

コンポーネント指向型マルチエージェントシミュレータ Arisco の評価に関する研究

多田 龍之介[†] 和泉 信生[†]

崇城大学 情報学科[†]

1 はじめに

従来広く使われているマルチエージェントシミュレータ(以降 MAS)はオブジェクト指向の継承を用いたモデルを利用しているが、再利用性に難がある。本研究室ではコンポーネント指向型に基づいた MAS:Arisco を提案している。

Arisco はエージェントの行動をオブジェクト間の結合を疎にしたコンポーネントの組み合わせで定義するモデルを用いており、再利用性の面で有利であると考えられる。しかし、従来のオブジェクト指向を用いた MAS に対して Arisco が有効であることが十分に評価されていない。

本研究の目的は Arisco の再利用性の定量的な評価を行うことで、従来型の MAS との比較を行う。

2 コンポーネント指向型の利点

Arisco は従来の MAS と違いコンポーネント指向型の MAS である。ここで、火災が起きた際の建造物からの避難シミュレーションを例として両者の定性的な比較により、コンポーネント指向型の利点を述べる。

従来のオブジェクト指向型の MAS ではエージェントクラスを継承して避難者クラスを、避難者クラスを継承して誘導者クラス、救助者クラスを作成する。(図 1 左)これにより避難者、誘導者、救助者のエージェントが定義される。これに対してコンポーネント指向型の MAS では避難者のエージェントは避難行動のコンポーネントを保持すると考える。避難者は避難行動のコンポーネントを持ち、誘導者は避難行動と誘導行動を持ち、救助者は避難行動と救助行動を持つとする(図 1 右)。

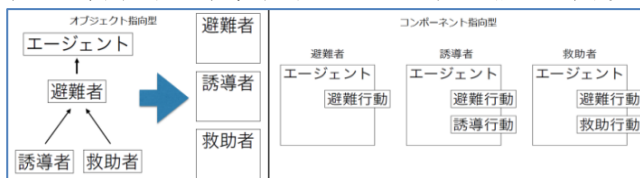


図1 オブジェクト指向型とコンポーネント指向型

Arisco: Proposal of component-oriented multi-agent simulator
[†]Ryunosuke Tada · Sojo University Faculty of Computer and Information Science
[†]Shinobu Izumi · Sojo University Faculty of Computer and Information Science

ここで、新たにロボットというエージェントを基本モデルに追加することを考える場合、従来のオブジェクト指向型では、ロボットは避難の必要がないため、避難者を継承するのは適切ではなく、ロボットの独自の行動を取るため、図 2 に示すようにエージェントクラスから派生する形で新たに3つのクラスを作成しなければならない。

コンポーネント指向型では行動のアルゴリズムを別コンポーネントとして扱うため、新しくロボットのエージェントを追加した時でも図3のように1つのクラスを新たに作れば他のクラスで再利用できる。結果として作成するクラスやコード量が削減できる。

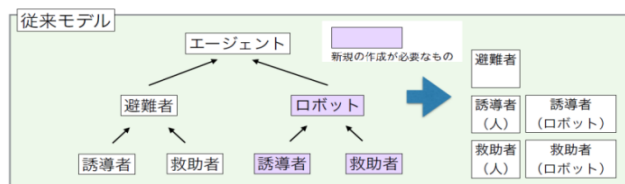


図2 オブジェクト指向型のモデル

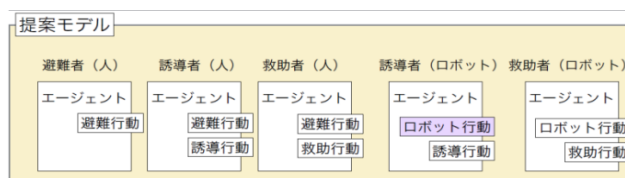


図3 コンポーネント指向型のモデル

3 定量的評価の方法

基本モデルが存在する状況を仮定し、それを再利用して発展させた拡張モデルへの変更を行う際の追加したコードの量、再利用できるコードの量、コピー・ペーストしたコード量を比較する。

ここで再利用できるとは継承や移譲による利用が可能なプログラムのことであり、コピー・ペーストしたコードは含めないものとする。継承や移譲により再利用するプログラムは変更を行う場合、そのプログラムを書き換えれば済むが、コピー・ペーストしたものは、それぞれのプログラムを修正する必要があるため、後の修正や変更において手間がかかるためである。

