人　工　知　能 99－2
（1995．3．6）

エージェントの挙動の表現法について
― 交渉による集結の一例 ―

五坪賢次　　笹倉万里子　　荒木啓二郎
奈良先端科学技術大学院大学

本論文では、エージェントによるアニメーション記述のためのエージェントの挙動モデルについて述べる。われわれはアニメーション作成は抽象的な記述によるものが望ましいと考えている。それを実現するためには、エージェントが抽象的に示された目的から具体的なゴールを決定し、さらにそこから各エージェントの局所ゴールを決めてそれを達成することにより指定された目的を達成するという機構が望ましい。ここでは、このような機構を実現するために交渉を行なって簡単な集結問題を解決するモデルを提案する。

Model for movements of agents to solve a meeting problem with negotiations

Kenji Gotsubo, Mariko Sasakura and Keijiro Araki
Nara Institute of Science and Technology

In this article, we describe a model for movements of agents which compose an animation. Our final goal is to specify an animation by abstract descriptions. To achieve this, agents must have an ability to decide their each concrete goals from an abstract goal for a whole animation and to achieve their goals individually. As the first step of this approach, we propose a simple model which realizes agents which solve a meeting problem with negotiations.

1 はじめに

われわれは抽象的に記述したシナリオから簡単にアニメーションを作成するためのモデルについて研究を行なっている。この研究における大きな課題は、いかに抽象的な記述から具体的なアニメーションに変換するか、である。記述はできるだけ抽象的なレベルで行いたい。

アニメーションを簡単に作成するためのモデルとしてオブジェクト指向モデルが多く使われている。このモデルでは、アニメーションに出てくる登場人物の性質・可能な行動を個々に記述し、それらの登場人物に対してこれから作成しようとするアニメーションでそれぞれの登場人物がどのように動くかを指定する。これは、実世界におけるシナリオ (脚本) のメタファであるといえる。実世界では個々の役者は与えられたシナリオにしたがってそれぞれの役割を演ずる。シナリオは普通抽象的に書かれていて、それを現実にどのように演ずるかは役者に任されている。アニメーションにおけるオブジェクト指向モデルはこのシナリオメタファを実現するものとして使われている。

しかし、オブジェクト指向モデルでは結局抽象的に書かれたシナリオの解釈法を登場人物の性質・可能な行動の記述の中に埋め込んでしまわなければならない。その際、オブジェクト指向モデルではオブジェクト間の「協調」という考え方がないので、複数のオブジェクトが協調して何かを行なうようなアニメーションが作りたい時にはその協調の仕方を明示的にシナリオに書かなければならない。これは抽象的に書きたいというシナリオの目的と矛盾している。

シナリオに具体的な協調の仕方を書かなくてすむようにするためには、各登場人物の性質・可能な行動としてある程度の基本的な協調動作ができるような記述を行なわなければならない。このように協調動作可能なオブジェクトからなるモデルをわれわれはエージェントモデルと呼ぶことにする。エージェントモデルの構築のためには一般的基本的な協調動作とはどんなものか、ということがわかっていなければならない。そのためにまず簡単なモデルで協調動作について研究を行う必要がある。

一般にエージェントの協調行動には大きく分けて次の 2 種類が考えられると言われている [1] [2]。

1. 協調問題解決
2. 交渉と均衡化

協調問題解決とは、複数のエージェントが協力して一つの問題を解決しようとするものである。ここで課題となるのは、問題をどのように部分問題に分割してエージェントに分担させるか、である。

一方、交渉と均衡化とは、複数のエージェントがおのおの解決すべき問題を個別に持っているものである。おのおののエージェントは独自に問題を解決しようとして様々な資源を使う。ここで問題となるのは複数のエージェントが同一の資源を使おうとした場合である。この場合、一般的にはエージェント同士で交渉を行って解決する。

エージェントのアニメーションへの応用を考えてみよう。例えば、いくつかのエージェントで A 地点付近に積んであるブロックを B 地点へ移動する、というようなアニメーションは協調問題解決の例であるといえよう。また、複数のエージェントがそれぞれの目的地へぶつからないように移動するアニメーションは交渉と均衡化の例と考えられる。

エージェントによる協調問題解決のアニメーションは、一種の分散状態空間探索問題であると考えられる。すなわち、次の課題を解決する必要がある。

1. 大局ゴールを設定する。
2. 各エージェントが局所ゴールを設定する。
3. 大局ゴールが達成されるように、各エージェントが達成した局所ゴールの結果を合成する。

