

RoboCup3DSimulation 環境におけるエージェント開発支援の為のデバッグプレイヤーの設計と実装

荒川 篤邦[†] 瀧澤 崇[†] 久保 長徳^{††} 高橋 勇[‡] 黒岩 丈介[‡]
 小高 知宏[‡] 小倉 久和[‡]

[†]福井大学 工学部 ^{††}仁愛大学 人間学部 [‡]福井大学 大学院

1 はじめに

本研究では, RoboCupSoccer3DSimulationLeague におけるエージェント開発を支援するシステムについて検討する. Simulation League では, 選手エージェントの行動をプログラムし, 計 22 体のエージェントが, SoccerServer と呼ばれる仮想フィールドを再現するサーバーと通信を行う事で試合を行う. また, 動作の様子は SoccerMonitor と呼ばれるツールを用いることで確認できる. エージェントの開発者はエージェントをプログラミングした後, サーバーに接続させて単体の動作確認をするが, SoccerMonitor では動いている様子を確認する事しかできず, エージェント開発者が意図しない行動をした場合に原因を特定し難く, エージェントの修正に大きく時間を費やしてしまうこともある.

本研究では, 試合後にサーバーが生成する, 各シミュレーションステップにおけるエージェントの座標等が記述されているモニターログの他に, エージェントに組み込んで任意のメッセージを生成させて出力ができる開発ユーザログ機能を実装した. それらを利用して, 各ログの情報を全て 2 次元フィールド上に可視化するデバッグプレイヤーを作成する. このプレイヤーはエージェント開発を支援する機能を備えている.

2 デバッグプレイヤーの構成

2.1 ログファイル

本研究では 3 種類のログファイルを使用する. モニターログは試合毎に SoccerServer が生成し, シミュレーションステップ毎のエージェントやボールの座標とプレイモード, スコア等が記述されている. モニターログはサーバーが直接生成したログで, その座標は正確である. ユーザーログはデルログと行動ログの 2 つからなる.

ワールドモデルは, ある時点でエージェントが見ているサッカーフィールドのモデルである. サーバーからは, 自分から見た他のエージェントやボール等のオブジェクトへの相対角度と距離の情報が送られる. 各エージェントには視界が設定されており, 視界の中に入っているオブジェクトについてだけがエージェントに送られる. エージェントはその情報を基に各オブジェ

クトの座標を計算することができ, ワールドモデルを形成する. エージェントが意図しない行動をする原因の一つとして, 間違ったワールドモデルを形成しているということがある. ワールドモデルログを利用すれば, エージェントがどのようなモデルを保持しているかを確認することができる.

エージェントの行動ログは, エージェントがどのような行動をしたか記録する. エージェントの行動コマンドとしては Kick, Move 等が用意されている. しかし, どのコマンドが選択されたか, 特に Kick に関しては SoccerMonitor ではなかなか見分けが付きにくく, いつ Kick コマンドが選択されたかが分かりづらい. コマンドにも言える. 行動ログは, 例えばエージェントが "Kick" という行動を選択すれば "Kicking" というアクションメッセージや, 別のエージェントへパスを試みたのならば, "Pass to No. ○" というアクションメッセージを出力させたりと, 開発者は自由にメッセージを決定できる.

ワールドモデルログと行動ログは, それを出力するためのプログラムを作成した. このプログラムをエージェントのソースプログラムの中に直接埋め込んでログを生成する.

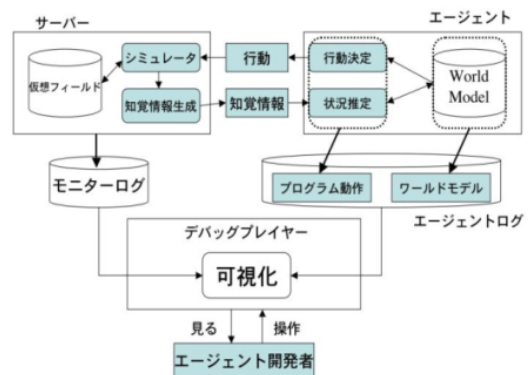


図 1: システム構成図

2.2 可視化機能の実装

これらのログを利用して, 実際に試合を再現する. 図 1 は本システムと SoccerServer, エージェント開発者との関係を示した図である. サーバーとエージェント

