

5F-01 RoboCupにおける行動予測機能を持つ クライアントフレームワークの実装と評価*

山崎 誠 武田 正之†

東京理科大学大学院 理工学研究科 情報科学専攻‡

1 はじめに

ロボカップ (Robot World Cup Initiative) は、サッカーを題材として広範囲の技術の統合が試されうるような標準問題を与えることにより、様々な技術分野での研究を育てようとする試みである。RoboCup は複数のリーグに別れているが、本研究では計算機上でのシミュレーションによってゲームを行うシミュレータリーグをとりあげる。

シミュレータリーグではゴールキーパーを含む最大 11 人の独立したプレーヤで 1 チームが構成される。このプレーヤに指示を出して試合を進めて行くわけであるが、試合に勝つためには敵や味方の動き、その中でもボールを持っている敵プレーヤの動きを予測する事が重要になる。敵がどのような条件の場合にどのような行為を行うかがわかれれば自分がとるべき行動を的確に決定することができる。本研究では、このような予測の機能をもったプレーヤを作成することを支援するためのフレームワークを Java 言語を用いて構築し、そのフレームワークを用いたプレーヤの実装を通してフレームワークの評価を行う。

2 フレームワークの概要

2.1 機能

作成したフレームワークは次のような機能を持つ。

- サーバとの接続や設定情報の管理
- サーバからのメッセージを解析し情報の加工とオブジェクトの構築
- 敵の動きの予測

- 敵プレーヤの戦略の分析

2.2 構造

今回実装したフレームワークは、図 1 に示すような機能ごとのいくつかのレイヤに分けられている。

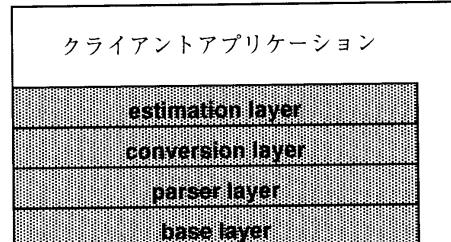


図 1: フレームワークの構造

それぞれのレイヤと機能は以下の通りである。

● base layer

コマンドラインオプションの解釈、サーバとの接続、サッカーサーバの設定情報の管理、思考ルーチンの定期的な呼び出し

● parser layer

サーバからのメッセージを解析し、メッセージの種類に応じたオブジェクトに変換

● conversion layer

視覚情報から自分の絶対位置の計算、相対座標系と絶対座標系の相互変換

● estimation layer

敵プレーヤの動きの予測。コーチクライアント¹と連携することで、より正確な予測を行うことも可能

*Implementation and Evaluation of a RoboCup client framework with estimation

†Makoto Yamazaki and Masayuki Takeda

‡Dept. of Information Sciences, Science University of Tokyo

¹フィールド上の全ての物体の位置情報を取得できる非プレーヤクライアント

それぞれのレイヤは下位のレイヤのみに依存しているのでサーバとの通信を行う基本部分のみを利用して、他の部分は独自に実装することも可能である。

3 予測の概要

3.1 予測の基本戦略

ボールをキープしているプレーヤの動きを予測する場合、そのプレーヤのとり得る次の行動としては、ドリブル、パス、シュートが考えられる。プレーヤが自分の動作を決定する際のパラメータとしてはボールとの距離、ゴールとの距離、ゴールからの角度、味方のプレーヤとの距離と角度、敵のプレーヤとの距離と角度のうちの一部もしくは全てであると仮定する。

ボールをキープしている敵のプレーヤが、どのような状況でどのような行動を選択したかを記録していく。この状況とその後の敵の行動の組を以降イベントと呼ぶことにする。もし周囲の状況のみによって動作を選択しているならば、記録されている状況と同じ状況が発生した場合、その時敵がとった行動を今回も繰り返すことになる。

この事から敵の次の行動の予測を行う。

3.2 戦略の学習

基本的には現在の状況とその時の敵の行動を記録していくべきよいのであるが、単に全て記録するだけの方法はいくつか問題がある。

一つ目の問題は視覚情報が送られてくる度に情報が増えしていくので、単に過去の情報全てと比較を行う方法では予測を行う際に時間がかかるてしまうことである。この問題を解決するために、現在イベントの構造化などいくつかの方法を検討中である。

二つ目の問題は記録した状況(ゴールとの距離や角度、敵との距離や角度等)のうちの全てを用いて敵が行動を決定しているとはかぎらないということである。そこで敵が行動を決定する際に利用していないと思われるものを発見することができれば過去の状況と完全に一致しない場合でも正確な予測が行われる可能性が高くなる。また探索するイベントの数を減らすことができるので予測時間の短縮にもつながる。

最近のRoboCupシミュレータリーグにおいては、試合中のコーチクライアントの使用が認められている。そこでコーチクライアントがフィールド上の全てのプレーヤとボールの情報を正確に得ることができる点を利用してもより正確な戦略分析が可能になる。

4 実装について

本フレームワークの開発はBlackdown版のJDK1.2を用いてVine Linux上で行った。ソースコードはSunのJava Language Specificationに準拠している。各クラスはjp.ac.sut.noda.is.mt.zak.FKTに2.2に示した各レイヤー毎にそれぞれパッケージとして分類されている。Intel Celeron Processor 450MHzの環境においてSoccer Serverのシミュレーションサイクル(10msec)毎に最低1回の思考ルーチンを実行することが可能であることが確認された²。

5 おわりに

3.1では予測の際に、敵のプレーヤはその時点での周囲の状況だけから行動を決定すると仮定した。もし敵のプレーヤが過去の経験を生かして行動を決定している場合、この方法での予測はあまり意味を持たない。また、動作の決定にランダム性が含まれる場合も正確な予想は難しくなる。

プレーヤを新たに作る場合、サーバへの接続、設定情報の管理、メッセージの解析、相対座標から絶対座標への変換など決まりきった部分がかなり存在する。これらの部分を一から作成するのではなく、既存の物を利用できるようにすることでRoboCupシミュレータリーグに新たに参加しやすくなる。また、思考ルーチンを組み込むだけでプレーヤとして機能するので、さまざまな思考ルーチンの実験を手軽に行うことができる。

本フレームワークは本学情報科学科3年次の演習にも利用を予定している[2]。

参考文献

[1] Soccerserver Manual Ver.5

<http://www.dsv.su.se/~johank/RoboCup/manual/download.html>

[2] 杉本徹, 武田正之, 奥乃博: RoboCupシミュレータ・リーグを大学3年次の演習課題に,
ロボコンマガジン, No.6 1999

²ただしGCなどの要因により常にこれが保証できない場合もあり得る