実際には、この3つの課題が独立には解決できないことが問題である[3]。例えば、各局所ゴールを達成した結果大局ゴールが達成されるようにどのように合成するかは、そもそも大局ゴールからどのように局所ゴールを設定するかに深く関わっている。あるいは、局所ゴールを合成した結果が大局ゴールになるようにするために、そもそもの具体的な大局ゴールの設定の仕方を変えるというアプローチもある。

そこで、本論文では、簡単なエージェントの集結問題を例として分散状態空間探索問題を考える。集結問題とは、ばらばらに配置したエージェントを一ヶ所に集める問題であり、エージェントの協調性を利用して解ける問題として[4]で扱われている問題である。千村他は[4]において複数のエージェントの集結問題について論じた。しかし、そこではエージェント同士が情報を交換したり交渉することはない。本論文では、エージェント同士が情報を交換して随時大局ゴールを変更しながら集結するモデルについて述べる。

2 集結問題

この節では、本論文で扱う集結問題を設定する。そのあと、集結問題を解決するための問題点について述べ、本論文で扱う範囲を明らかにする。

2.1 問題設定

まず、2次元格子状に区切られた有限な大きさを持つ平面を考える。その格子マスのいくつかにエージェントが存在する。エージェントは4方向に1マスずつ移動することができる。平面上にはランダムに障害物が存在し、エージェントは障害物のある格子マス上には移動することができない。また、エージェントは自分がいる格子マスの4方向の隣に障害物があるかどうかを知ることができるが、それ以外のどこに障害物があるかを知ることはできない。

この時、2つのエージェントに対して「合流せよ」という目的が与えられる。この目的は外部から、例えばユーザから与えられるものとする。「合流する」とは、2つのエージェントが同じ格子マスの中に入る状態をいう。「合流せよ」とは、指定された2つのエージェントがいつかどこかの格子マスの中に入った状態に達することを要求することである。合流するように指定された2つのエージェントは、互いに合流相手のエージェントがどれであるかを知り、そのエージェントと通信ができるものとする。

以上が本論文で扱う集結問題である。

2.2 問題点

指定された2つのエージェントが合流するためには次の問題点がある。

1. 具体的にどのマスで合流するかを決めなければならない。この具体的なマスのことを移動目標または目標と呼ぶことにする。
2. 各エージェントが移動目標に向かって移動する経路を決定しなければならない。

次節では、これらの問題をエージェント同士で交渉することによって解決していくアルゴリズムを述べる。

3 ミーティングアルゴリズム

本節では、合流する目的を持った2つのエージェントが用いるミーティングアルゴリズムについて述べる。その前に準備をする。まず、このアルゴリズムを用いる舞台となる世界やこの中に存在する物体を定義する。

定義 3.1 世界

世界とは、$m \times n$ の 2 次元格子をなしており、左下を $(1,1)$、右上を (m,n) とした座標を持つ。また、上を北とした東西南北で向きを表す。北向きを表す単位ベクトルは $\vec{N} = (0,1)$、東向きを表す単位ベクトルは $\vec{E} = (1,0)$ とし、南は $\vec{S} = -\vec{N}$、西は $\vec{W} = -\vec{E}$ とする。

□

この世界には、次の 2 種が存在する。それぞれの数はいくつでも良い。

定義 3.2 エージェント

エージェントとは、合流する当事者であり一つのマスに収まる大きさをなしており、次の特徴を持つ。また、エージェントの名前を $A, B, C, \ldots$ で表し、その位置を $\vec{A}, \vec{B}, \vec{C}, \ldots$ で表す。

- エージェント自身の意思により、東西南北のいずれかの方向に 1 マスづつ動くことができる。
- 自分のいる位置 (世界における絶対座標) を知ることができる。
- 2 つ以上のエージェントが同じマスに存在することができる。
- 後に定義する障害物と同じマスに存在することはできない。

□

定義 3.3 障害物

障害物とは、エージェント以外の物体であり、いくつかのマスで構成されても良い。また、障害物の名前を $O, P, Q, \ldots$ で表す。

□

例えば、図 1 は、7×7 の世界であり、その中に 3 つのエージェントと 2 つの障害物が存在している。各々のエージェントの位置は $\vec{A} = (2,6)$、$\vec{B} = (6,4)$、$\vec{C} = (4,2)$ であり、エージェント C は障害物 P があるために北の方向 $(\vec{N})$ には移動できない。

