群ロボットシミュレーションのフレームワークの提案――スケジューラーについて――

群ロボットシミュレーションのフレームワークの提案
-スケジューラーについて-
Proposed Framework for Swarm Robot Simulation: About the scheduler

東北学院大学　大学院人間情報学研究科人間情報学専攻
博士課程前期課程１年　佐々木　夏輝

1.　研究背景

近年、コンピュータの性能の向上により、コンピュータシミュレーションの精度も向上し、その有用性が世の中に認められ、シミュレーションの結果が幅広く使われている。その一例として気象シミュレーションを上げることができるが、気象庁の天気予報の精度検証結果[1]によると，翌日の降水の有無の的中率の年平均は1985年から2021年の36年間で約83%から約88%へと増加している（図１）。また、天気の予測だけでなく、交通渋滞の予測やコロナウィルスの感染者数の増減の予測など様々なシミュレーションが私たちの生活の至る所で活用されている。シミュレーションには、物理的モデルを利用して現実世界の現象を表現する物理シミュレーションや登場人物同士もしくは登場人物と空間の相互作用から現実世界の現象を表現するマルチエージェントシミュレーションなどがある。

図１出典：気象庁「天気予報の精度検証結果」

図２　エージェントのイメージ図

マルチエージェントシミュレーションは、エージェントと呼ばれる自律的に行動する主体をモデルとする。エージェントの特徴は、環境に関する情報や他のエージェントとの位置関係などから自身が置かれている状況を認知し、その状況に基づいた判断から自身の行動を変化させる点である。個々のエージェントの行動は非常に小さいものであるが、行動が互いに影響し合ったり積み重なったりすることで、現象全体で見ると大きな変化につながっていく。このように、小さな変化の相互作用や蓄積から複雑な現象を理解し、予測しようとするシミュレーションがマルチエージェントシミュレーションである（図2）。

鳥海・山本(2014)によるとマルチエージェントシミュレーションには、意味世界志向と物理的世界志向という二つの志向性が存在すると説明されている。意味世界志向のシミュレータでは、株価のシミュレーションのようにエージェント（投資家）の行動ルールとその行動ルールの結果（売/買行為）が相互作用する対象（株価）との関係性を必要とする。この時、何時何分に株価が上昇もしくは、下

落したかという時間は必要ではなく、どのような行動ルールを持ったエージェントが株価と相互作用し、どのような株価の変化につながったのかが重要となる。一方、物理的世界志向のシミュレータでは、エージェントの物理的空間における位置や、エージェントの周りの空間情報が重要となる。このように、物理的世界志向のシミュレータでは、現実世界の時間の概念をできる限り反映させる必要がある。

近年の物理的世界志向のマルチエージェントシミュレーションの研究事例として、Abe ら(2022)の研究がある。この研究では、筑波大学周辺の地理データを読み込み、その地図データにお店や施設を用意し、施設やお店における人と人との接触が、COVID-19 の感染拡大へどのような影響を与えるのかを調べるために、専用のシミュレータ開発が行われている。

2. 研究目的

近年のマルチエージェントシミュレーションの研究では、シミュレーションする対象に特化したシステムを構築する場合が多いものの、利用者の目的に応じたマルチエージェントシミュレータの開発を支援するフレームワークの開発に関する研究も進められている。

現在は、対象を Java 言語のメソッドの内容記述ができる程度のプログラミング能力を持つ学部学生とした群ロボットシミュレーションのフレームワークの開発を進めている。特に本稿では、フレームワーク内で用いられるエージェントの行動タイミングを管理するスケジューラーに関する提案を行う。

