condition of tactics and the player set variable are operated as the tactics parameter. This enables us to parameterize tactics more flexibly. For example, it becomes possible to add tactics like the setplay in the specific situation, without affecting an original strategy and no-related players.

In this paper, tactics is described by the following parameter set.

- The operating condition of tactics
  The conditions which should be fulfilled when tactics are applied. This is expressed by logical connectives of the atomic condition of field state. We can use the game play-mode, the ball position and so on as a condition of field state.
- The set of the player with which tactics are applied
  Tactics is applied only to players belonging to this set. The size of a set is arbitrary. It is possible not only to specify the player directly by the uniform number, but also to specify the player that fulfills specific conditions.
- The basic position of each player
  The position coordinates to which the player should move. This parameter forms a team formation. It is possible to specify not only the static coordinates but also the relative position to a certain object.
- The priority of each action of each player
  Player agents have several kind of actions such as pass, dribble and so on. This parameter gives the priority of each action for the evaluation function. A value is the real number. The ranges of a value are [0, 1].
- The positiveness of each action of each player
  If this value is high, a more offensive play will be prefered, and a safer play will be prefered if low. This parameter gives the additional priority of each action for the evaluation function. A value is the real number. The ranges of a value are [0, 1].
- The priority of each action target of each player
  This parameter determines the priority of pass partner, dribble target point, and so on. The description format of this parameter differs according to the kind of action.

Player agents give these parameters and their world model information to the evaluation function.

The parameters of tactics describe not only the feature of the individual play by one player but also the feature of the cooperative play by two or more players. For example, if we want to realize a cooperative defense play like a zone-press [8], we should select only the players near to the ball. In this paper, both an individual play and a cooperative play are called tactics and are not distinguished.

Strategy includes one or more tactics parameter set. But, strategy must have one "basic tactics". Basic tactics do not have an operating condition and an applied player set. All player agents in the team are automatically selected as an applied player. Instead, we can not omit other parameters. On the other hand, tactics other than basic tactics must include an operating condition. We call this type of tactics "additional tactics". We can omit a part of parameters in additional tactics. If the parameter in additional tactics is omitted, the parameter defined in basic tactics is derived and used. That is, additional tactics can become the differencial parameters to basic tactics.

When a player agent makes decision, he verifies the operating condition of all additional tactics using his internal world model information. If the additional tactics in which an operating condition agrees are not found, player agents make decision based on basic tactics. When two or more additional tactics are found, priority is given to the tactics added later.

D. Action Classification

In CLang, some basic actions and conditions for soccer are already classified. We follow this classification fundamentally. Table I shows the classified actions. We assume that these actions are implemented to player agents in advance and superviser does not modify those. This is because the quality of the individual action gives big influence to the cooperative play. If basic actions are modified, we have to retest all related tactics. Therefore, we fix the basic action while improving the team strategy.

TABLE I
CLASSIFIED PLAYER'S ACTION

Positioning	Kicking
Basic position	Shoot
Chase the ball	Keep the ball
Recover stamina	Dribble
Get free	Clear
Mark	Pass
Press	
Block the pass	
Block the shoot	
Offside trap	

Some actions have further options, the move distance of a dribble, the choice of challenging or safer pass, and so on. These options can be expressed by the positiveness parameter of actions. This parameter corresponds to "The positiveness of each action of each player" in Secton III-C. And also, we set up the target priority about the action that has the target. This parameter corresponds to "The priority of each action target of each player" in Secton III-C. These options must be also implemented in advance.

When player agents receive the advice, they try to perform the described action in the maximum. However, if they are in the situation that the described action cannot be performed, they never perfome that action. For example, if a player agent has no sufficient stamina, almost all positioning actions are rejected because he cannot perform any dash action.

Finally, the flow of player agents' decision making is shown as Figure 2.

