

# アドバイスによって動的環境へ適応するチームモデルの設計

Team Model Design to Adapt to Dynamic Environment by External Advice

秋山英久      片上大輔      安村禎明      新田克己  
Hidehisa Akiyama      Daisuke Katagami      Yoshiaki Yasumura      Katsumi Nitta

東京工業大学大学院  
Tokyo Institute of Technology

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 capability is restricted, it is difficult to judge the situation exactly. In such an environment, the supervisor who overlooks the whole and gives advice to agents plays an important role. We use the RoboCup Soccer Simulator as experiment environment. In this report, we propose a team model that adopts the supervisor's advice effectively. This model parameterizes team strategies and player agents' tactics, and provides some of rules about player agents' process. Team strategies include not only basic tactics but also additional tactics. Additional tactics is a subset of one strategy and can be expressed by CLang. This enables us to store the set of good tactics without futility, and coach agent can use those in the official game to improve the team performance as a supervisor.

## 1. はじめに

マルチエージェント環境においては、各エージェントが状況に応じて協調動作を動的に適応させ改善していく必要がある。しかし、エージェントの知覚能力やコミュニケーションに制限が加えられている環境では、各エージェントが正確な状況判断を行うことがまず困難であり、行動改善のための意思疎通も望めない。このことは人間の場合も同様であり、複数人数によるチームが構成されている場合はチームを管理するスーパーバイザが存在することが多い。

我々はこのようなスーパーバイザの役割に注目し、そのアドバイスによってチームパフォーマンスを改善することを目的としている。本研究では、RoboCup サッカーシミュレーションリーグで用いられているサッカーシミュレータ [Noda 1998] を実験環境とする。従来研究においては、アドバイスをを行うスーパーバイザに関する研究は盛んであるものの [Riley 2002][Kaminka 2003]、スーパーバイザからのアドバイスを受け取るエージェントに関する研究例は少なく [Carpenter 2003]、アドバイスが有効活用されているとは言い難かった。そこで、本稿では、スーパーバイザのアドバイスを受け入れ、協調レベルの行動へ有効に反映するためのチームモデルの提案を行う。

## 2. RoboCup サッカーシミュレーション

### 2.1 シミュレータ上のエージェント

サッカーシミュレータ上で動作するエージェントとして、プレイヤーとコーチの二種類が使用可能である。いずれもすべて異なるプロセスで動作するプログラムであり、それぞれが独立したセンサを所有している。プレイヤーの視覚センサには厳しい制限が設けられており、プレイヤーはフィールド上の物体の位置情報を局所的にしか得られない。プレイヤーがフィールドの状態を正確に認識することは極めて困難であり、個々のプレイヤーの判断でチームレベルの協調動作を改善することは困難となっている。

一方、コーチはフィールド上の物体の完全な位置情報を得ら

れる。更に、試合中、コーチはプレイヤーに対してアドバイスを送ることができる。プレイヤーはすべて同格の存在だが、コーチはスーパーバイザとしての役割を持つことができる。

### 2.2 コーチのアドバイス

シミュレータには標準コーチ言語 (Standard Coach Language: CLang) と呼ばれるアドバイス専用言語が組み込まれている [Noda 2003]。CLang には、独自の単語が定義可能であるなど、ある程度の拡張性を備えた言語仕様が用意されている。コーチからのアドバイス内容は CLang で記述することが義務づけられている。本研究でも CLang の文法に沿ったアドバイスを使用する。

CLang では、アドバイス内容はプロダクションルール形式で表現される。アドバイスを受けるプレイヤーは、記述されている行動ルールを理解し実行できなければならない。

アドバイスをを行うコーチは、試合をリアルタイムに観察、分析しなければならない。更に、試合への過度の干渉を防ぐためにアドバイス回数と使用頻度には制限が加えられている。コーチはその制限の範囲内で有効なアドバイスを行わなければならない。

