

ロボカップサッカーエージェントにおける行動ルール表現の設計と実装

久保長徳† 滝澤崇† 河原林友美‡
†福井大学工学部

黒岩丈介† 小高知宏† 小倉久和†
‡福井高等専門学校

1 はじめに

RoboCup サッカーシミュレーションは、マルチエージェントや機械学習のテストベッドまたコンピュータサイエンスの教材として注目されており、様々なチームやエージェント、フレームワークが提案されている。しかし提案されている多くのエージェントやフレームワークを用いてもエージェントの開発が困難であるのが現状である。

エージェント開発が困難である理由の一つは、エージェントを実装するプログラミング言語にあると考える。サーバとの通信などの低レベルな部分と戦略や戦術など高レベルな部分を同様に記述することに無理があるのではないかと考えている。例えば戦略や戦術などは専用の言語を用意して、記述した方が開発しやすいのではないかと考えた。

本研究ではエージェントを記述するための行動ルール記述言語、記述されたエージェントを動かすためのフレームワーク、効率的に開発するための開発支援環境の開発を行なう。エージェントフレームワークでのエージェント開発はマルチエージェント環境および知識再利用を前提としており、そのためのプログラミングモデルの紹介も行う。本研究によりエージェント開発の負担軽減、知識の再利用の実現が期待できる。

2 RoboCup エージェント開発の現状

RoboCup サッカーシミュレーションシステムは当初 UNIX システム上に C++ で実装された。その影響(ソー

言語	ライブラリ数
C,C++	13
Java	8
Prolog	2
Lisp	1

表 1: 言語による公開ライブラリの数
(<http://www.ida.liu.se/~frehe/RoboCup/Libs>)

Design and Implement of Soccer Agent Programming Language

†Takenori KUBO, †Takashi TAKIZAWA, ‡Tomomi KAWARABAYASHI, †Josuke KUROIWA, †Tomohiro ODAKA, †Hisakazu OGURA

†Faculty of Engineering, University of Fukui

‡Fukui National College of Technology

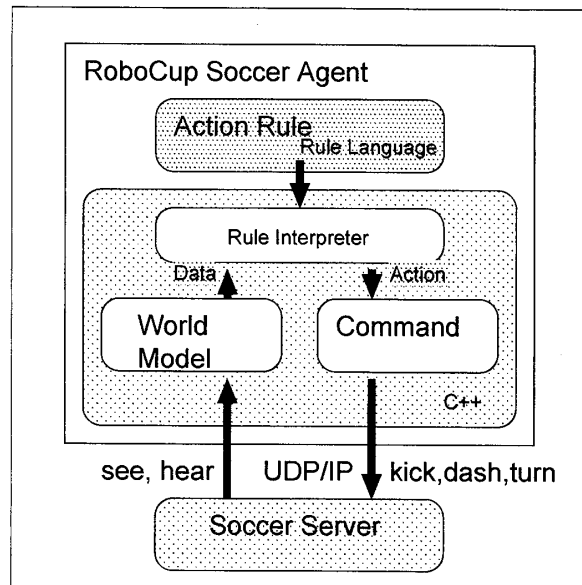


図 1: エージェント構造

スコードの流用など)もあって、多くのエージェントやライブラリは C,C++ で実装されている (表 1)。特に CMUnited[1], UVA Trilearn[2], Helios[3] などの強豪チームが C++ で実装され基本部分のソースコードを公開したことで、C++ での実装が一般的となっている。C++ 言語によるエージェント開発における利点として、既存のエージェントやライブラリが利用できること。エージェントプログラムの処理速度の高速化。がある。逆に問題として、コンパイル時間などのロスタイムが開発作業のリズムを狂わし快適な開発を妨げることが考えられる。