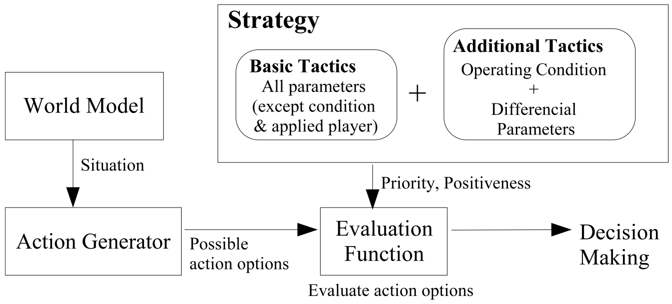

Fig. 2. The flow of player agents' decision making

IV. Adaptation by Advice

A. Advice

Because CLang adopt the production rule structure, actual advice is expressed in the form of "condition + action (+ action target)". The condition part is directly adopted as the tactics parameter. For the positiveness of actions, because this cannot be expressed only by CLang, we define the new actions divided into some positiviness levels and use them properly. And also, CLang enables us to validate or invalidate the defined production rules. This means that the rules can be promoted or prohibited. If the rule is validated, the priority of actions included in that rule is increased, if invalidated, the priority is decreased.

We assume that only additional tactics is used as advice. This is because basic tacitcis cannot be expressed by CLang. If all elements contained in basic tactics are once expressed, the size of advice message will become huge. Also in real soccer, too much long advice is unnatural. Moreover, it is almost impossible to express many numerical parameters by CLang. First of all, it is unnatural that advice is given with the numerical parameters. However, because additional tactics includes only differencial information to basic tactics, additional tactics can fully be described by CLang. Therefore, we also assume that all agents have basic tactics as a common knowledge in advance.

Advice can include several rules. This enables us to describe a little complicated tactics by one advice. We assume that one tactics is describled by one advice. If the size of advice becomes large and advicle cannot be described by one advice, we define our original words and compress the advice message.

B. Store the Tactics

In order to acquire the effective tactics, it is indispensable to verify the validity of tactics by the prior test. However, it is very difficult to acquire effective tactics automatically. For the complex tasks like a soccer, first of all we have to analyze and immitate a human's decision-making process. The system that enables us to join the simulated soccer as a player is already proposed, and shows human's high adaptation capability. So, we do not try the automatic tactics acquisition, but try to acquire tactics by off-line training that human trainer gives advice. In this case, human trainer works as supervisor instead of a coach agent. In our research, we are developing the tool that can convert the human trainer's instruction to CLang and store as the tactics [3]. Figure 3 shows the flow of the instruction and store of the new additional tactics.

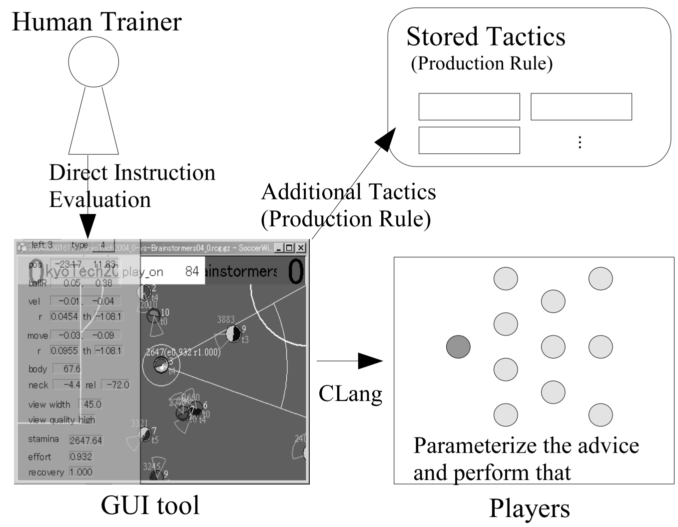

Fig. 3. Instruction and Store of the Tactics

The instruction information from human trainer is converted to advice message discribed by CLang and given to player agents. Player agents generate differencial parameters based on the received advice and basic tactics using the common parameterizing algorithm. If a human trainer judges that team performance is improved and new effective tactics is generated, a human trainer stores all production rules as the new additional tactics using GUI tool, and if necessary, player agents store all differencial parameters. This production rule set is used as an advice to give one additional tactics.

