

RoboCup Rescue Simulation 新規参入者のための エージェントフレームワークについて

高柳 和央 *¹ 鷹見 竣希 *² 幸塚 義之 *¹ 伊藤 暢浩 *²
Kazuo Takayanagi Shunki Takami Yoshiyuki Kozuka Nobuhiro Ito

*¹愛知工業大学大学院経営情報科学研究科
Graduate School of Business Administration and Computer Science

*²愛知工業大学情報科学部情報科学科
Faculty of Information Science in Aichi Institute of Technology

As one of a variety of approaches to recent large-scale natural disasters, there is RoboCup Rescue Simulation. Current, information on the development of agent is poor. So, it's too difficult for new entry to develop. In order to solve the problem in this paper, the design and implementation of agent framework was presented. Particularly, it was designed to consider the readability and reusability of the developed agent.

1. はじめに

近年の大規模な自然災害に対する様々な取り組みの1つとして、RoboCup Rescue Simulation(以降、RCRS)がある。RCRSは、仮想的な災害空間における災害救助シミュレーションを、マルチエージェントベンチマークとして提供し、AIの研究の促進と、その成果による社会貢献を目指している。

しかし現在のRCRSは、年々複雑化するシステムに対し、情報の更新が遅れているため、新規参入が困難な状況である。そこで本研究では、エージェントの開発環境として、エージェントフレームワーク(以降、フレームワーク)の設計及び実装をおこない、新規参入の問題を解決することを目的とする。

2. RoboCup Rescue Simulation について

2.1 RoboCup Rescue Simulation の概要

RCRSは、仮想都市上に、地震による建物の倒壊や火災、道路の閉塞を発生させ、それに対し災害救助隊(以降、エージェント)が、火災の消火や道路閉塞の啓開、市民の救助を行うシミュレーションである。エージェントは、救急隊、消防隊、土木隊、各部隊を統括する司令部の6種類に分けられる。

RCRSでは、各エージェントの自律行動による災害戦略や協調動作を通して、減災の効率化を目指している。救助効率はスコアとして数値化され、毎年行われる競技会で優秀なスコアを残したエージェントのソースコードは、Webで公開される。

2.2 エージェント開発の現状と問題について

新規参入者が現状のRCRSでエージェントの開発をおこなうには、主に以下の3つの問題が挙げられる。

- 開発情報の不足に関する問題
- ソースコードの共有に関する問題
- 特定分野の研究に関する問題

まず、現在の仕様合ったエージェント開発環境の構築方法やエージェント自体の開発方法など、エージェントの開発情報が不足している。開発情報は、古いバージョンのマニュアルや大会のコードサブミッションガイドライン、公開されているソースコードなどから得られるが、現在の仕様と違う場合も多

い。さらに公開されているソースコードは、チーム毎に独自の実装がされているため、ソースコード間の互換性が低く、仕様の理解に理解に莫大な時間が必要となる。

またRCRSには研究課題として、経路探索や資源割り当てなど多くのアルゴリズムと、それらの複合問題であるリアルタイムプランニングなどが存在する。しかし現状では、経路探索など特定分野の研究のためのエージェントは、各自で用意する必要がある。つまり導入の段階で、莫大な時間が必要となり、新規参入が難しくなっている。

3. エージェントの開発支援手法の設計

3.1 エージェントの開発支援手法の設計

本研究では、2.2節で挙げた問題の内、特に特定分野の研究に関する問題とソースコードの共有に関する問題を解決するため、フレームワークの導入を提案する。フレームワークの設計は、以下の2点を中心におこなう。

- 可読性の高いエージェントの設計
- 基本的な動作を行うエージェントの設計

3.2 可読性の高いエージェントの設計

RCRSのエージェントには、複数の継承元のクラスに必要な変数やメソッドが定義されているなど、冗長な部分が存在する。これに対し、図1のように2つの部分で構成されるエージェントを設計することで、冗長な部分の最適化や可読性の向上を図る。

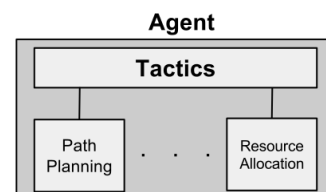


図1: フレームワークのエージェント構成図

図1中のTacticsは、エージェントの動作を決める行動アルゴリズムを記述するクラスである。また図1中の経路探索や資源割り当ては、各アルゴリズムをモジュール化するためのインターフェースを示している。

連絡先: 高柳和央, 愛知工業大学, 愛知県豊田市八草町八千草1247, TEL:(0565)48-8121, FAX:(0565)48-0277, uranos.maslab@gmail.com