がそれぞれ出力したログを、デバッグプレイヤーで読み込み可視化することで、エージェント開発者がログの内容を把握できる。試合の再生にはモニターログを用いる。シミュレーションステップ毎に各エージェントとボールをフィールドに配置し、それを連続することによって再生する。モニターログを表示した上にワールドモデルログ、行動ログの情報を表示する。ワールドモデルログの各オブジェクトの座標を用いて、フィールド上に各オブジェクトを配置する。この配置はモニターログによる配置とは位置情報の精度が変わる、一般には異なる。行動ログのアクションメッセージは、モニターログによって配置されているエージェント付近に表示させる。これらの状況を把握し易くするために、2次元フィールド上に描画する。SoccerMonitorは3次元表示であるが、そのためにエージェント間の位置関係などが非常に把握しづらいからである。

この3つを同時に見ることでエージェントの動作に関して多くのことが把握できる。例えば、モニターログで描画されたオブジェクトとワールドモデルログによって描画されたものの位置のずれが大きいならば、エージェントがワールドモデルを形成する際の計算方法が間違っていたり、精度が低いということがわかる。

3 システムの実装

本システムのログ解析部は、主にRubyにより実装した。これはRubyがテキスト処理能力に優れているためである。始めにログファイルを解析・整理し、それをフィールド表示部に渡す。ログは、各オブジェクトの座標や行動ログのアクションメッセージをシミュレーションステップ毎にまとめ、整理する。これによってシークバーなどによる巻き戻し、早送り、再生スピードの調整等を実現する。RubyにてGUIを作成するために、表示部はruby-gnomeを利用した。これによりGtk+をRubyから利用することができる。表示部では解析後のデータを基にシミュレーションステップ毎にオブジェクトを描画し、ムービーとする。各エージェントのワールドモデルは、選択されているエージェントだけ表示する。これは複数エージェントのワールドモデルを重ねると画面が非常に見にくくなるからである。

実装したシステムのインタフェース画面の例を図2に示す。画面中央にフィールドを表示して、そこにオブジェクトを配置した。画面下部には各種操作ボタンをまとめる。ここでは、再生・巻き戻し・早送りの他に再生スピードを操作することもできる。



図 2: 本システムのインタフェース画面

4 考察

本システムでは、サーバーログとエージェントが生成する3種類のログを用いることによって、エージェントの判断・行動と、プログラムとの対応関係を把握することができるようにした。エージェントのワールドモデルとモニターログを比較する形をとることにより、エージェントがその時に認識している状況が正確であるかどうかの判断や、様々な自己位置同定手法を実装した際の動作確認を支援することが可能となった。さらに、行動ログを用いて、エージェントの行動毎にメッセージを出力させることを可能にしたことにより各エージェントがそのときどのようなルールを発火させ、どのような行動を選択したのかを比較的簡単に把握できるようになる。

今後の課題として、当面は表示部の機能を充実させる必要がある。拡大・縮小機能はその一つである。全体を見ることも重要であるが、一部のエージェントを局所的に観察するほうが必要なこともある。ユーザーが任意に選択した部分を拡大する機能はそのような時に有用である。将来的な課題としては、エージェントの行動予測等があげられる。サッカーではある程度ボールや相手の行動を予測して自分の行動を決定する。エージェントも他のエージェントがどのように移動しているかというようなことを予測し、行動させることがある。各シミュレーションステップでエージェントがどのような予測をしたかということを知ることは、行動ルールを見直したりする際に大きな助けとなるはずである。

参考文献

- [1] ロボカップ日本委員会 監修 高橋友一, 伊藤暢宏 『RoboCupではじめるエージェントプログラミング』 共立出版 2001
- [2] ロボカップサッカー シミュレーション 2D リーグ 必勝ガイド 秋山英久ロボカップ日本委員会 監修 秀和システム 2006