### 2.3 シミュレーションサッカーにおける従来戦略

本稿では、単体もしくは複数のプレイヤーによる局所的な行動ルールを戦術と呼ぶ。内部に戦術を複数含み、チーム全体としての特徴を備えたルールセットを戦略と呼ぶ。戦略はチーム内のすべてのプレイヤーとコーチによって共有される。試合中、チーム全体では同じ戦略を参照して行動する。チームは戦略を複数持つことができるが、同時に参照できる戦略はひとつだけである。

これまでのシミュレーションリーグでは、戦略は基本の行動パターンに影響する戦術パラメータをひとつだけ持ち、特殊な状況下ではエージェントに組み込まれた特別な行動ルールで動作する、という単純なモデルが採用されている例がほとんどであった。この場合、戦略を変更するには、基本戦術パラメータをすべて交換するか、プログラムを直接書き換えなければならない。戦術の修正や追加を柔軟に行うことも困難となり、環境への適応性が低い。従来のプレイヤーエージェントが採用するモデルでは、外部からのアドバイスを有効に利用できているとは言い難かった。

連絡先: 秋山英久, 東京工業大学大学院総合理工学研究科知能システム科学専攻, 〒226-8502 神奈川県横浜市緑区長津田町4259, Tel: 045-924-5205, akiyama@ntt.dis.titech.ac.jp

そこで、本稿では外部からのアドバイスによって戦略に含まれる戦術の修正や追加が可能なチームモデルを提案する。

### 3. チームモデル

#### 3.1 概要

本稿におけるチームモデルとは、チーム全体の戦略を統合的に管理し、戦術の修正や追加を一貫性をもって行うための枠組を意味する。チームモデルはひとつ以上の戦略を持ち、戦略自体の交換、または戦術の追加を外部から実行できるインタフェースを持つ。戦略と戦術は特定のフォーマットで記述される。

チーム内のプレイヤーは既知の戦略すべてのコピーを内部に持つ。試合中、プレイヤーはそのうちのひとつの戦略を参照する。各プレイヤーはその戦略に従い、戦略に含まれる戦術を参照して状況に応じた意思決定を行う。そして、外部からのアドバイスを受け取ると、すべてのプレイヤーは内部に持つ戦略の変更や戦術の追加を行う。

このチームモデルはスーパーバイザの存在を前提としている。スーパーバイザは環境を観察、分析し、インタフェースを通してプレイヤーにアドバイスを与える。チームを構成するプレイヤーは、指定の戦略フォーマットを理解し、スーパーバイザからのアドバイスを解釈できなければならない。更に、プレイヤーはアドバイス内容に基づいた戦略の変更または戦術の追加を行わなければならない。これらの処理はチーム内のすべてのプレイヤーで共通化され、チームモデルに組み込まれる。

状況に応じた戦術を適用するために、スーパーバイザは適切なアドバイスを生成しなければならない。しかし、有効な戦術を試合中に獲得することは困難であり、戦術の有効性を検証するには事前のテストが不可欠である。よって、戦略に追加可能な戦術は試合前に獲得しておかなければならない。更に、獲得済みの戦術は指定のインタフェースで使用可能な形式に変換できなければならない。

本稿では、戦略へのインタフェースプロトコルとして CLang を用いる。これは、RoboCup の試合ではコーチエージェントのみがアドバイスを許可されているためである。しかし、スーパーバイザはコーチエージェントには限らない。試合前のトレーニングではコーチエージェント以外のスーパーバイザがアドバイスを与えても良い。よって、トレーニング時であれば、特殊なトレーナーエージェントや人間によるアドバイスも可能である。本稿では、このような事前トレーニングによって有効な戦略や戦術の蓄積を行うことを想定している。

#### 3.2 プレイヤーの意思決定

プレイヤーによって実行される行動オプションはチームモデルの構成要素のひとつであり、あらかじめ組み込まれている。CLang でもすでにある程度の行動オプションが分類されており、すぐに利用することができる。

