

RoboCup サッカー 2D のための 簡易プログラミング環境の構築

江成 駿† 木下 克也‡ 飯塚 泰樹†

† 東海大学理学部情報数理学科 ‡ 東海大学大学院理学研究科数理科学専攻

1 はじめに

RoboCup サッカー 2D は, RoboCup の競技で最も古いリーグであり, マルチエージェントシステムのテストベッドとして多岐に利用されてきた [1]. そのため近年の RoboCup サッカー 2D は, 古くからの研究により複雑なプログラムが使われており, プログラミング初心者の RoboCup 参入意欲を下げている. 我々は, RoboCup サッカー 2D へ関心を持つ学生の参加を促進し, マルチエージェントの学習支援を目的として, 簡易にサッカーエージェントプログラムを開発できる環境を構築した. 本稿では, 構築した環境の詳細と評価を報告する.

2 研究目的

RoboCup サッカー 2D では, エージェント単体の情報だけでなく, チームメイトや敵エージェントの情報も考慮しながらプログラム開発を行うため, ハードルが高い. 秋山らは, 毎年勝ち続けている HELIOS のベースプログラムである「agent2d」を公開してきた [1]. ユーザは, agent2d のプログラムに修正・追加を行う事でエージェント開発が可能になる. しかし, agent2d のソースコードそのものが長大で難解になっている.

西野らは中高生向けのプログラミング教育を目標にしたバーチャルサッカーロボットキット「OZED システム」を構築した [2][3]. OZED のプログラミングは, 直感的なエージェントの位置設定と, エージェントの行動をコーディングする 2 段構成になっている. しかし, 直感的なプログラミングからエージェントの行動に関するコーディングへのステップアップは敷居が高い.

本研究では, プログラミング言語を学習中のユーザのサッカーエージェントプログラム開発を目的に簡易プログラミング環境の構築を進めてきた. この環境は, プログラミング初心者だけでなく中級者まで, プログラミングスキルに応じた開発を段階的に行える環境を目標に構築した.

3 スクリプト言語を用いたプログラム開発

3.1 スクリプト言語

本研究では, エージェント開発用言語にスクリプト言語 Lua を用いた. スクリプト言語は, インタプリタで処理される事が多く, 動的型付け及びコンパイル時間短縮の特徴を持つ. これらの特徴はプログラムの開発期間を短くし, ソフトウェアの生産性を高める事に貢献する.

スクリプト言語の中でも Lua は, 他のスクリプト言語に比べて処理速度に優れ, 修得しやすい言語である. また Lua は, C/C++ と連携が容易である. 本研究において, Lua を agent2d に埋め込む事で, エージェント開発の単純化を目指した.

3.2 環境構成

本研究では, サッカーエージェント開発環境として「yapEd」を開発した. エージェント開発を簡単にする事のみ追求した場合, 自由度が小さくなり, 似たようなサッカーチームしか作れずマルチエージェントの研究効果を下げってしまう. マルチエージェント教育を促進するためには, コーディングの容易性と自由度のバランスが保たれてなければならない. この課題を解決するため環境構築において, 以下のツールを開発した.

(1) Visual Editor

直感的なエージェント開発が可能であり, プレイヤーがおかれている試合の状況に対して, パラメータを調整可能なツールとして用意した. エージェントやサーバの起動, フォーマーションの設定といった RoboCup を扱うのに必要な機能もあるため, プログラミング言語を知らないユーザでもチーム開発及びテストが可能である.

(2) AgtSC Lua

エージェント単体の開発を目的としたプログラムである. プログラムは, Lua で記述されており, エージェントに与える行動や行動に伴うパラメータを設定できる.