3. 従来のマルチエージェントシミュレーションについて

フレームワークの開発の成果として MASON[4]や Repast[5]などを挙げることができる。これらの特徴は、表現する現象に適したエージェントとエージェントが移動する空間を利用者が用意することでマルチエージェントシミュレーションを行える設計がなされている。そのため、エージェントの移動に必要な演算処理や、現象の変化を利用者に伝えるための描画処理 (GUI)、そして描画処理と演算処理のズレが生じないように制御する制御処理などが予め用意されている。MASON の時間の取り扱い方は、システム内で取り扱う最小の時間 (step) を設定し、その step 間隔毎に時間を進め、その時間が更新される毎に、次の行動をどうするのかということをシステムがエージェントへ問合せを行うことで、エージェントの次の行動決定を促す仕組みとなっている。この時間の取り扱い方でシミュレーションを行うと、エージェントの行動決定のタイミングは step と同期する。しかし、行動決定のタイミングは現実世界では、個人ごとにばらつきが生じていると考えることができる。Repast では、step 時間に相当する tick という時間を採用して、エージェント毎に異なる tick を持たせることができる。そして、各エージェントの tick に合わせて、次の行動決定を促す仕組みが採用されている。本研究の時間の概念は、Repast のようにエージェント毎に tick 時間を持たせ、その時間に基づいてエージェントの行動決定を促す仕組みを採用する。

4. フレームワークで取り扱う時間

Repast では、エージェントの状況判断が 1tick 毎に繰り返され、この間に意志決定と行動が同時に実行される。これは、エージェントの意思決定が瞬時に行われていると考え、モデル化されていると捉えることができる。しかし、現実世界で同様の過程について考えると意思決定には 1tick を超える時間を要することは明白であり、その決定の後に行動を変化させている。このような事象を従来の方法で表現するためには、思考に要した時間を（何もしない時間として）プログラムに

組み込まないといけない。

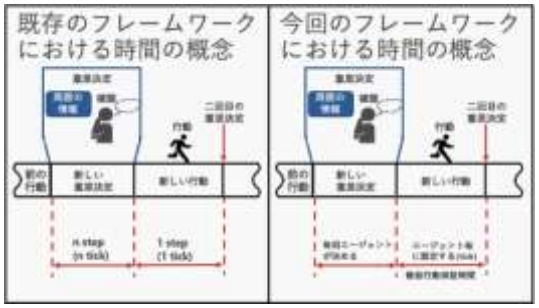

図 3　時間の概念の違い

今回のフレームワークでは、意思決定に要する時間を意思決定時間、意思決定の内容に基づいて行動を継続する時間を最低行動保証時間とし、二つを分けて取り扱う（図 3）。

意思決定時間は、エージェントが周囲の状況を認知し、次の行動内容を決定する時間である。意思決定では、他のエージェントとの位置関係や周囲のエージェントがどの方向にどれくらいの速さで進んでいるのかという情報を得ることができる。この情報を活用し、自身の次の行動内容（移動の方向・速さ・相互作用する対象など）を決定する。本提案では、意思決定毎にエージェントが自由に意思決定時間の長さを決めることができ、その時間分だけエージェントの次の行動を遅延させる。

最低行動保証時間は、既存のフレームワークの step・tick に相当する時間である。意思決定時間で決定した行動内容に基づいた行動を最低限継続する時間であり、エージェント毎に固有の時間を持っている。この時間経過後、システム側からエージェントへ次の行動の問合せが再び行われる。エージェントは問い合わせに対して、新たな行動もしくは、現在の行動の継続のいずれか一つを選択する。今回のフレームワークでは、意思決定と行動の二つのイベントを異なる時間に分離して取り扱うため、エージェント毎に異なるイベントの開始時刻を管理するスケジューラーが必要となる。

5.　スケジューラーについて

スケジューラーの主な役割は、予約されているイベントの中で最も開始時刻が早いイベントを見つけ出すことであり、システム側からエージェントに対する新たな行動の問合せを適切に行うことを支援することである。