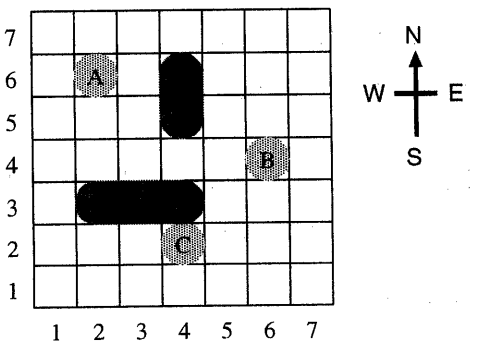

図 1: 世界の例

次に、エージェント同士が交渉などのために用いるメッセージを定義する。

定義 3.4 メッセージ

send-position(POSITION)	:	自分の位置 (POSITION) を相手に送信する。
send-direction(DIRECTION-SET)	:	目標を DIRECTION-SET のうちのいずれかの方向に変更したいことを送信する。
accept(DIRECTION)	:	相手から提示された変更方向のうち、DIRECTION を受け入れたことを送信する。
reject()	:	相手から提示された変更を受け入れられなかったことを送信する。

□

最後にアルゴリズムの説明に用いられる用語を定義する。

定義 3.5 移動可能

移動可能な方向とは、その方向に移動した後に次の状態にならない方向を指す。

- 障害物の上にいる。
- 世界の外にいる。

□

3.1 アルゴリズム

お互いに合流する目的を持ったエージェントは以下の手順によって合流する。

1. 次の手順にしたがって、移動目標を決定する。

   (a) 両者とも合流する相手に自分の現在の座標を伝えるために、自分の位置を$\vec{X}$とすると、メッセージ *send-position*($\vec{X}$) を送信する。

   (b) 自分の位置と、相手から受信した位置が同じ位置の場合は、合流が完了したとしてこのアルゴリズムは終了する。

   (c) 自分の位置を (x_1, y_1)、相手から受信した位置を (x_2, y_2) すると、移動目標$\vec{G}$を、$\vec{G} = (\lfloor (x_1 + x_2)/2 \rfloor, \lfloor (y_1 + y_2)/2 \rfloor)$ と定義する。

2. 次の手順にしたがって、各々のエージェントが移動目標$\vec{G}$ の方向へ移動を試みる。

   (d) エージェントの位置を$\vec{a}$とし、k, lを正の整数とする。ただし複号同順とする。
   $\vec{G} - \vec{a} = (\pm k\vec{N}) + (\pm l\vec{E})$ において、

   $k = 0$ かつ $l \neq 0$ の場合 $\pm\vec{E}$の方向に移動する。

   $k \neq 0$ かつ $l = 0$ の場合 $\pm\vec{N}$の方向に移動する。

   $k = 0$ かつ $l = 0$ の場合 移動はしないが、移動行為は成功したものとみなす。

   その他の場合 $\pm\vec{N}$または$\pm\vec{E}$のどちらか移動可能である方向に移動する。両方向に移動可能である場合は、エージェント自身が移動方向を決定する。

一方でも (d) の試みに失敗した場合 自分が移動すべき方向に障害物などがあったエージェントは、移動を断念する。そして相手に移動目標の変更を交渉するために、メッセージ *send-direction*(*make-direction-set*($\vec{G}$)) を送信し、3 へいく。

- 両者とも移動に失敗した場合、両者ともにメッセージを送信しようとするが、メッセージの送信に成功するのは、どちらか一方であり、どちらが成功するかはランダムである。
- 移動可能であったエージェントはすでに移動が終了しているものとする。
- *make-direction-set*() は自分が回避すべき方向の集合を作成する関数であり、後に定義する。

両者とも (d) の試みに成功した場合 1 にもどる。

3. 次の手順にしたがって、移動目標の変更を行う。

(e) メッセージ *send-direction* を受信したエージェントは、受信したDIRECTION-SET を Receive、自分の *make-direction-set*($\vec{G}$) の結果を Result とすると、$\vec{F} \in$ (Receive $\cap$ Result) とおく。ただし過去に受信または送信したうち、最新の *accept*($\vec{X}$) が、$\vec{X} \in$ (Receive $\cap$ Result) となる場合は、$\vec{F} = \vec{X}$とする。

- DIRECTION-SET と *make-direction-set*($\vec{G}$) のどちらか一方でもϕの場合は、このアルゴリズムは失敗するが、この問題については本論文では扱わないことにする。

$\vec{F}$が存在する場合 相手に方向$\vec{F}$に変更することを受理したことを伝えるために、メッセージ *accept*($\vec{F}$) を送信する。メッセージを送信したエージェント、受信したエージェント共に、$\vec{G} = \vec{G} + \vec{F}$ を行い 2 へもどる。