しかし、CLang はプロダクションルール形式で記述されるため、CLang で記述したアドバイスをそのままの形で受け入れると既存のルールの条件を上書きして不具合が発生する可能性が高い。プレイヤーが CLang でアドバイスを受けた場合は、それを緩やかに受け入れることが望ましい。

そこで、本稿ではプレイヤーが意志決定を行う際は一定の評価関数に基づいて行動オプションの評価を行う手法 [Lubbers 1999] を採用する。この手法では各行動オプションの優先度がパラメータ化されており、元の特徴をある程度残し、柔軟性を保ったまま行動の特徴を変更することが可能となっている。

以上のことから、本稿における戦術は各行動オプションの優先度を表すパラメータセットを持つことになる。本研究では更にいくつかのパラメータを追加し、より詳細な戦術を表現する。通常時、プレイヤーはこのパラメータセットを参照して意思決定を行う。そして、スーパーバイザからのアドバイスを受け取ったとき、初めてパラメータの修正を行うことになる。

評価関数とパラメータの修正ルールは全プレイヤーで共通であり、チームモデルに組み込まれている。

#### 3.3 戦術のパラメータ化

従来研究においても、基本ポジションや行動優先度などの戦術パラメータによってプレイヤーの協調行動の特徴づけを実現した例がいくつか存在する [Reis 2001]。本研究ではそれを更に拡張し、戦術の使用条件や戦術が適用されるプレイヤー集合をパラメータとして取り込む。これによって、より細かな戦術のパラメータ化が可能になる。例えば、特定状況下における少人数の連係プレイのような戦術を、元の戦略や他のプレイヤーに影響を与えることなく追加することが可能になる。

本稿では戦術を以下のパラメータセットで記述する。

- 戦術の使用条件  
戦術が適用されるために満たされるべき条件。状態を記述した命題の論理結合によって表現される。状態としては、特定のプレイモード、ボールから一定距離に存在、ある物体が特定の領域に存在、などが使用可能である。
- 戦術が適用されるプレイヤーの集合  
この集合に属するプレイヤーにのみ戦術が適用される。任意の数のプレイヤーを指定できる。背番号による指定や特定の条件を満たすプレイヤーの指定などが可能である。
- プレイヤーごとの基本ポジション  
この戦術でプレイヤーが移動すべき位置座標を決定する。固定の位置座標だけでなく、ある物体に対する相対位置なども指定可能である。
- プレイヤーごとの各行動の優先度  
プレイヤーはパスやドリブルなどの基本行動のオプションをいくつか持っており、このパラメータはその中からどれを優先させるかを決定する。値は  $[0, 1]$  の実数である。
- プレイヤーごとの各行動の積極性  
パスやドリブルなどの行動が積極的か保守的かを決定する。積極性が高ければより攻撃的なプレイを、低ければより安全なプレイを行う。行動の種類ごとに設定される。値は  $[0, 1]$  の実数である。
- プレイヤーごとの各行動対象の優先度  
パスを出すときに相手として誰を優先するか、などを決定する。パスの場合は本人以外のすべてのプレイヤーに対する値を指定する。値は  $[0, 1]$  の実数である。

プレイヤーが持つ評価関数は、これらのパラメータと行動決定時の世界モデルを入力とする。

戦術パラメータは、単体のプレイヤーによる個人プレイの特徴だけでなく複数のプレイヤーによる協調プレイの特徴も記述する。例えば、ゾーンプレスのような協調的な守備戦術 [市川 2004] であれば、“ボールに近い  $N$  体のプレイヤー” というように適用対象が指定される。本稿では個人プレイと協調プレイのいずれも戦術と呼び、区別しない。