スケジューラーの基本的な流れを図 4 に示す。まず初めに、シミュレーションの開始と同時にエージェント（A_1, A_2）が誕生するが、誕生直後のエージェントは次の動作が決定していないため、最初の意思決定を促すためのイベントとその開始時刻（t_1=0, t_2=0）がスケジューラーに予約される。このイベントに従って、システム側から A_1 と A_2 に対して意思決定を問合せ、意思決定時間を考慮した新たなイベントの開始時刻（t_1=0.3, t_2=0.7）をスケジューラーへと設定する（図 4（i））。次にスケジューラーは、予約済みのイベントの中で開始時刻が最も早いものを見つけ出し、システム側に伝える。今回の場合、A_1 のイベントの開始時刻が t_1=0.3 で最も早いため、その時刻までシミュレーション内の時間を進める。そして、システム側から次の行動が A_1 に促され、A_1 は移動を開始する（図 4（ii））。A_1 の次のスケジュールは、A_1 の tick 分進んで t_1=0.8 となる。次のイベントの所有者は A_2 であり、t_2 の時刻までシミュレーション内の時間を進めると A_2 が移動を開始し、先に行動し始めた A_1 もこの時間の経過分だけ移動を継続する（図 4（iii））。A_1 の tick 分だけ時間が経過した t_1=0.8 が次のスケジューラーのイベントのため、その時刻まで、シミュレーション内の時間を進め、システム側から A_1 に次の行動の問い合わせが行われ、A_1 の新たなイベントがスケジューラーへ追加される。この流れを続けていくことで、異なるタイミングで行動するエージェント達の管理を行う。

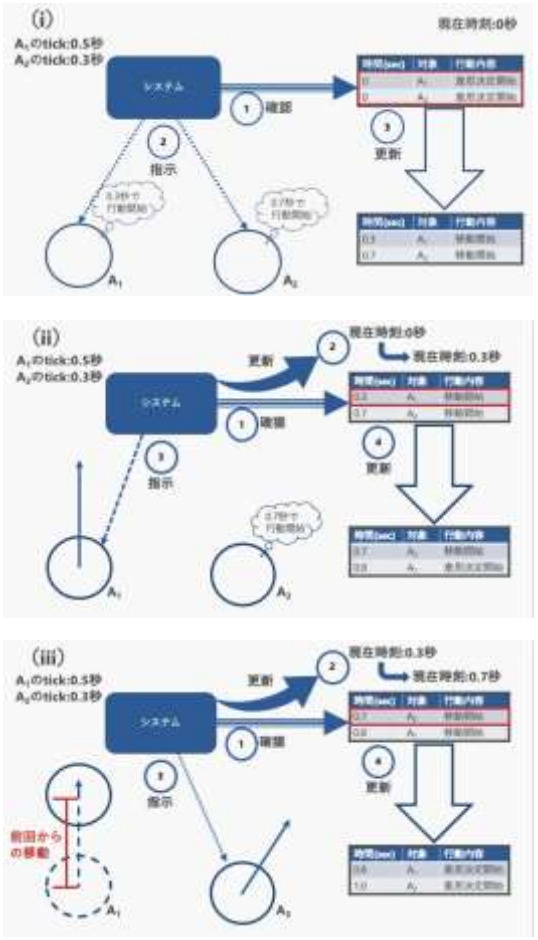

図 4　(i) エージェントの誕生
(ii) A_1 の行動開始
(iii) A_2 の行動開始と A_1 の移動

6. 非同期のイベント

スケジューラーに予約されるスケジュールは、基本的にエージェント毎に設定した tick と意思決定時間によって決まるがそのスケジュールを変更しなければならない現象として非同期のイベントがある。

6.1. 衝突

エージェント同士もしくは、エージェントと障害物の衝突は、各エージェントの最低行動保証時間内で、エージェントの移動が遮られる場面で生じる。エージェントは、衝突すると最低行動保証時間が過ぎるまでは、衝突した地点で停止し、最低行動保証時間経過後に新たな意思決定を開始するという仕様にした（図 5)。まず初めに、スケジューラーから最も開始時刻が早いイベント($t=t_1$)を探し出し、現在時刻からそのイベント開始時刻までに、各エージェント(A_1,A_2,A_3)が意思決定に基づいて移動した場合に衝突する点が存在するかを確認する。存在しない場合、通常のスケジューラーの流れ通りに、t_1 まで時間を進め、エージェント達もその分移動させる。衝突する点が存在する場合、システムは、エージェントが衝突する点まで移動する場合の所要時間を求め、その時間が早いものから順に、エージェントを衝突する点まで移動させる。このようにすると A_1 と A_2 の衝突が発生し、A_1 と A_3 の衝突が起こらなくなる。その後、t_1 まで時間を進め、その時間分 A_3 を移動させ、A_1 に意思決定を促す。