$\vec{F}$が存在しない場合 相手に変更を受理できなかったことを伝えるために、メッセージ *reject*() を送信する。メッセージを送信したエージェント、受信したエージェントは各自$\vec{F} \in$ *make-direction-set*($\vec{G}$) となる$\vec{F}$の方向に移動し、1 へもどる。

- 集合中のどの方向が選ばれるかは、ランダムである。

次に上記のアルゴリズム中に使われた関数を定義する。

定義 3.6 *make-direction-set*($\vec{\Phi}$)

この関数は、現在の位置$\vec{a}$において移動可能な方向の集合を I、次によって求まる、$\vec{\Phi}$と反対の方向の集合を Jとすると、$I - J$を返す。ここで k, lは正の整数とする。また複号同順とする。

$\vec{G} - \vec{a} = (\pm k\vec{N}) + (\pm l\vec{E})$ において、

$k = 0$ かつ $l \neq 0$ の場合 $\mp\vec{E}$の 1 方向からなる集合。

$k \neq 0$ かつ $l = 0$ の場合 $\mp\vec{N}$の 1 方向からなる集合。

$k = 0$ かつ $l = 0$ の場合 空集合。

その他の場合 $\mp\vec{N}$、$\mp\vec{E}$の 2 方向からなる集合。

□

3.2 例

上で述べたミーティングアルゴリズムについて図 2 の場合を例にして説明する。

エージェント A の位置が$\vec{A} = (3,6)$、エージェント Bの位置が$\vec{B} = (4,2)$ であるので両者の合流の手順は次のようになる。

1. 移動目標を決定するために、両者が座標を交換しあう。その結果移動目標は、

$$\vec{G} = (\lfloor (X_a + X_b)/2 \rfloor, \lfloor (Y_a + Y_b)/2 \rfloor) = (\lfloor (3+4)/2 \rfloor, \lfloor (6+2)/2 \rfloor) = (3,4)$$

となる。両者の座標が異なるのでまだ合流していない。

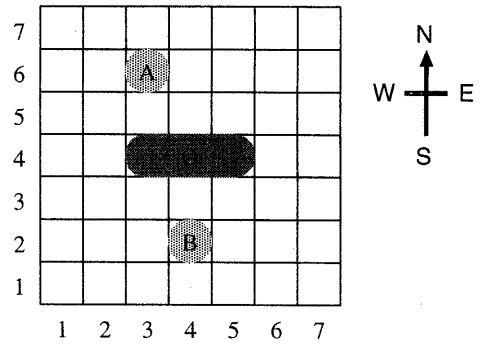

図 2: 例 1:初期状態

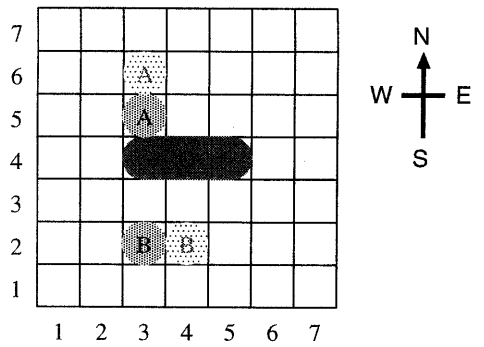

図 3: 例 1:A は南に移動不可能

2. 両者ともに移動目標$\vec{G} = (3,4)$に向かって移動する。A は南に移動して$\vec{A} = (3,5)$ となるのは明らかであるが、Bは移動可能な方向が北と西の 2 方向ある。ここでは、エージェント Bは西へ移動して$\vec{B} = (3,2)$ となったとする (図 3)。両者ともに移動に成功したので次は移動目標を決定する。

3. 同様に、両者の位置から移動目標は$\vec{G} = (3,3)$ となる。

4. 両者ともに移動目標$\vec{G}$に向かって移動する。Bは北に移動して$\vec{B} = (3,3)$ となるが、A は障害物 Oのために移動に失敗する。移動に失敗したので、A は Bに移動目標の変更を伝えるため、メッセージ *send-direction*$(\{\vec{W},\vec{E}\})$ を送信する。

5. エージェント Bはメッセージ *send-direction*を受信した。そこで、$\vec{F} \in \{\vec{W},\vec{E}\} \cap \{\vec{N},\vec{W},\vec{E}\}$ となる$\vec{F}$を求める。過去にメッセージ *accept*を受信または送信していないので、この$\vec{F}$が求めるべき方向である。ここで$\vec{F} = \vec{W}$を選択したとすると、メッセージ *accept*$(\vec{W})$ を A に送信し、両者の移動目標は$\vec{G} = (2,3)$ となる。