戦略にはこの戦術パラメータセットが複数含まれる。ただし、戦略は基本戦術を必ずひとつ持たなければならず、基本戦

術以外の戦術は必ず使用条件を含まなければならない。基本戦術では使用条件と戦術適用プレイヤーの指定が省略され、戦術適用プレイヤーは自動的にチーム内の全プレイヤーとなる。その代わりに、その他のパラメータを省略することはできない。行動決定時、各プレイヤーは自身が内部に持つ世界モデルを用いて戦術の使用条件を検証する。所有する全戦術の中から条件の合致するものが見付からなければ、基本戦術のパラメータに基づいた行動が実行される。条件部に競合が発生した場合は、後から追加された戦術が優先される。

基本戦術以外の戦術では、パラメータの一部を省略することが可能である。省略されたパラメータには基本戦術のパラメータの値が継承して使用される。戦術パラメータセットに含まれるすべての要素を一度のアドバイスで表現すると、アドバイス内容が膨大になる。また、パラメータ自体をアドバイスすることは CLang では認められていない。しかし、アドバイス内容が差分情報のみで良いならば、CLang でも十分に追加戦術を表現できる。これによって、実用範囲内で新しい戦術を追加することが可能になる。

### 3.4 行動の種類

CLang ではサッカープレイヤーとして意味のある条件と行動が既にいくつか分類されている。本稿でも基本的にこの分類に従う。表 1 に分類した行動を示す。これらの基本行動は事前にプレイヤーに組み込まれており、スーパーバイザによって調整されることは想定していない。個々のプレイヤーの動作精度の違いは協調レベルの行動にも非常に大きな影響を与えるため、もし基本行動に修正が加えられれば、それまでに獲得した戦術がうまく機能しなくなる可能性があるからである。

いくつかの行動は更にオプションを持っている。例えば、一回のドリブルの移動距離、安全なパスとリスクはあるが成功すれば有利になるパスの選択、などである。これらのオプションは、行動の積極性のパラメータとして置き換えられる。3.3 節の“各行動の積極性”がこれに対応する。また、パスのように動作の対象が存在する行動に関しては、対象の優先度の指定も行われる。これは“各行動の対象の優先度”に対応する。これらのオプションは行動ごとにあらかじめ組み込まれている。

また、プレイヤーは最大限にアドバイスを受け入れるが、アドバイスで指示された行動が実行不可能な状況であれば、その行動を決して実行しない。例えば、ポジショニング動作に関しては、プレイヤーに十分なスタミナが残っていなければ、移動自体が不可能になり、ほとんどの行動がキャンセルされることになる。

表 1: プレイヤーの行動の種類

ポジショニング動作	キック動作
基本ポジション	シュート
ボール追跡	キープ
スタミナ回復	ドリブル
スペースへの移動	クリア
マーク	パス
プレス	
パスブロック	
シュートブロック	
オフサイドトラップ	

## 4. アドバイスによる適応

### 4.1 アドバイス内容

実際のアドバイスは、“条件部 + 行動の正負 + 行動の種類 (+ 行動の対象)” という形式になる。条件部に関しては、戦術パラメータとしてそのまま適用される。行動の正負は do または dont を値として取り、その行動を促進するか禁止するかを意味する。行動の種類と行動の対象に関しては、行動の正負に応じて優先度の増加か減少のいずれかが行われる。行動の積極性に関しては、CLang に組み込まれた単語だけでは表現できない。そのため、積極性をいくつかのレベルに分け、それぞれのレベルに対応した新しい行動として再定義することで対処する。

また、アドバイスには複数のルールを含めることができる。これによって、ある程度複雑な戦術であっても一度のアドバイスで記述可能となっている。本稿では、ひとつの戦術はひとつのアドバイスで表現される。アドバイス内容が大きくなり一度のアドバイスで表現しきれない場合は、独自単語の定義を行うなどの対処を行う。

