

01. artisocレシピブック

基本的な操作について学び、モデルを動かそう

本ドキュメントについてのご質問、『複雑系勉強会』のお問合せは、下記までご連絡ください。

(株)構造計画研究所
社会デザイン・マーケティング部
artisocマーケティング担当 玉田
Tel: 052-222-8461
E-mail: tamada@kke.co.jp

社会現象をシミュレーションしよう

■ユーザフレンドリーなマルチエージェント・シミュレータ『artisoc』を体験します。

- 3つのステップではじめるマルチエージェント・シミュレーション
- エージェント同士を相互作用させよう！



3つのステップではじめるマルチエージェント・シミュレーション

■artisocでシミュレーションを行うための手順は次の通りです。

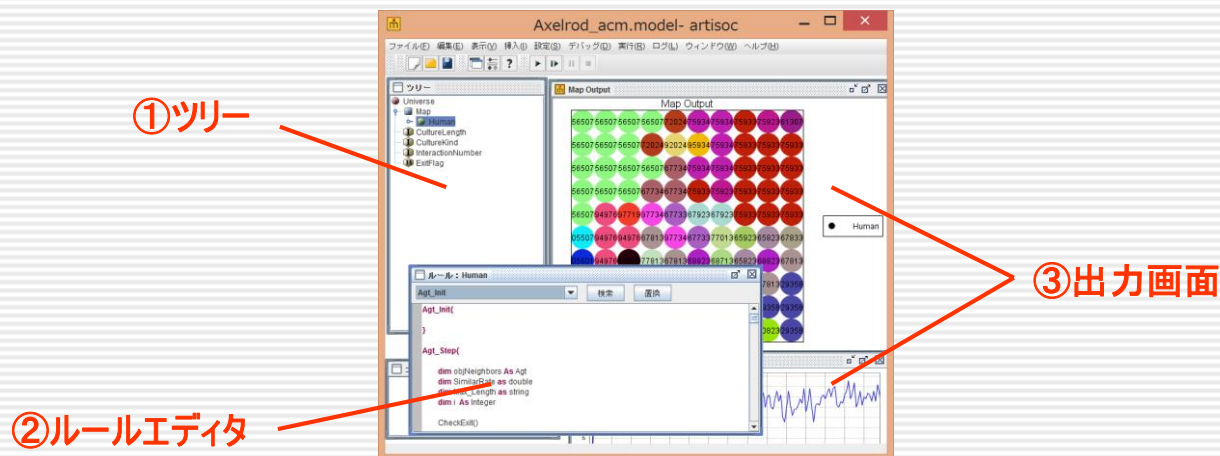
操作手順

ステップ1: ツリーで「エージェント」を定義します。

ステップ2: エージェントのルールを記述します。

ステップ3: 出力設定を行います。

シミュレーションを実行します。



① エージェントを定義する

■ ツリーで「空間」と「エージェント」と「変数」を定義します。

- ツリーの「Universe」で右クリックして、「空間の追加」を選択します。

□ 空間名: Map

- ツリーの「Map」で右クリックして、「エージェントの追加」を選択します。

□ エージェント型名: Person

□ 生成エージェント数: 3

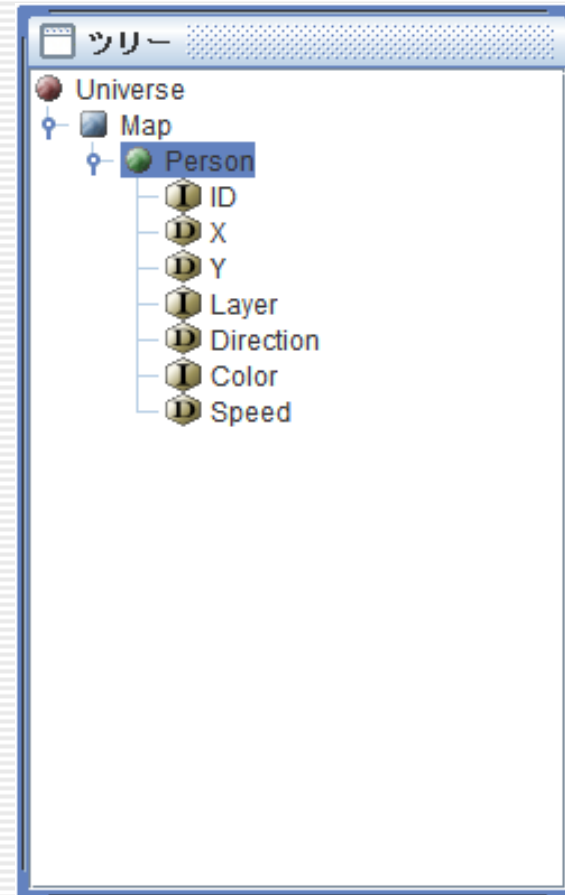
- ツリーの「Person」で右クリックして、「変数の追加」を選択します。

□ 変数名: Color

□ 変数の型: 整数型

□ 変数名: Speed

□ 変数の型: 実数型



② エージェントのルールを記述する

■ エージェントの色と歩行速度を指定し、画面中央に移動、向きを変える。

● ツリーの「Person」で右クリックして、「ルールエディタ」を選択します。