Yet Another Programming Environment for RoboCup Soccer 2D

†Shun Enari ‡Katsuya Kinoshita †Yasuki Iizuka

†Department of Mathematical Sciences, Tokai University

‡Graduate School for Sciences, Tokai University

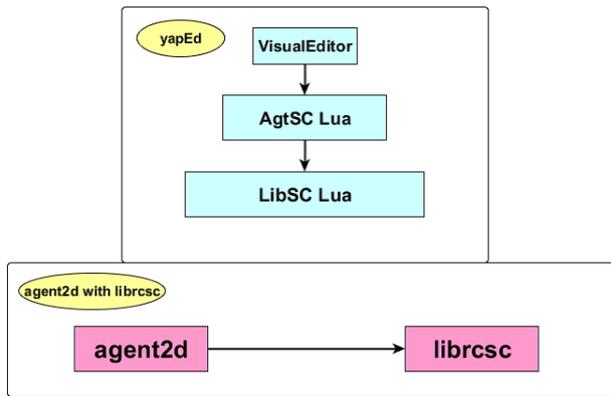


図 1: エージェント開発モデル

(3) LibSC Lua

エージェントの行動詳細が記述されたライブラリプログラムである。具体的には, agent2d with librcsc をベースにした Lua プログラムとして用意した。LibSC Lua は, AgtSC Lua に比べてプログラム規模は大きい, 記述自由度が高く, チーム強化が期待できる。

これらの開発物を yapEd として統合し, agent2d with librcsc と組み合わせた(図 1)。プログラミングは, 3 段階構成のため, ユーザのプログラミングスキルに応じてエージェント開発が可能である。

4 構築した環境の評価

プログラム評価は, 東海大学理学部情報数理学科の学部学生を対象に行った。評価方法は, 学生に対し, yapEd の導入及び Lua のエージェントプログラムとエージェントの行動に関する関数仕様が記載された資料, agent2d のプログラムを提示し, 対面式及び記述式のアンケートに回答してもらった。

その結果として, yapEd を用いたエージェント開発は容易との評価を得た。Lua でのエージェント記述が単純で, スムーズに開発が行えるとの意見も受けた。agent2d に記述されているヘッダーファイルの導入部分や複数行にまたがる条件式を隠蔽したためと考えられる。実際にプログラムの開発を行うと, C/C++ のエージェントプログラム 82 行に対し, Lua では 21 行で同じ能力を持つエージェントが開発できた(表 1)。プログラム例を図 2 に記載する。アンケートでもエージェント記述は, 「Lua のみで行いたい」が 27%, 「Lua で行いたいだが C/C++ の開発にも興味がある」が 64%, 「C/C++ で行いたいだが Lua にも興味がある」が 9%, 「C/C++ のみで行いたい」が 0% との結果を得ている。本方式の課題はエージェントを記述するために, Lua の修得が必要な事である。Lua

表 1: コード比較表

	Lua	C/C++	Code ratio
Agent	21 lines	82 lines	3.9
Library (move module)	52 lines	99 lines	1.9

```
require("robocup")
function mainLoop()
  -- キックできる状態か
  if(isKick())then
    Kick() -- 蹴る
  else
    Move() -- 動く
  end
end
function Move()
  basic_move() -- 基本の行動を実行
end
function Kick()
  -- 基本のキックを行うかどうか
  if(not basic_kick())then
    offensive_kick() -- 攻撃用キックを実行
  end
end
```

図 2: プログラム例

を修得していなければ, プログラミングのスキルに関わらず, エージェントのカスタマイズが難解に感じるとの意見を受けた。初心者にとって新しい言語を用いる事は, プログラム開発の意欲を下げってしまう事がわかった。

5 おわりに

本研究において, エージェント開発を容易にするために yapEd を構築した。ユーザにスキルに応じて, プログラミングが可能のため, RoboCup に関わる機会が増え, ユーザのプログラミングスキル向上も期待される。

参考文献

- [1] 秋山 英久: ロボカップシミュレーション, 人工知能学会誌, 26 巻 2 号, pp.204-207(2011)
- [2] Junji NISHINO, Takenori KUBO, Hiroki SHIMODA, Tomoharu NAKASHIMA : Educational Soccer Simulation system OZED, SCIS&ISIS2006 @ Tokyo, Japan, pp.63-68(2006)
- [3] 西野 順二: このへんファジィとそのあたり, 知識と情報(日本知能情報ファジィ学会誌), Vol.20, No.5, pp.776-784(2008)