### 4.2 戦術の修正と蓄積

有効と判断できる戦術を獲得するには、事前に十分なテストが必要である。しかし、有効な戦術の自動獲得は現在のところ実現が困難である。そこで、本研究では戦術の自動獲得は行わず、人間の教示者がトレーニングを行うことで戦術を獲得する。この場合、人間がコーチに変わってスーパーバイザとして働く。本研究では、人間の教示者からの指示を CLang に置き換え、戦術として蓄積可能なインタフェースの構築を行っている[秋山 2003]。図 1 にその概要を示す。

人間の教示者の情報は CLang へと変換され、各プレイヤーへ与えられる。各プレイヤーはその内容に基づいてパラメータを修正し、内部メモリで一時保管する。そして、パラメータが十分に修正されたら教示者が判断すれば、教示者は与えたすべてのアドバイスを保存する。このアドバイスセットが、ひとつの戦術を追加するためのアドバイスとして使用されることになる。パラメータセットの扱いはプレイヤーに任されるため、教示者がパラメータの詳細を知っている必要はない。教示者はアドバイスで使用可能なプロトコルを知っているだけで良い。

パラメータの修正がチーム全体に渡って行われ、それが十分に安定したならば、そのパラメータセットは基本戦術として使用することが可能である。この場合、その基本戦術を持つひとつの戦略を新たに獲得することになる。基本戦術は純粋なパラメータセットであるため、CLang では表現できない。また、パラメータセットは教示者にとって理解が困難なものかもしれない。そのため、通常の戦術修正とは異なり、修正後のパラメータセットはプレイヤーによって保存される。

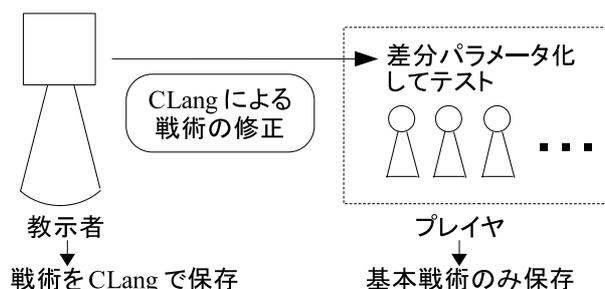


図 1: 戦術の修正と蓄積

### 4.3 戦術の追加

実際の試合では、コーチが利用するのは獲得済みの戦術のみである。コーチは試合を分析し、チームが現在使用している戦略に対して戦術を追加することによってチームパフォーマンスの動的な改善を試みる。コーチがアドバイスを与えると、各プレイヤーは使用している戦略に戦術を追加する(図2)。

アドバイスは複数のルールで構成されているため、プレイヤーはアドバイスを個々のルールの分解し、更にそのすべてを差分の戦術パラメータへと変換する。アドバイスによって追加される戦術はパラメータ化されている必要はない。差分の戦術パラメータと CLang によるアドバイスとは一対一の関係で蓄積されているため、トレーニング時と同じパラメータセットが生成されることになる。

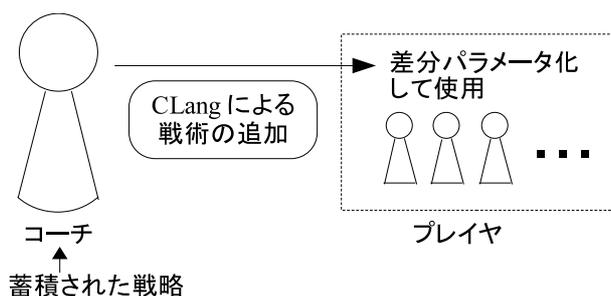


図 2: 戦術の追加

### 4.4 戦略の変更

戦略は基本戦術を主な構成要素とする。基本戦術は差分情報ではないため、情報量が膨大である。そもそもアドバイスがパラメータで与えられるのは不自然であり、CLang の言語仕様でもパラメータを直接表現することは想定されていない。よって、基本戦術だけは事前にパラメータ化されて、知識として共有されていなければならない。チームを構成するすべてのプレイヤーは、試合開始前にこのパラメータセットを取り込まなければならない。

