

震災時の避難行動に及ぼす情報活用の効果～避難行動のシミュレーション～

計画マネジメント・皆川研究室 渡邊 裕介

1. はじめに

(1) 研究背景

日本は世界でも有数の地震大国と言われている。日本周辺では世界の地震のエネルギーの10分の1が集中しているとも言われている。阪神・淡路大震災や新潟中越地震は記憶に新しい。そして地震の脅威は収まる事を知らず、今後、東海地震、東南海地震、首都直下型地震など様々な巨大地震の発生が予測されている。地震の発生を防ぐ事をまぬがれることはできない。そこで、現在の情報技術を用いて、多少なりとも震災後の劣悪な環境を改善、被災者をサポートすることは可能であると考えた。震災時の人々の動きを予測し避難シミュレーションを行うことで、人々が安全、に避難する事をサポートし被害を軽減できると考えられる。震災時の避難シミュレーションの研究は既に行われているが、現在では、情報技術の進歩と共に携帯情報端末から様々な情報をいつでも容易に手に入れることが可能である。そこで、その状況を考慮し被災者が効率的に避難出来るための情報を被災者に与えることも可能である。震災時に被災者に効率的に避難できる情報を時間経過と共に与え、その動きをシミュレーションする。

(2) 研究目的

研究背景で述べたように、現在の日本は様々な情報を容易に手に入れることが可能な状況にある。そこで、未だ行われていない、情報利用による震災時の避難シミュレーションを行い、現代の情報化社会のなかでの安全で円滑、効率的な避難行動をサポートし、少しでも震災の二次的被害を軽くする事が本研究の目的である。

2. 研究内容

(1) マルチエージェントシステム

現在、シミュレーションを行うにあたって最も有効で注目されているのはマルチエージェントシステムである。マルチエージェントシステムとは、まず多数の自律的に行動するエージェントから構成されるシステムであり、それぞれのエージェントは自分の目標を達成するよう動き、システム全体の振る舞いはエージェント同士が相互に作用することによって決定される。つまり、複数の人々の動きをシミュレートするには最適なシステムである。よって本研究ではマルチエージェントシステムを軸にシミュレーションを行うこととした。最終的にはこのマルチエージェントシステムを駆使して正確なシミュレーションを行い、様々な情報を人々に与えるという条件をプログラムに加え、より現実に近い形での避難行動を再現する。そして、時間経過と共に避難した人々の数をカウントしグラフ化、動画として出力する。しかし、マルチエージェントシステムはとても複雑なプログラミングであり、そのプログラミング言語を習得するのにかなりの時間を要するというのが現状である。そこで、StarLogoTNG というシミュレーションソフトを用いて基本的なプログラムの構造を理解し、後に複雑なプログラミングに移行していく方針をとった。

(2) StarLogoTNG

StarLogoTNG はマルチエージェントの基本的なシステムである。言語のブロックによるプログラミングが可能で、それによって作成したプログラムを容易にグラフィック化できることである。ブロックをパズルのように組み立てることによってプログラ

ムを作成していく。StarLogoTNG で基礎的なトレーニングを行い、基本的な知識をつけた。その後、より複雑なプログラミングを行うために複雑なシミュレーションが可能な artisoc2.5 というシミュレーターを導入した。