<pre>Agt_Init{ My.Color = COLOR_BLUE My.Speed = 1 MoveToCenter() Turn(Rnd() * 360) } Agt_Step{ Forward(My.Speed) }</pre>	<p>表示色は青色</p> <p>歩行速度は1</p> <p>画面中央に移動</p> <p>指定した角度(°)だけ回転する</p> <p>前に進む(マイナス値のときはバック)</p>
---	---

大文字、小文字の区別はありません。

行頭のタブ(インデント)はなくても動作しますが、入れることによりルールが見やすくなります。

「My.」と記述すると、属性の候補が表示されます。

関数名のはじめの数文字を入力した途中で Ctrl + SPACE キーを押すと関数の候補が表示されます。

関数の詳細については、ヘルプメニュー、

もしくはスタートメニューの[artisoc]–[documents]–[help Japanese] (help-ja.pdf)をご確認ください。

ルールエディタの文字が小さい場合は、設定メニューの[その他の設定]–[フォントの指定]で調整してください。

③ 出力設定を行う

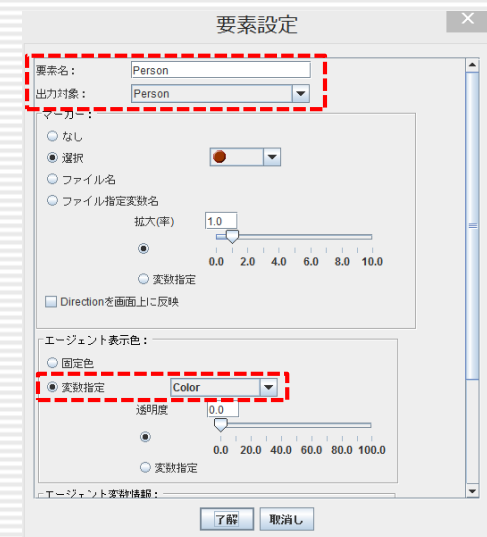
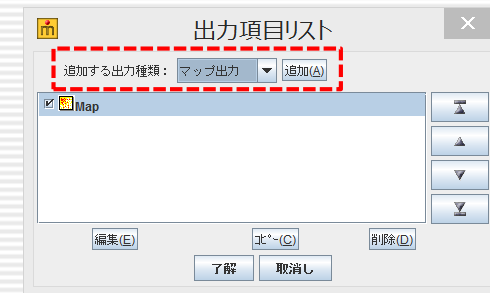
■ 出力項目の設定を行います。

- ツールバーの[設定]-[出力設定]を選択します。
- 出力項目リストが表示されますので、追加する出力種類を「マップ出力」にして「追加」ボタンをクリックします。

■ マップ名: Map

- マップ要素リストで「追加」ボタンをクリックします。

- 要素名: Person
- 出力対象: Person
- エージェント表示色:
 - 変数指定: Color



シミュレーションを実行します

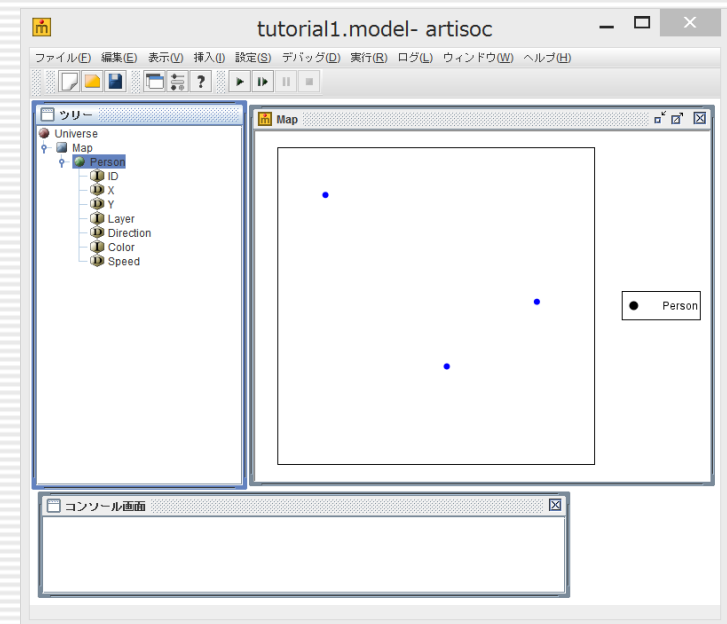
■ シミュレーションの実行と停止を操作します。

- 「ステップ実行ボタン」を何回かクリックして、ステップ毎の動きの違いを確認します。
- 「実行ボタン」をクリックしてシミュレーションを実行します。
- 「停止ボタン」をクリックしてシミュレーションを停止します。