4 実験に用いるモデル

・基本モデル

人はランダムに動きまわり、発症者が視界 n 以内にいる場合に逃げる。発症者はランダムに動きまわり、距離 m 以内の人を感染者にする。感染者は潜伏期間 p ステップ後に発症者になる。武装者は距離 r 以内の発症者の動きを s ステップ止める、止めたあとは h ステップの間は止められない。救助者は感染者が距離 e 以内にいる潜伏期間の人を感染していない状態に戻す。

・拡張モデル

基本モデルにロボットという感染者にならないが人と同じ動作をするエージェントを追加する。ロボットを継承してロボットの武装者と救助者を作成する。

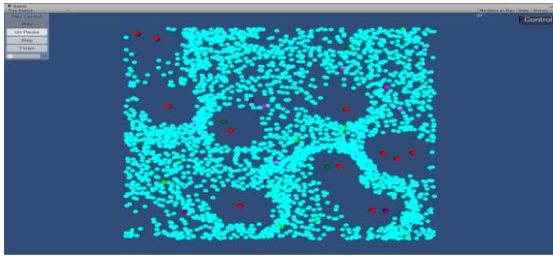


図 4 Arisco シミュレーション実行画面

5 実装の内容と結果

パンデミックモデルのシミュレーションをオブジェクト指向型として Artisoc、Repast を、コンポーネント指向型として Arisco を用いて実装した。拡張モデルに変更する際に新たに作成したファイルを数えたものをそれぞれの表の一番下に記述する。

Artisoc では継承などによる差分プログラムが行えないため、多くの部分をコピー・ペーストしてプログラムを記述した。

表 1 Artisoc でのコード量

Artisoc	行数	文字数
ベースモデル	14	4815
拡張モデル	62	1352
拡張モデルでコピー・ペーストしたコード量	39	955
再利用したコード量	なし	
新たに作成したファイル	2	

Repast ではロボットという基底クラスを作成し、それを継承した上で救助者や武装者のプログラムをコピー・ペーストして実装した。

表 2 Repast でのコード量

Repast	行数	文字数
ベースモデル	340	10012
拡張モデル	145	4376
拡張モデルでコピー・ペーストしたコード量	123	3472
再利用したコード量	なし	
新たに作成したファイル	3	

Arisco では大部分がコンポーネントとして再利用することができ、拡張モデルではロボットの基底クラスの作成のみであった。

表 3 Arisco でのコード量

Arisco	行数	文字数
ベースモデル	165	3756
拡張モデル	9	147
拡張モデルでコピー・ペーストしたコード量	なし	
再利用したコード量	60	1386
新たに作成したファイル	1	

6 考察

国内で広く使われている MAS である Artisoc、国外でメジャーな Repast、提案している Arisco でシミュレーションを作成し、基本モデルから拡張モデルへの変更にとれほどのコード量が必要になるのかを比較した。結果として、今回のモデルでは本研究室の Arisco での作成が最もコード量が少なかった。

文字数を比較するため基本モデルの何%の文字数で拡張モデルへ変更できるか比較した。Artisoc では基本モデルの 28.08%の記述で済み Repast では 43.71%と他に比べて多い、Arisco では 3.92%であった。

Arisco は Artisoc を参考にエージェントの取得など便利な関数を取り入れている。Repast はそういったものがないため、このようにコード量が多くなってしまったと考えられる。

7 まとめ

コンポーネント指向型の MAS を用いることでプログラムが再利用できコード量が少なく済むという利点を示した。コンポーネント指向型の利点はこれだけではなく基本モデルを拡張する時にコンポーネント単位で使い回せる、1 つのコンポーネントを編集するだけで全体のエージェントの動作を変更できるという利点もある。しかし、コンポーネント指向型の MAS はコンポーネントを取得する処理の負荷が高いため、多用すると処理が重く動作速度が低下してしまう。今後、コンポーネント指向型の MAS をより実用的にするためにはコンポーネントの取得処理に工夫が必要である。

参考文献

- [1] 福原 康二 コンポーネント指向マルチエージェントシミュレータ 2011 年, 崇城大学修士論文
- [2] 山陰 進、人工社会構築指南—artisoc によるマルチエージェント・シミュレーション入門、2007 年