(3) artisoc2.5

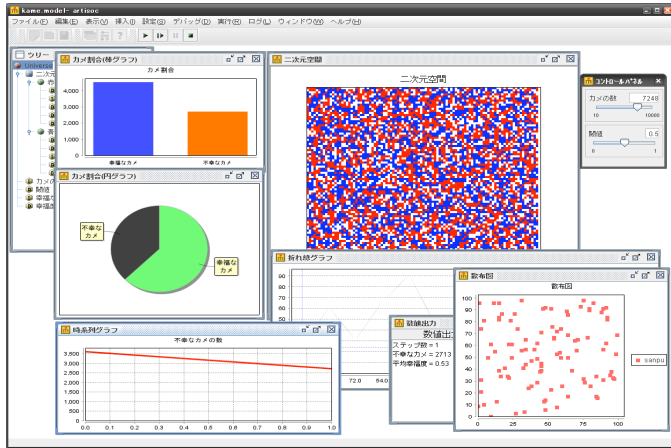


図-1 artisoc 使用画面

より複雑なプログラミングを行うために、次の段階へ移行した。artisoc2.5 は、アイデアをすぐにモデルへ反映できる、直感的な操作性を兼ね備えたマルチエージェント・シミュレータである。複雑系とマルチエージェントのアプローチによる研究活動を支援するため、artisoc を教育目的で使用する場合は、無償貸与するとのことであり、すぐに申請、導入に至った。導入後 artisoc のサンプルモデルの中に遊園地モデルというものを見つけ、目標としているシミュレーションモデルに非常に近いものであると考えた。

(4) 遊園地モデル

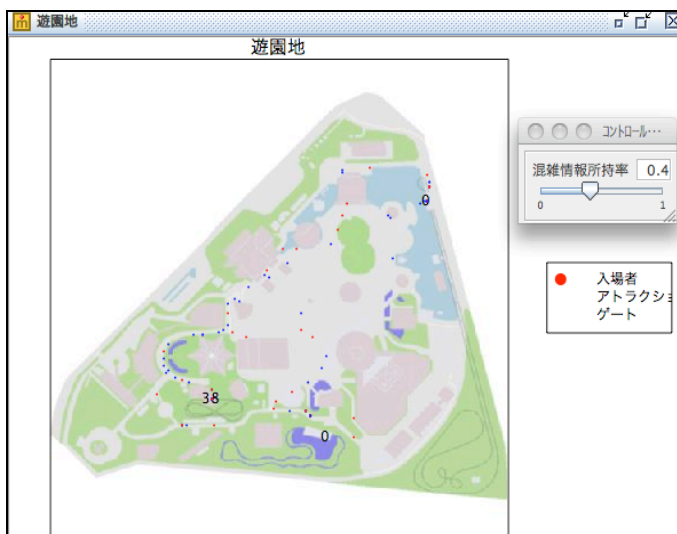


図-2 遊園地モデルのマップ

2次元マップ上の遊園地に三つのアトラクションが用意されている。エージェントはそれぞれアトラクションに向かうように設定されているが、エージェントに対し、アトラクションの混雑情報を設定した割合だけ与える事ができる。よって、各アトラクションがどのように混雑するのかをシミュレートできる。この遊園地を被災地と置き換え、アトラクションを避難所、避難所までのルートと見立てれば目指しているシミュレーションとかなり近いものであると考えた。入場者というエージェントは入場口で生成され、その後、情報をもっていないものは、ランダムにアトラクションに向かっている。一方、情報を持っているものは、一番すいているアトラクションへ向かうように設定されている。この際に複雑な条件分岐が必要となる。グラフ、変数も多く存在するために、複雑なプログラムとなっている。

(5) 初歩的なモデル

いきなり遊園地モデルのようなモデルを作成する事は困難であると考え、まずは artisoc の基本操作、基本知識を習得した上で初歩的なモデル作りから取り組む事とした。初歩的なモデルから段階的に条件、要素を増やす事により、正確で詳細なシミュレーションを行えると考えた。

(6) 花火モデル

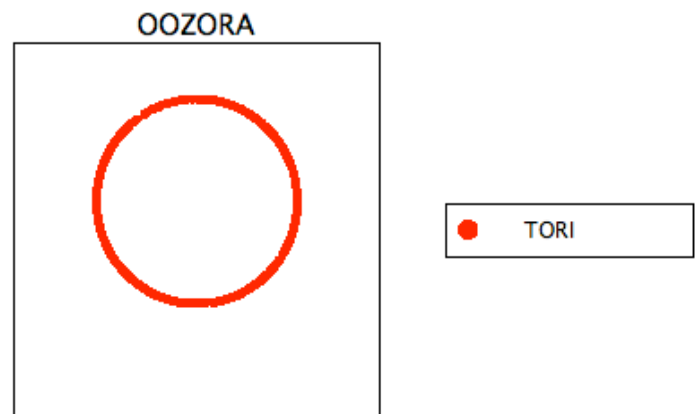


図-3 花火モデルのマップ

まずは、エージェントの基本的な動作方法を学習するために、空に打ち上がる花火を模した花火モデルを作成した。この花火モデルには単純ではあるがエージェントの基本的な動きをさせるための要素が入っている。まず、無数のエージェントがマップ上の中央下部から打ち上げられ、ある程度上昇した際