「Person」の移動が速すぎる場合は、設定メニューの
[実行環境設定]-[実行ウェイト]を調整してください。

空間がループする場合、空間の上と下、右と左が
つながっています。

うまく動いたら、エージェントの数を100にして、
ステップ実行してみましょう。



01-1.model

エージェント同士を相互作用させよう！

■ぶつかると赤くなるエージェント

- エージェントをまとめて扱うときは、エージェント集合型という変数を使うことができます。
- 周りを見回して隣人を探します。

■ループしない空間とランダムな配置

- 空間がループしない場合は端点の処理が必要になります。
- エージェントを空間上にランダムに配置します。

■エージェントに個性を持たせる

- 違う色、違う歩行速度を持つエージェントを定義します。
- 会話をしているエージェントを線で結びます。

ぶつかると赤くなるエージェント

■ 隣人 (Neighbor) を定義して、エージェント同士を相互作用させます。

● ツリーの「Person」で右クリックして、「変数の追加」を選択します。

□ 変数名: Neighbor

□ 変数の型: エージェント集合型

● ツリーの「Person」で右クリックして、「ルールエディタ」を選択します。

Agt_Step{

Forward(My.Speed)

MakeAllAgtSetAroundOwn(My.Neighbor, 1, False)

If CountAgtSet(My.Neighbor) > 0 **Then**

Turn(Rnd() * 360)

My.Color = COLOR_RED

Else

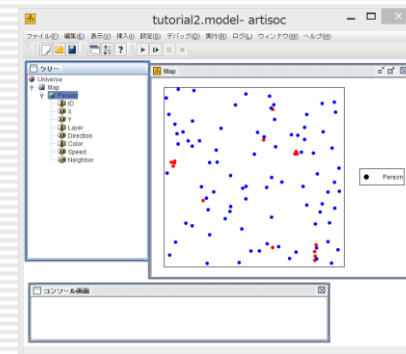
My.Color = COLOR_BLUE

End If

}

周りを見回して隣人を取得

隣人の数をカウントして
ぶつかっていたら赤くなる



01-2.model

ループしない空間とランダムな配置(1)

■ループしない空間を定義します。

- ツリーの「Map」で右クリックして、「プロパティ」を選択します。

□ 端点の処理: ループしない

- ツリーの「Universe」で右クリックして、「ルールエディタ」を選択します。

```
Univ_Init{
```

```
    Dim personAgtSet As AgtSet
```

```
    MakeAgtSet(personAgtSet, Universe.Map.Person)
```

```
    RandomPutAgtSet(personAgtSet)
```

```
}
```

変数の定義

Personの集合を取得

Personをランダムに配置

- ツリーの「Person」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Init{
```

```
    My.Color =COLOR_BLUE
```

```
    My.Speed = 1
```

```
    Turn(Rnd() * 360)
```

```
}
```

ループしない空間とランダムな配置(2)

■ 壁にぶつかったときの処理を加えます。

● ツリーの「Person」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Step[
```

```
  If Forward(My.Speed) <> -1 Then
```

```
    Turn(Rnd() * 360)
```

```
  End If
```

```
  MakeAllAgtSetAroundOwn(My.Neighbor, 1, False)
```

```
  If CountAgtSet(My.Neighbor) > 0 Then
```

```
    Turn(Rnd() * 360)
```

```
    My.Color = COLOR_RED
```

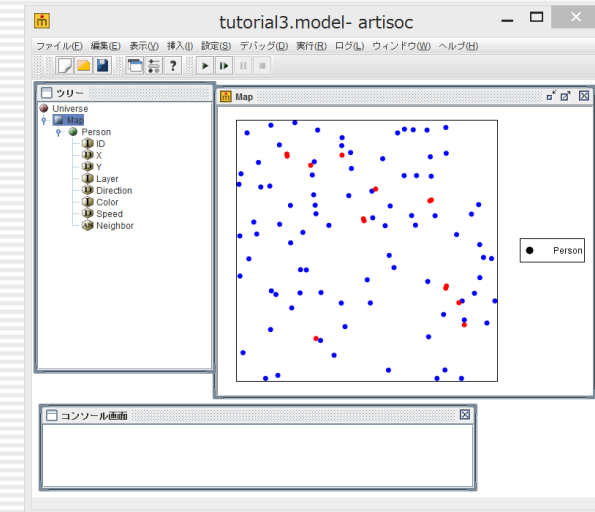
```
  Else
```

```
    My.Color = COLOR_BLUE
```

```
  End If
```

```
}
```