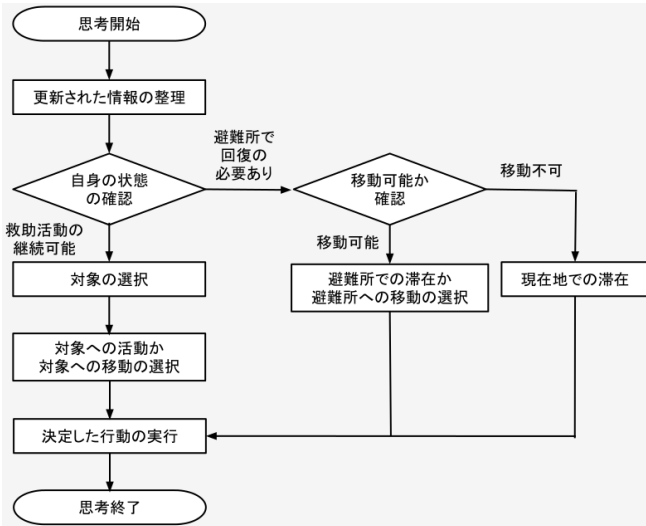


図 2: 基本的な行動アルゴリズムのフローチャート

3.3 基本的な動作を行うエージェントの設計

特定分野の研究に対しては図 1 中のインターフェースを用いる, 基本的な行動アルゴリズムの実装した Tactics を用意することで, アルゴリズムの比較の単純化を図る. 図 2 は, 基本的な行動アルゴリズムのフローチャートである.

4. フレームワークの実装

4.1 フレームワークの実装

3章で挙げた設計の実装を, 以下の4点を中心におこなった.

- Tactics の実装
- モジュール化のためのインターフェースの実装
- Communication Library の使用と改良
- 基本的な動作をおこなう Tactics の実装

4.2 Tactics の実装

エージェントは図 3 のように, 行動アルゴリズムを実装する `think` メソッドや自身の ID など, 必要なメソッドや変数が複数の継承元に別れて定義されている. また `think` メソッドは, 決定した行動を明示的に返す仕様ではないため, 現在のエージェント開発に最適ではない.

これに対し Tactics では, 図 3 のように複数の継承元から必要な変数やメソッドを集約する実装をおこなった. また `think` メソッドを最適化するため, 図 3 の右下のような Action クラスのインスタンスを返す実装へ変更した.

4.3 モジュール化のためのインターフェースの実装

本研究では主に, 経路探索と救助対象の選択のアルゴリズムに対して, モジュール化のためのインターフェースの実装をおこなった. 経路探索を実装するインターフェースである `RouteSearcher` には, ダイクストラや RTA* などのアルゴリズムを実装できる. 同様に, 救助対象の選択をおこなうアルゴリズムを実装する `TargetSelector` には, オークションアルゴリズムなどを実装することが可能である.

4.4 Communication Library の使用と改良

エージェント同士が通信するための通信プロトコルも, 各自で実装する必要がある. これに対し, 先行研究である `Communication Library`(以降, `Comlib`) [3] を使用することで, 実装をおこなった. これに合わせ, `Comlib` の実装を見直し, イベ

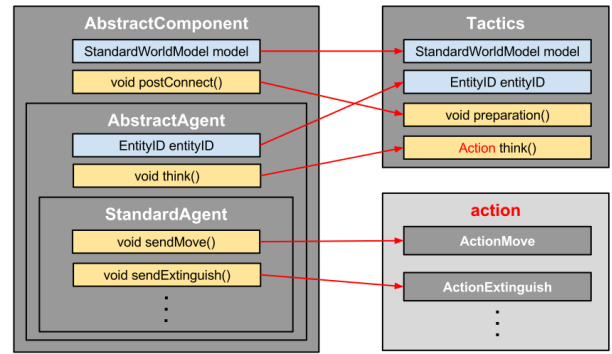


図 3: Tactics による仕様の最適化

ント駆動化の実装や現在の仕様への最適化, メッセージに関する研究への対応をおこなった.

4.5 基本的な動作をおこなう Tactics の実装

経路探索など特定分野の研究の単純化のため, 基本的な動作をおこなう Tactics を実装した. 4.3 節のインターフェースを用いて実装したアルゴリズムを, アルゴリズムの比較が可能になっている.

5. 提案手法の評価と考察

5.1 提案手法の評価と考察

フレームワークの妥当性の確認や問題点の洗い出しのため, RCRS でのエージェント開発の初心者を中心に, フレームワークとマニュアルに関するアンケートをおこなった.

5.2 評価

アンケートにより, 主に以下のような意見が得られた.

- フレームワークなどの導入が簡単だった
- 戦術として, Tactics などの分割方法が良かった
- クラスやメソッドの詳しい説明が欲しい

5.3 考察

アンケートより, Tactics などによって, エージェントの開発方法が明確になったという意見を得られたので, フレームワークの設計や実装は妥当であると考えられる. しかし, クラスの詳しい説明が足りないなどの意見があったため, マニュアルの内容を改善する必要があると考えられる.

6. おわりに

本研究では, 新規参入者のためのフレームワークの設計及び実装をおこなった. 実装では仕様の最適化などをおこない, アンケートではフレームワークの有効性を確認できた.

今後はアンケートで得られた不満点の改良や GUI を用いた開発補助ツールなどの研究をおこない, アルゴリズムをより簡単に実装できるエージェント開発環境の提供が望まれる.

参考文献

- [1] "RoboCupRescueSimulation"
<http://roborescue.sourceforge.net/>
- [2] "ロボカップ日本委員会" <http://www.robocup.or.jp>
- [3] 尾橋大, 川口裕貴, 堀部省伍, 太田健文, 伊藤暢浩, 鳥海不二夫, "RoboCup Rescue Simulation のための情報共有ライブラリ", (2013 年 2 月)