にランダムに 360° 方向に散らばる。ここで必要となる動きは、エージェントが垂直方向に前進、あるステップ数を超えたら、360° のランダムな方向に向きを変えろというものである。プログラムのテキストは図-4 に示す。

```

Agt_Init{
my.X=25
my.Y=0
my.Direction=90
}
Agt_Step{
if GetCountStep()==30 then
if rnd()<0.5 then
my.direction=Rnd()*360
end if
Else
forward(1)
End if
}

```

図-4 花火モデルエージェントルール

(7) 立ち話モデル

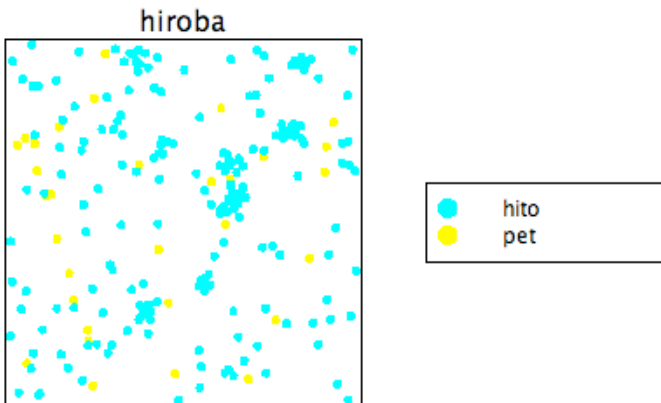


図-5 立ち話モデルのマップ

エージェントの基本的動作を学習した後に取り組んだのが立ち話モデルである。このモデルで最も重要になることは、周囲のエージェントから影響を受けながらエージェント自身が行動するという点にある。さらに、2種類のエージェントを混同させ、それぞれを区別しながら動くことができる。コントロールパネルを用いて変数を自由に操り、時系列のグラフを出力することも可能になった。このモデル内でのエージェントの動きは、人とペットという2種類のエージェントがそれぞれランダムに配置される。そして、人エージェントには周囲のエージェントを調べさせ、そのエージェントの数が設定数以上の場合、立ち止まり集団をつくる。この際にペットエージェ

との数はカウントせずに人エージェントだけに注目する。プログラムのテキストは図-6 に示す。図-6 中にある MakeOneAgtsetAroundOwn という関数が周りのエージェントを調べるものである。マルチエージェントのシミュレーションにおいて、この関数は非常に重要なものとなる。

```

Agt_Init{
my.Direction=rnd()*360
}
Agt_Step{
MakeOneAgtsetAroundOwn(My.mawari,universe.shiya,
If CountAgtset(My.mawari)>=universe.nakama Then
Universe.tachidomari=universe.tachidomari+1
Else
universe.arukimawari=universe.arukimawari+1
Forward(1)
Turn(rnd()*15)
End if
}

