

1N-07 マルチエージェント理論構築のための 1N-7 離散時間記号シミュレータの開発

石田 和成
電気通信大学 大学院
情報システム学研究科
情報ネットワーク学専攻
ishida@ohta.is.uec.ac.jp

立花 裕
日揮 株式会社
yutaka@ohta.is.uec.ac.jp

山本 匡
電気通信大学 大学院
情報システム学研究科
情報システム運用学専攻
yamamoto@ohta.is.uec.ac.jp

太田 敏澄
電気通信大学 大学院
情報システム学研究科
情報システム運用学専攻
ohta@ohta.is.uec.ac.jp

1 はじめに

本論文は、離散時間のマルチエージェント理論構築のためのシミュレータの開発について報告するものである。離散時間マルチエージェントモデルとして記述される、社会、経済、経営などの事象に関して、本シミュレータは簡潔で柔軟なモデル構築・検証方法を提供する。

本シミュレータは、モデル構築者がモデルに専念できるように、データ構造として黑板と一階述語、シミュレーションエンジンとして代替案生成・選択機構を持つことによって、言語の習得、モデルのプログラミング、そしてモデルのデバッグに関するモデル構築者の負担を軽減する。シミュレータにおけるモデル記述言語の詳細については[1]で報告した。

本論文では、基本的データ構造とシミュレーションエンジンを持つ本シミュレータが、マルチエージェントモデルの構築のための有効なツールの1つとなることを示すために、マルチエージェントモデルの例として、ファミリーレストランの厨房におけるコックの間の協調現象を取り上げ、そのモデル化の過程に沿って、本シミュレーションツールの説明を行なう。

2 モデル化の過程

本節では、マルチエージェントモデルの例として、ファミリーレストランの厨房におけるコックの協調現象を取り上げ、モデル化の過程を説明する。

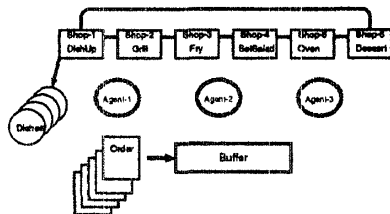


図1 レストランの調理場

ファミリーレストランの厨房では、「調理人」が、オープンやグリルなどの「調理場」を用いて、お客の

注文する料理を「調理」する。ここで、コックを「エージェント」、調理場を「ショップ」、調理を「ジョブ」と定義し、シミュレーションが実行可能な詳細なモデル化を進める。

エージェントのモデル化: エージェントのデータ構造を考える。複数のエージェントを個別に認識するための名前と、状態を考慮して、エージェントの述語テンプレートを、AGENT(<Name>,<Stat>)と定義する。

エージェント間の協調形態のモデル化: コックの間の協調に関して、現実のファミリーレストランのキッチンにおいて人の配置の仕方を観察した結果、「集中型」、「分散型」、「統合型」という3種類の協調形態が存在することが分かった。

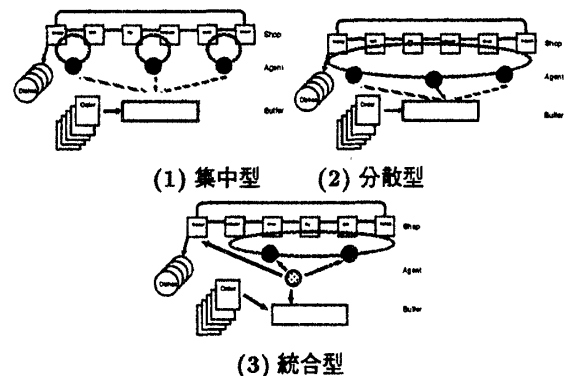


図2 エージェントの協調形態

第1の配置である集中型は、エージェントの受け持ちショップが、エージェント間で重複することなく配分されるというものである。第2の配置である分散型は、全てのショップが全てのエージェントによって共有されているというものである。第3の配置である統合型は、一人のエージェントが調整に特化し、他のエージェントの調理に関する指示を行い、その指示に従って、他のエージェントは柔軟に様々なショップを担当するものである。

この3つの協調形態は、ショップの使用権 Shop Permission(<Shop>, <Agent>) とエージェント間の管理関係 COORDINATED(<Worker>, <Coordinator>) によって、それらの間の構造上の相違を記述することができる。