チームが戦略を変更する場合は、使用する基本戦術パラメータセットとデフォルトの追加戦術をすべて交換しなければならない。そこで、コーチによる戦略変更に関しては、チームが所有している戦略の中から適用すべき戦略の識別子を指示のみとする。よって、戦略のアドバイスに関しては独自のフォーマットでアドバイスを送る。戦略変更を行うコーチもチームが所有する戦略についての知識を持つことになる。

## 5. まとめ

本稿では、外部からのアドバイスを受け入れ、戦術を追加することで戦略を修正するチームモデルを提案した。チームに属するすべてのプレイヤーは、統一されたフォーマットの戦略と戦術のパラメータセット、そして共通の行動評価関数とパラメータの修正ルールを持つ。アドバイスによって表現される戦術は差分パラメータであるため、蓄積された戦術を無駄無く試合中に追加することが可能である。これによって、従来は有効活用されているとは言い難かったスーパーバイザからのアドバイスを、有効にプレイヤー内に取り込むことができるようになった。

しかし、差分の戦術の数が増えすぎると、戦術の使用条件を検証するための計算量も増えてしまう。サッカーシミュレーション上のエージェントは実時間処理を求められるため、この

計算負荷がボトルネックとなる可能性がある。戦略に組み込まれる戦術数の調整は解決すべき問題である。

また、コーチ自身の戦術の獲得についても今後解決しなければならない問題である。試合中に有効な戦術や戦略を選択するには、試合状況の判定、特定の戦術の有効性の分析、戦略自体の有効性の分析など、解決すべき様々な問題がある。これらについても、トレーニング時に人間の教示者がコーチに対して教示を行えるインタフェースを用意し、コーチも含めたチームモデルへの拡張を考えている。

## 参考文献

- [Carpenter 2003] Carpenter P. et al.: Integration of Advice in an Action-Selection Architecture, RoboCup2002: The Sixth RoboCup Competitions and Conferences, Springer Verlag, (2003).
- [Jinyi 2003] Jinyi Y. et al: Technical solutions of TsinghuaAeolus for Robotic Soccer, In proceedings of RoboCup 2003 Symposium, (2003).
- [Kaminka 2003] Kaminka A. G et al.: Learning the Sequential Behavior of Teams from Observations, RoboCup2002: The Sixth RoboCup Competitions and Conferences, Springer Verlag, (2003).
- [Riley 2002] Riley P. et al.: An Empirical Study of Coaching, Distributed Autonomous Robotic Systems 5, pp. 215-224, Springer Verlag, (2002).
- [Lubbers 1999] Lubbers J. et al: The Priority/Confidence Model as a Framework for Soccer Agents, Robocup-98: Robot Soccer World Cup II, pp. 162-172, (1999).
- [Noda 1998] Noda, I. et al.: Soccer Server: A Tool for Research on Multiagent Systems, Applied Artificial Intelligence, (1998).
- [Noda 2003] Noda, I. et al.: RoboCup Soccer Server Users Manual for Soccer Server Version 7.07 and later, <http://sserver.sourceforge.net/>, (2002).
- [Reis 2001] Reis P. R. 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).
- [Reis 2002] Reis P. R. et al: COACH UNILANG - A Standard Language for Coaching a (Robo)Soccer Team, RoboCup 2001: RobotSoccer World Cup V, Springer Verlag, (2002).
- [秋田 2001] 秋田 純一, 他: RoboCup シミュレーションリーグ人間参戦システム OZ-RP の提案, 人工知能学会第 12 回 Sig-Challenge 研究会資料 pp. 23-28, (2001).
- [秋山 2003] 秋山 英久, 他: コーチのアドバイスによるマルチエージェント環境への動的適応, 人工知能学会第 17 回全国大会, (2003).
- [市川 2004] 市川 学, 他: RoboCup サッカーにおける協調的な守備戦術の実現, 人工知能学会第 18 回全国大会, (2004).