```

図-6 立ち話モデルエージェントルール

(8) 避難シミュレーションの初期モデル

上記に挙げた2つのモデルは、ほんの一部である。上記以外にも様々なモデルを作成し基礎的な知識を深めた。そこで、実践的なモデル作りを開始した。まだまだ知識が不足しているために、遊園地モデルのような複雑なものは作成出来ないが、避難シミュレーションという部分に重点を置き、単純ではあるがモデルを作成した。このモデルでは、まずマップ画像の交差点上にポイントエージェントというものを初期配置している。ポイントエージェントを交差点上に配置することにより、道路をネットワークの様に捉えることができる。それぞれの被災者エージェントはこのポイントエージェントに向かうように設定し、ポイントに到着したエージェントは次のポイントに向かうように動く。この際にポイントエージェントに到着したと判断する基準として被災者エージェント自らとポイントエージェントの距離を計測しながら移動している。ポイントエージェントまでの距離が0に近づいた時に、被災者エージェントはポイントに到着したと判断する。ポイントエージェントに到着した際に次のポイントエージェントを選ぶ場合、ランダムに同じ確率で選ぶように条件を

設定、ランダムにゴールに近づく仕組みである。しかし、ランダムにゴールに近づくだけでは目標としているシミュレーションとは遠い、そこで被災者エージェントに、よりスムーズにゴールに近づける情報を与えることとした。情報をもっている被災者エージェントと持っていない被災者エージェントを色分けという方法で区別させる。赤色の被災者エージェントは情報を持っていない、一方、緑色の被災者エージェントは情報を持っている。情報をもっている被災者エージェントは、ほぼ無駄な動きをせずゴールに向かう。さらに、被災者の人数、移動速度、情報をもっている被災者エージェントの数は変数を操作することで随時変更が可能となった。プログラムの一部テキストは図-9、図-10に示す。



図-7 避難シミュレーションマップ

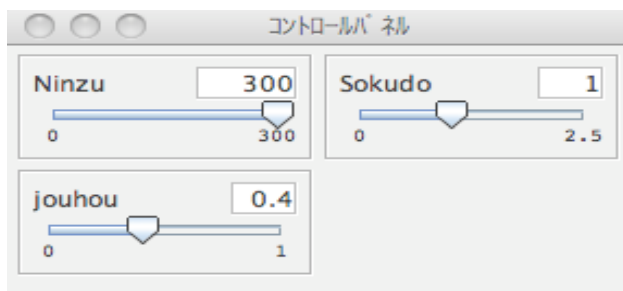


図-8 コントロールパネル

```
for i = 0 to universe.ninzu-1
  one = createagt(universe.hisaichi.hisaisha)
  if Rnd()<universe.jouhou then
    one.condition = color_green
  else
    one.condition = color_red
  end if
next i
```

図-9 情報のルール

図-9 中のエージェントの初期設定で universe.jouhou の値を操作することにより 2 種類のエージェントの区別を行っている。すなわちエージェントの持っている情報の有無を区別している。図-10 中の B 地点でのルールでは 2 種類のエージェントを区別、情報を持っていない場合はランダムに次の目的地を選択するが、情報をもっている場合は指定された方向に動く設定とした。

```
//B地点到達
if my.condition == color_green then
  if distanceB<1 then
    my.Direction=getdirection(my.x,my.y,Cx)
  end if
else
  if distanceB<1 then
    if rnd()<0.5 then
      my.Direction=getdirection(my.x,my.y,Ax)
    else
      my.Direction=getdirection(my.x,my.y,Cx)
    end if
  end if
end if
```

図-10 ポイント B 地点でのエージェントルール

3. まとめと今後の課題

マルチエージェントのモデルの基本的な学習を行った。簡単ではあるが初期の避難シミュレーションモデルを構築することができた。しかし、瀧川の設定しているシナリオ構成、複雑な条件は十分に反映することができていないのが現状である。今後の課題、改良しなければならない点は、引き続き遊園地モデルの理解、シミュレーションの範囲の拡大、エージェントの種類の増大、エージェントの動きの複雑化、エージェントを均衡配分、エージェント同士で情報の共有を行えるようにすることである。今回構築した避難シミュレーションの初期モデルを元に地道に条件、変数を増やしていくことで、瀧川のシナリオ、設定条件を十分に反映させたシミュレーションモデルを構築することが可能であると考えられる。

【参考文献】

- 1)MAS コミュニティー
http://mas.kke.co.jp/ 2009.12.09.
- 2)山影進:人口社会構築指南
pp.1-446 2010.1.18