6.2. 相互作用

システムに実装される Agent クラスには、エージェント同士が相互作用を行うために、フレームワーク利用者が実装すべき幾つかのメソッドを保持している。表 1 に、これらのメソッドの引数と返り値が持つ意味を示す。

非同期で活動するエージェントが、他のエージェントと相互作用して一つの作業に取り組む場合、このエージェント達の行動開始時刻は同期される必要がある。相互作用を開始する側のエージェントは、システムから呼び出された action メソッドの返り値(オブジェクト)に、「自身の意思決定の結果が相互作用であること」、「相互作用する相手のエージェント番号」、「エージェントに渡す情報」、「タイムアウト(相互作用される側からの返答を待つ時間)」をオブジェクトにセットしてシステムに返す。

システムは、このオブジェクトから「エージェントに渡す情報」を回収し、相互作用される側のエージェントのisInterruptableメソッドにこの情報と相互作用する側のエージェントの番号をメッセージとして送る（「エージェントに渡す情報」と相互作用する側のエージェントの番号を引数として

isInterruptableメソッドを呼び出す）ことが可能か否かを「タイムアウト」の期間と相互作用される側の次の行動開始時刻の関係性を基に判断する。相互作用される側のエージェントの次の行動開始時刻が「タイムアウト」の期間内の場合、システムは相互作用される側のエージェントにメッセージを送ることができる。一方で、相互作用される側のエージェントの次の行動開始時刻が「タイムアウト」の期間外の場合、相互作用は失敗となり、システムは相互作用される側のエージェントのisInterruptableメソッドを呼び出さず、相互作用される側のエージェントの行動開始時刻を「タイムアウト」の期間が終了する時刻としてスケジューラーに登録する（図 6）。

相互作用されるエージェントが相互作用を受け付ける場合、isInterruptableメソッドは「相互作用を受け付けたこと（true）」をオブジェクトにセットしてシステムに返す。しかし、相互作用をされるエージェントが相互作用を受け付けない場合、isInterruptableメソッドは「相互作用を受け付けないこと（false）」をオブジェクトにセットしてシステムに返す。このとき、相互作用は失敗となり、相互作用される側のエージェントは、システムから次の行動を即座に促され、相互作用する側のエージェントは、システムによって次の行動開始時刻を「タイムアウト」の期間が終了する時刻としてスケジューラーに登録される（図7）。

システムは、相互作用される側のエージェントの isInterruptable メソッドから「相互作用を受け付けたこと(true)」を返り値として受け取ると相互作用する側のエージェントの interrupting メソッドを呼び出す。そして、interrupting メソッドの内部で相互作用の処理がされ、「相互作用に要した時間」を返り値のオブジェクトにセットしてシステムに返す。この際、システムは、「相互作用に要した時間」を相互作用する側のエージェントの次の行動開始時刻としてスケジューラーに登録する。そして、相互作用される側のエージェントの interrupted メソッドに相互作用をする側のエージェントの番号と「エージェントに渡す情報」をメッセージとして送る。interrupted メソッドの内部でも同様に相互作用の内容が処理され、「相互作用を終了してから次の行動を開始するまでに必要となる時間」を返り値のオブジェクトにセットしてシステムに返す。この時間が 0 秒であれば、相互作用された側のエージェントの次の行動開始時刻は、相互作用する側の行動開始時刻と同期する。一方で、0 よりも大きい場合には相互作用する側のエージェントの行動開始時刻にこの時間を加えた時刻を相互作用された側のエージェントの次の行動開始時刻としてスケジューラーに登録する（図 8）。

