

ROSEリポジトリいばらき（茨城大学学術情報リポジトリ）

Title	エージェント指向モデルの一提案とその応用の試み
Author(s)	畠山, 正行 / 直井, 稔
Citation	全国大会講演論文集, 50(5): 69-70
Issue Date	1995-03-15
URL	http://hdl.handle.net/10109/1879
Rights	情報処理学会

このリポジトリに収録されているコンテンツの著作権は、それぞれの著作権者に帰属します。引用、転載、複製等される場合は、著作権法を遵守してください。

お問合せ先

茨城大学学術企画部学術情報課（図書館） 情報支援係
<http://www.lib.ibaraki.ac.jp/toiawase/toiawase.html>

エージェント指向モデルの一提案とその応用の試み

6J-6

畠山 正行 直井 稔

茨城大学工学部情報工学科

1 はじめに

現在、オブジェクト指向に基づいたコンピュータ・シミュレーションシステムが開発され、実用化されている。これによって、コンピュータ内の仮想世界に形成された対象世界が従来よりも柔軟にかつ複雑な世界をシミュレーションできるようになった [1]。

しかし、オブジェクト指向によるシミュレーションにも欠点がある。例えば、オブジェクトは他のオブジェクトからのメッセージパッシングによってのみ振舞う、いわば受動的なものなので、自らが振舞いを起こすことはできない。しかし、本来の対象世界では周囲の状況や自己の内部状態に合わせて振舞いを自身の判断で決め、振舞うのが最も自然なモデルである。このモデルを実現するのがエージェント指向モデルである [2]。

2 エージェント指向モデル

2.1 エージェント指向モデルの条件

我々はエージェントをオブジェクトの拡張と捉えているので、エージェントの条件としてはオブジェクトの条件に加えて、自律性とそれを実現するための自発性を持っていることであるとした。

自律性とは、目的をエージェント自体が持っている、その目的を達成すべくエージェント自身が周りの状況や自分の内部状態を評価し、判断して自らの振舞いを決定し、また、学習していく機能を指す。

また、エージェントは自己の目的にしたがって振舞うため、自分自身で振舞いを起こせなければならない。この機能が自発性である。

An Agent Oriented Computing Model and Its Applications.

Masayuki Hatakeyama Minoru Naoi
Ibaraki University

2.2 エージェント指向モデルの特徴

エージェント間の相互関係の発現はメッセージ通信であっても相互作用リンクであってもその内容はあくまでも命令ではなく、情報でなければならない。従って、メッセージを受けとったエージェントがそのメッセージをどう処理するかについては受けとったエージェントに任されている。

また、振舞いについてもエージェントは周囲の状況や情報および自身の内部状態によって自己の判断機構を起動して判断し、自己の振舞いを起動する。あくまで自身のメソッドは自身が起動・発現しなければならない。ただし、どちらの場合も例外がある。

3 応用例

3.1 2人蹴鞠

エージェント指向モデルの例の一つとして2人蹴鞠を実現した。この中でプレーヤがエージェント、ボールがオブジェクトとなっている。2人蹴鞠の目的は1バウンドでボールオブジェクトが相手エージェントに届くようにするものである。

3.1.1 ボール

これは、自発オブジェクトとして実装されている。自分の位置を計算して、自分の位置をブロードキャストしている。

3.1.2 プレーヤ

このエージェントは人間によって相手と使用するボールを指定されて起動されると、その相手に自発的に呼びかけてお互いに蹴鞠を始める。

このエージェントの目的はボールを相手に1バウンドでとどくようにすることである。

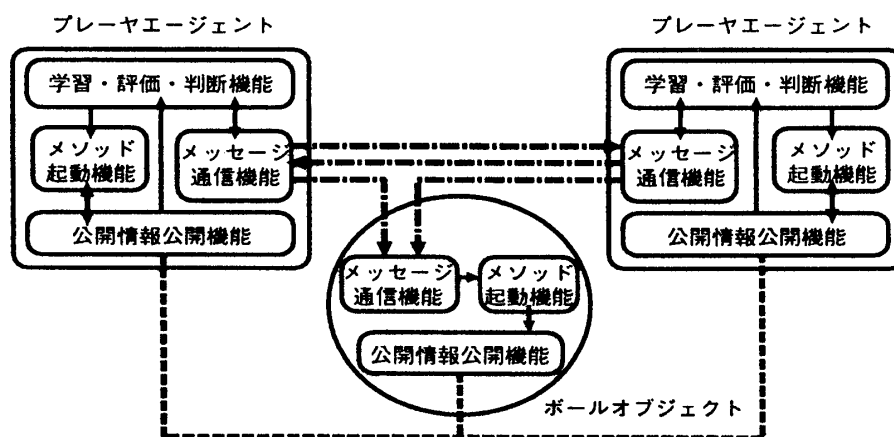


図 1: 二人蹴鞠の実装モデル

実装モデル図 1 からわかるようにオブジェクトとエージェントでは内部の情報の流れに違いがある。オブジェクトではメッセージ通信機能によって受けとったメッセージによってそのままメソッドを起動しているが、エージェントではメッセージ通信機能で受けとったメッセージも情報の一つとして学習・評価・判断機能に送られ、その判断に基づいてメソッド起動機能に情報が送られ、メソッドが起動されるという仕組みになっている。

3.2 流体計算の自動メッシュ生成への応用

まず、核となるエージェントを一つ用意する。これにメッシュのマップを与えると、自己のコピーを生成してリンクを張り、また、メッシュの周囲に当たったノードは境界ノードに変容する。各ノードは常に自分の周りのノードエージェントの状態を観測していて、その状態が変化したら自分の状態も変化させて、あるいはコピーを生成したり、消滅したりして適した状態を保つ。

また、メッシュの中に障害物が入った場合には、障害物と各エージェントが相互作用をすることによって、新たに境界ノードを生成し、不要になったら自己消滅する。

4 結果と考察・結論

今回どちらの例においても、C++ および C を用いて実装した。

2人蹴鞠の例では、解を求める方程式を用いずに数百回の学習によってうまく振舞うようになった。

また、人間がボールに働く重力をゲームの最中に変化させることができる。この場合でも、学習するうちにうまく振舞うようになった。

メッシュの自動生成の例では、プログラミングにおいて、メッシュ全体の状態をほとんど考慮する必要がなく、ノードの周囲とお互いの相互作用について精密に記述しておけばよく、また、動的な変更に対しても柔軟に対応できた。

エージェント指向モデルでは、オブジェクト指向モデルのようにトップダウンでなく、ボトムアップでシステムが構築できる。つまり、局所的なところを精密に実現すれば、大域的な結果はエージェント同士の相互作用の結果として現れる。これからも、オブジェクトよりエージェントの方がシミュレーションには適しているといえよう。

参考文献

- [1] 畠山正行、金子 勇、「オブジェクトベース機構：オブジェクト指向一貫モデリング過程論に基づくシミュレーションの実現」、情報処理学会第 17 回プログラミング研究会研究報告、Vol.94, No.49, PP.33 - 44, 1994 年 6 月 3 日
- [2] "Intelligent Agents", Special Issue of the 'Communications of the ACM', Vol. 37, No.7, pp.18-147, July 1994.