It may be difficult for a trainer to understand the meaning of parameters. In order to avoid this difficulty, tactics parameterizing operation is managed only by player agents. Because of this, it is not necessary for the human trainer to know about the detail of parameterizing. The trainer should just have the knowledge of the instruction protocol.

If the modification of tactics parameters is performed over the whole team and the team performance becomes stable, we can use that parameters as the basic tactics. This means that we acquired a new strategy.

C. Using Stored Tactics

In an actual game, a coach agent only uses the stored additional tactics as advice. A coach agent analyzes a game and tries to dynamically improve the team performance using the stored additional tactics. When a coach agent gives the advice to player agents, each player parameterizes the given advice, combines with basic tactics and uses the generated parameters as a new tactics (Figure 4).

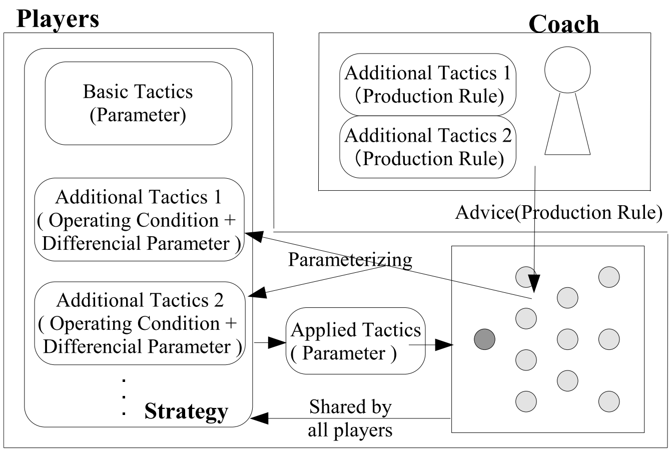

Fig. 4. Using stored additional tactics

Since advice consists of one or more rules, player agents interpret advice to the individual rule and convert them to the differencial parameters. Note that it is not necessary to parameterize the additional tactics in advance. If player agents have the same parameterizing algorithm, they can regenerate the same parameters.

As described in Section III-C, if the conflict is found for the conditions in several tactics, priority is given to the tactics added later.

D. Change Team Strategy

Because strategy has basic tactics as main component and basic tactics cannot be described by CLang, we cannot express the whole strategy by CLang. When the team tries to change the whole strategy, all agents in the team must exchange the basic tactics parameter and reset all applied additional tactics and a coach agent must exchange all stored additional tactics. So, for the change of strategy, all agents have the consensus about that in advance, and the coach agent gives the message that specifies the change of strategy. In order to realize this, it is necessary to extend CLang and define a original protocol. Of cource, if the team has only one strategy, this definition is not needed.

V. A Training Tool

We are developing a training tool that enables human to become a trainer. This tool has following features.

- It can visualize all information that a coach agent can get.
  This tool can visualze not only all objects in the environment, but also communication messages between player agents. We can get all coach agent's sensory information visually.
- It has the interface for the human trainer.
  This tool enables us to input the several instruction information using a mouse and a keyboard. This means that we can easily input the intuitive judgement and evaluation.
- It can generate advice message described by CLang.
  The input data is converted to the actual advice message and sent to player agents. And also, all of generated advice can be stored to the disk as additional tactics.
- It can make the specific situation. This feature helps us to retry the same situation and to design a set-play.
- It can visualize the player agents' internal information.
  This feature helps us to recognize a cross perceptual aliasing [9] between a player agent and a coach agent.

Especially, we think that it is important to consider about the problem of cross perceptual aliasing. Cross perceptual aliasing means a gap of recgnition between a teacher and a lerner. Here, a teacher is a coach agent and a learner is a player agent. Since a coach agent has global vision and a player agent has local vision, a gap of recognition may become very large. If the gap is too large, the effect of advice may not be expectable even if the effective tactics is acquired in advance.