C++ に限らず一つの言語でエージェント全体を実装した場合、エージェントがサーバと毎ターンごとに 0.1 秒間隔で通信を行なうため、全ての手続きは 0.1 秒以内に終了しなければならない。このためドリブルやパスなどの複数ターンに渡る行動を実装する場合には、単純な行動の繰替えしであってもソース上では単純に表現することが困難である。Helios では動作クラスや意図クラスを定義することで、CMUnited や UVA Trilearn ではコマンドキューという仕組みを用いて複数ターンの手続きを実装している。

そこで、サーバとの通信や基本的な行動知識などを C++ で実装し、行動ルールをルール記述言語で記述す

属性	機能	備考
コマンド	基本コマンド コマンド	kick, dash, turn pass, shoot, centering
ワールドモデル	位置情報 各種位置関係	選手、ボール、ゴールの相対座標 推定される絶対座標 ボールに近いプレイヤー、自分に近いプレイヤーなど
自動処理	メインループ ポジショニング 視線のコントロール	サーバとの通信、ワールドモデルの更新、ルール実行
動作予約	CommandQue, 動作意図クラス	制御構文実装用

表 2: C++で実装する機能

ることで、行動ルールの開発に適した開発環境を検討する。

関連する研究としては OZ-RP[4] がある。OZ-RP では、行動ルールの代わりに人間プレイヤーが GUI を使ってエージェントを操作することでエージェントプログラムを実現している。

3 エージェント開発環境

開発環境はエージェントプログラムと行動ルール開発ユーティリティから構成される。エージェントプログラムを再コンパイルすること無く、行動ルールの開発を中心とした作業スタイルを想定している。

3.1 エージェントプログラム:基本機能

エージェントプログラムは C++ で実装されており、サッカーサーバとの通信機能と行動ルールインタプリタ機能を持つ (図 1)。エージェントプログラムは行動ルールを記述したテキストファイルを読みこんで、そこに記述された行動ルールを実行する。必要であればファイル読み込み時に中間コードを生成し、処理速度の向上を目指す。エージェントプログラムに実装される基本機能を表 2 に示す。

3.2 エージェントプログラム:ルール記述言語

ルール記述は 2. コマンドを実行順に記述する。コマンドについては利用頻度の高いものは必要であれば基本機能に追加していく。

ある程度複雑なルールを記述するために if, while 等の制御構文と手続定義の文法を備える予定である。制御構文や手続定義は CommandQue や動作意図クラス相当の仕組みを用いて実装する。

リードオンリーな変数として表 2. ワールドモデルを利用でき、if や while の条件文やコマンドの引数とし

て利用する。ワールドモデル以外の変数は無く、変数へ代入、変数間の演算は予定していない。

3.3 行動ルール開発ユーティリティ

ユーティリティとしてデバッガと GUI ルールエディタを検討している。

4 まとめ

RoboCup サッカーエージェントプログラミングに関して、行動ルールインタプリタを内蔵したエージェントを提案し、行動ルール開発環境についての紹介した。

今回報告できなかった本開発環境の詳細については発表当日に紹介する。

参考文献

- [1] Stone Peter, Veloso Manuela, and Patrick Riley. *RoboCup-98: Robot Soccer World Cup II*, Vol. 1604 of *Lecture Notes in Artificial Intelligence*, chapter Champion Teams: The CMUnited-98 Champion Simulator Team, pp. 61–76. Springer, 1999.
- [2] Remco de Boer and Jelle R. Kok. The incremental development of a synthetic multi-agent system: the UvA Trilearn 2001 robotic soccer simulation team. Master's thesis, University of Amsterdam, The Netherlands, February 2002.
- [3] 秋山英久. ロボカップサッカーシミュレーション 2 Dリーグ必勝ガイド. 秀和システム, 2006.
- [4] 西野順二, 久保長徳, 秋田純一, 下羅弘樹. Oz-rp システムを用いたバーチャルサッカーでの人間協調行動の観測. 第 11 回インテリジェントシステムシンポジウム講演論文集, pp. 149–152, 2001.