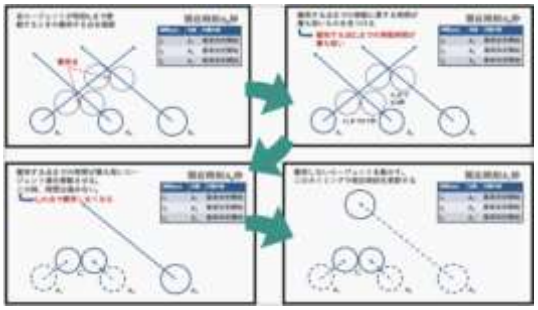

図 5　衝突発生時のエージェントの移動

表 1　相互作用に必要なメソッドとその引数・戻り値

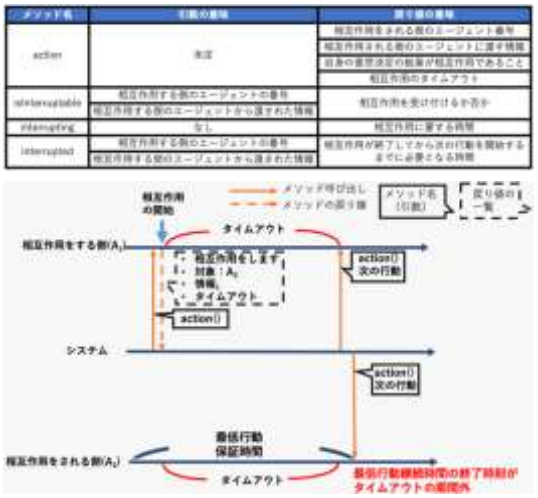

図 6　相互作用に反応できない場合

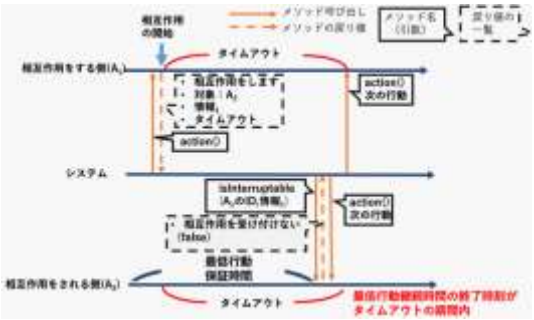

図 7　相互作用を拒否された場合

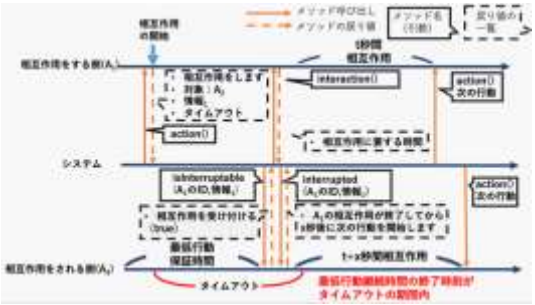

図 8　相互作用に成功した場合

7.　まとめ

本稿では、従来のフレームワークで用いられている時間の取り扱いを調査し、群ロボットのシミュレーションで用いられるマルチエージェントシミュレーションのフレームワークに実装すべきエージェントの意思決定時間を考慮したスケジューラーの提案を行なった。そして、提案するスケジューラーが、エージェントの移動を正しく動作させること、エージェント同士の衝突や相互作用の処理が適切に行われることを確認した。

参考資料

1. 気象庁「天気予報の精度検証結果」（最終閲覧日 2023 年 1 月 26 日）
https://www.data.jma.go.jp/fcd/yoho/kensho/yohohyoka_top.html

参考文献

2. 鳥海不二夫, 山本仁志.「マルチエージェントシミュレーション:1. マルチエージェントシミュレーションの基本設計」『情報処理』, 55(6), 530-538(2014).
3. Abe, Mitsuteru, Fabio Tanaka, Jair Pereira Junior, Anna Bogdanova, Tetsuya Sakurai, and Claus Aranha. Using Agent-Based Simulator to Assess Interventions Against COVID-19 in a Small Community Generated from Map Data. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems, 1-8 (2022).
4. Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. MASON: A Multi-Agent Simulation Environment. Simulation, 81(7), 517-527 (2005).
5. Michael J North, Nicholson T Collier, Jonathan Ozik, Eric R Tatara, Charles M Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with Repast Simphony. Complex adaptive systems modeling, 1(1), 1-26 (2013).