6. 両者ともに移動目標$\vec{G} = (2,3)$ に向かって移動し、$\vec{A} = (2,5)$, $\vec{B} = (2,3)$ となる。

7. 同様に、移動目標$\vec{G} = (2,4)$ を決め、移動すると$\vec{A} = (2,4)$, $\vec{B} = (2,4)$ となり、合流が終了する。

4 考察

3 節で示したアルゴリズムを使って次のような問題を解決することができる (図 4)。この場合、移動目標$\vec{G}$ には障害物があり、A も Bも移動に失敗し交渉が始まる。仮に A からの移動目標変更のメッセージが成功したものとする。A はメッセージ *send-direction*$(\{\vec{W},\vec{E}\})$ を送信する。この時 Bの動ける方向を示す Result も $\{\vec{W},\vec{E}\}$ なので、ここで Bが$\vec{F} = \vec{E}$を選択する場合と$\vec{F} = \vec{W}$を選択する場合がある。

$\vec{F} = \vec{E}$を選択した場合 一度$\vec{F} = \vec{E}$を選択すると、障害物がなくなるまで$\vec{E}$に移動し続ける。障害物の端まで来ると、これは 3.2 節で示した例の左右逆の状況なので、2 つのエージェントは合流することができる。

$\vec{F} = \vec{W}$を選択した場合 この場合、$\vec{W}$へ行くと、またしても障害物にぶつかり移動に失敗する。この時に行なわれる交渉では、Receive も Result も $\{\vec{E}\}$ なので$\vec{F} = \vec{E}$となる。その後は$\vec{F} = \vec{E}$を選択した場合と同じになり、最終的に 2 つのエージェントは合流することができる。

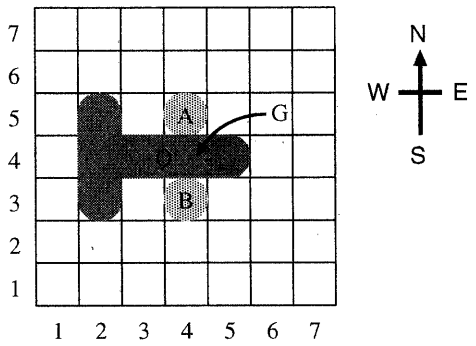

図 4: 例 2

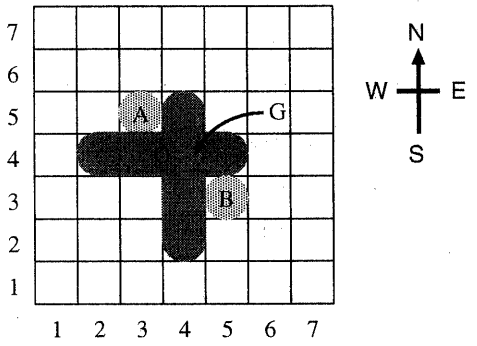

図 5: 集合できない例

このアルゴリズムでは次のような場合に合流することができない。これらを解決するのが今後の課題である。

設定した移動目標から離れる方向に動かなければ合流できない場合 図 5 のように設定した移動目標から一旦離れる方向に動かなければ最終的に合流することができないような場合、このアルゴリズムでは図の位置でエージェントが留まってしまって合流することはできない。

障害物があって決して合流することができない場合 決して合流することができないということを検出したい。しかし、このアルゴリズムでは同じ領域を永遠に行き来することになる。

5 おわりに

2 つのエージェントが交渉しながら合流するための簡単なモデルとアルゴリズムを示した。このモデルは、移動するたびに移動目標を計算しその移動目標に近付く方向へ移動する、という単純なものであるが、障害物があっても交渉によって移動目標を別なところに設定することによって 2 つのエージェントは合流することができる。

今後このモデルに対して次のような拡張を行なうことを考えている。

- 3 つ以上のエージェントの集結問題。
- 各エージェントの移動目標決定のアルゴリズムに違った性格づけを行った時の集結問題。
- 完全非同期なモデル化。

参考文献

[1] 石田亨, 桑原和宏, "分散人工知能 (1):協調問題解決", 人工知能学会誌 Vol.7, No.6, pp.945-954, 1992.

[2] 桑原和宏, 石田亨, "分散人工知能 (2):交渉と均衡化", 人工知能学会誌 Vol.8, No.1, pp.17-25, 1993.

[3] 横尾真, "分散探索とその周辺", コンピュータソフトウェア Vol.12, No.1, pp.33-42, January, 1995.

[4] 千村文彦, 所真理雄, "地図を用いた協調探索:複数エージェントの集結問題を例として", コンピュータソフトウェア Vol.12, No.1, pp.64-70, January, 1995.