3 モデル実行の過程

前節で示したモデルのプログラムを実行するためには、(1) モデルを表現するプログラムの意味的誤りを排除し、(2) シミュレーション実行時における、環境パラメータの変更や、モデル内の確率的要素の平均的な影響を知るため、同一環境パラメータにおける複数回数のシミュレーションの自動化、などが必要となる。

プログラムの意味的誤り検出: スケジューリングにおいて、プログラムの誤りによる実行不可能なスケジュールが生成されるならば、妥当なシミュレーションを行なうことができない。通常のプログラミング言語において、このような誤りを検出するには多くのプログラミングの手間を要する。しかし、本記述言語では、パターンマッチによって処理系が動作するので、デバッガとして、モデルにおいて存在が許されないパターンを記述しておけば、モデルを表現するプログラムの意味的な誤りが検出可能となる。例えば、モデル上における意味的誤りとして、実際には実行できないにも関わらず、同一の機械を使用するスケジュールに時間的な重複がある場合を検出する場合には、以下のような条件部を持つテンプレートを記述すればよい。

```
SCHEDULE(<Machine>$mc1, <Start>$st1, <End>$ed1)
SCHEDULE(<Machine>$mc2, <Start>$st2, <End>$ed2)
#equal($mc1, $mc2) // 同一の機械を使用

[#range($st1, $st2, $ed1) || #range($st1, $ed2, $ed1)]
```

シミュレーション条件の制御: 本処理系では、シミュレーションコントローラを記述することによって、モデルのパラメータを自動的に変更しながら、各パラメータにおいて、設定した回数のシミュレーション試行を行なうことができる。レストランモデルにおける環境パラメータは、ロード、すなわち開店中のオーダーの発生数である。またモデルにおいて発生する注文の種類や頻度は確率的な要素を含んでいるので、同一の環境条件のシミュレーションを複数行った後、平均値計算処理を行う。

シミュレーション結果の例: シミュレーションにおいて用いられる構造比較に用いる指標は、お客が注文して配膳されるまでの、お客の平均待ち時間である。シミュレーションにおけるパラメータは、ロード、す

なわち開店中のオーダーの発生数である。シミュレーション制御コントローラを用いてモデルを実行すると、図3に示す、集中型、分散型、統合型の比較結果が得られた。

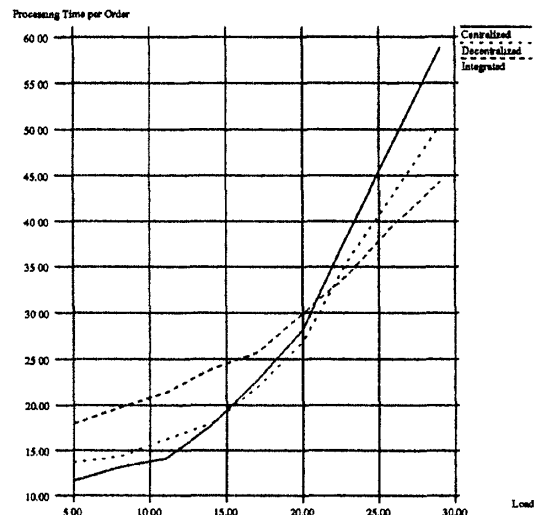


図3 シミュレーション結果

この結果によれば、ロードが低い場合には集中型が、ロードが中程度の場合には分散型が、そしてロードが高い場合には統合型がそれぞれお客の待ち時間が少ないことが分かる。モデル構築者は、ここで得られた結果を考察し、結論を導く、あるいはモデルの振舞いが異常であればモデルのプログラムのデバッグを行なう。

4 結論

本論文では、マルチエージェント理論構築のための離散時間記号シミュレータ開発について報告した。マルチエージェントモデルの例として、ファミリーレストランの厨房におけるコックの間の協調現象を取り上げ、そのモデル化の過程に沿って、本シミュレーションツールの説明を行なうことによって、基本的データ構造とシミュレーションエンジンを持つ本シミュレータが、マルチエージェントモデルの構築のための有効なツールの1つとなることを示した。

参考文献

- [1] 石田 和成 複数知的主体の相互作用の時間発展に関する理論モデル記述言語とその実装. 情報処理学会研究報告書, 96 - AI - 106, pp. 1-8, 1996.