VI. Conclusion and Future Works

In this paper, we proposed the new strategy framework to receive the external advice, that enables us to dynamically correct the team strategy using the stored additional tactics as advice. All agents belonging to the same team share the parameter set as the basic tactics and player agents have same algorithm to parameterize the advice message. Since the additional tactics described by advice can be converted to the differencial parameters to the basic tactics dynamically, stored additional tactics can be described by CLang and given to player agents during a game. This framework enabled player agents to take in supervisor's advice effectively.

However, the more advice are given, the more computational time to verify the operating condition of additional tactics is required. This computational load may become a bottleneck. Since the agents on the soccer simulator require the real-time processing, it is necessary to adjust the number of advice.

We have to consider about the acquisition of the coaching strategy. The choice of tactics and strategy is the task of a coach agent. There are several problems which should be solved such as a game analysis, opponent modelling, finding a counter strategy, and so on. For these problems, We plan to extend our strategy framework to the management of the coach strategy and to train the coach agent by human trainer.

And also, we think that cross perceptual aliasing is one of the most important problems. We checked that the recognition capability of our player agent has a certain amount of accuracy. The serious problem was not occured in our experimental environment. However, if a coach agent advice other developer's player agents and they have low recognition capability, we cannot generate and store the effective tactics. It is necessary to investigate the difference of the validity of the advice by using a player agent that has different recognition capability. This problem is also concerned with the modelling capability of a coach agent. We have to consider about the method to model the other agents more strictly.

REFERENCES

[1] Gal A. Kaminka, et al.: Learning the Sequential Behavior of Teams from Observations, RoboCup2002: The Sixeth RoboCup Competitions and Conferences, Springer Verlag, 2003.

[2] Hidehisa Akiyama, et al.: Dynamic Adaptation to Multi-Agent Environment by Coach's Advice, The 17th Annual Conference of JSAI, 2003.

[3] Hidehisa Akiyama, et al.: Team Model Design to Adapt to Dynamic Environment by External Advice, The 18th Annual Conference of JSAI, 2004.

[4] Itsuki Noda, et al.: Soccer Server: A Tool for Research on Multiagent Systems, Applied Artificial Intteligence, 1998.

[5] Itsuki Noda, et al.: RoboCup Soccer Server Users Manual for Soccer Server Version 7.07 and later, http://sserver.sourceforge.net/, 2002.

[6] Jan Lubbers, et al.: The Priority/Confidence Model as a Framework for Soccer Agents, Robocup-98: Robot Soccer World Cup II, pp. 162–172, Springer Verlag, 1999.

[7] Junichi Akita, et al.: OZ-RP system: RoboCup Simulation League Human Interface to beat replicant, The 12th JSAI SIG-Challenge Workshop, pp.23–28, 2001.

[8] Manabu Ichikawa, et al.: Achievement of cooperated defense strategy in RoboCup soccer, The 18th Annual Conference of JSAI, 2004.

[9] Chizuko Mishima, et al.: Active Learning from Cross Perceptual Aliasing Caoused by Direct Teaching, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1440–1425, 1999.

[10] Luis P. Reis, et al.: Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents, Balancing Reactivity and Social Deliberation in Multi-Agent Systems, pp. 175–197, 2001.

[11] Luis P. Reis, et al.: COACH UNILANG - A Standard Language for Coaching a (Robo)Soccer Team, RoboCup 2001: Robot Soccer World Cup V, Springer Verlag, 2002.

[12] Patric F. Riley, et al.: An Empirical Study of Coaching, Distributed Autonomous Robotic Systems 5, pp. 215–224, Springer Verlag, 2002.

[13] Paul Scerri and Johan Ydren. End user specification of robocup teams, Robocup-99: Robot Soccer World Cup III, pp. 450–459, Springer Verlag, 2000.

[14] Paul Carpenter, et al.: Integration of Advice in an Action-Selection Architecture, RoboCup2002: The Sixeth RoboCup Competitions and Conferences, Springer Verlag, 2003.

[15] Yao Jinyi, et al.: Technical solutions of TsinghuAeolus for Robotic Soccer, In proceedings of RoboCup 2003 Symposium, 2003.