壁に当たったら向きを変える



01-3.model

エージェントに個性を持たせる(1)

■エージェント毎に異なった色「Color」と歩行速度「Speed」を持ちます。

●ツリーの「Person」で右クリックして、「変数の追加」を選択します。

□変数名: Touch

□変数の型: 整数

●ツリーの「Person」で右クリックして、「ルールエディタ」を選択します。

Agt_Init{

My.Color = RGB(CInt(Rnd()*256), CInt(Rnd()*256), CInt(Rnd()*256))

My.Speed = Rnd()

Turn(Rnd() * 360)

}

ランダムに色指定

歩行速度をランダムに定義

エージェントに個性を持たせる(2)

■色「Color」と歩行速度「Speed」と接触回数「Touch」をルールに組み入れます。

●ツリーの「Person」で右クリックして、「ルールエディタ」を選択します。

```
Agt_Step[
```

```
  If Forward(My.Speed) <> -1 Then
```

```
    Turn(Rnd() * 360)
```

```
  End If
```

```
  MakeAllAgtSetAroundOwn(My.Neighbor, 3, False)
```

```
  If CountAgtSet(My.Neighbor) > 0 Then
```

```
    My.Touch = My.Touch + 1
```

```
    Turn(Rnd() * 360)
```

```
  End If
```

```
}
```

視野を3にする

接触回数を加算

エージェントに個性を持たせる (3)

■出力項目の設定を行います。


- ツールバーの[設定]-[出力設定]を選択します。
- 出力項目リストが表示されますので、「Map」を選択して「編集」ボタンをクリックします。
- マップ要素リストが表示されますので、「Person」を選択して「編集」ボタンをクリックします。

■エージェント変数情報:

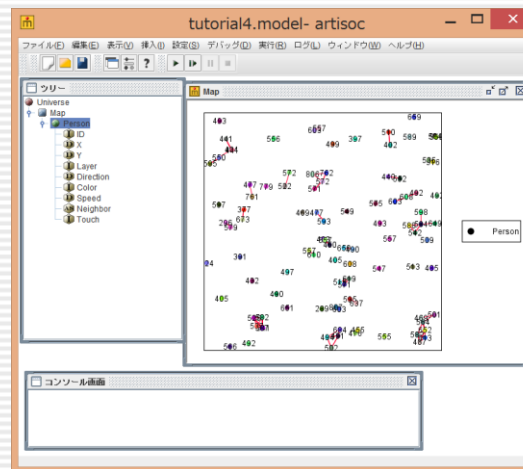
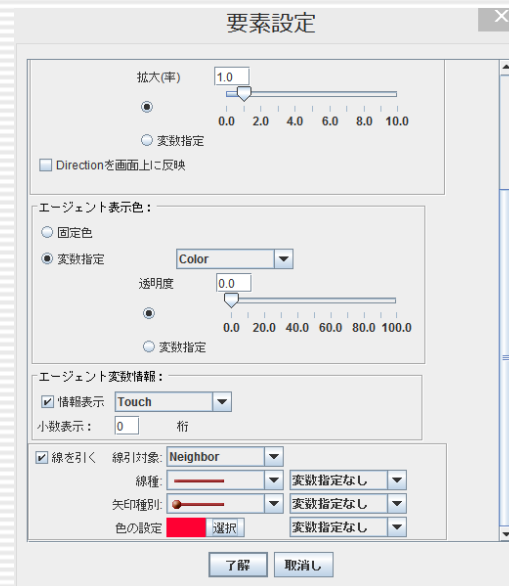
■ 情報表示: Touch

■線を引く

■ 線引き対象: Neighbor

■ 矢印種別: 

■ 色の選択: 赤



01-4.model

【参考】シミュレーション結果を分析するために

■シミュレーション結果をログ保存することで、計算結果を高速表示することができます。

- 実行メニュー の[記録して実行]を選択します。
- シミュレーションメモ入力ダイアログが表示されるので値を入力します。
 - シミュレーションメモ: テストログ
- 「停止ボタン」をクリックしてシミュレーションを停止します。
- ログメニューの [再生]を選択します。
- シミュレーション選択ダイアログが表示されるので「テストログ」を選択します。
- ログ再生パネルを利用して再生します。

