7D－4

マルチエージェントモデルに基づく実時間分散制御システムの提案*

根岸 征史† 中内 靖‡ 安西 祐一郎†

慶應義塾大学† 防衛大学校‡

1 はじめに

分散環境において、複数の自律エージェントの協調作業を支援するシステムの研究が近年活発に行われている [1]。協調作業の形態には、人間同士の協調作業や人間と自律移動ロボットとの協調作業など様々な形態がある。従ってこのようなシステムは、エージェント間の様々な形態のコミュニケーションを記述することのできる汎用モデルに基づいている必要がある。また一般に、エージェントが実行するタスクには、デッドライン等の実時間制約があり、これを満足することが望まれる。

そこで本研究では、実時間制約を満足するマルチエージェントモデルならびにマルチエージェント記述言語 MALOC(Maluti Agent description Language based On C++) を提案する。また、本システムの有効性を示すために開発した、OA 機器の分散制御のシステムについて説明する。

2 エージェント記述言語

マルチエージェント環境においてエージェントの記述を行うための言語として、エージェント指向プログラミング (Agent-Oriented Programing、以下 AOP) のフレームワークに基づいた、AGENT-0 が提案されている [2]。AOP はオブジェクト指向プログラミングを特殊化したものであり、いくつか相違点がある。まず、エージェントは mental state と呼ばれる内部状態を持つ。また、発話行為理論に基づいて、通知、要求、提供、拒否、受諾等のメッセージの種類を区別している。

AGENT-0 では mental state として belief、commitment、capability をデータベースとして持ち、メッセージとしては通知 (Inform)、要求 (Request) の二種類を扱っている。AGENT-0 で記述されたプログラムは Agent Interpreter によって処理される。あるメッセージがエージェントに到着すると、mental state を参照し、必要ならばこれを更新する。また、メッセージの種類が Request であれば、プログラムに記述されるルールに従って、要求されたアクションを受理するかどうかの判断を行う。受理されたアクションはメッセージに明示された時刻に実行される。

AGENT-0 では、制限されたメッセージと mental state によって様々なコミュニケーションの記述が可能である。しかしながら、アクションはある特定の時刻に実行するという記述しかできず、実時間制約の保証という観点からは現実に即したものとは言えない。

そこで、本研究で提案する MALOC は AOP のフレームワークに基づき、さらにデッドライン等の実時間制約を満足することを目的として設計する。

3 設計

MALOC の設計では以下のことを基本方針とした。

- マルチエージェント環境に対応していること。
- エージェント間のコミュニケーションを制限されたメッセージで記述できること。
- デッドライン等の実時間制約を満足すること。

3.1 MALOC のマルチエージェントモデル

エージェントは、固有の内部状態を持ち、他からの依頼に基づいて実行される様々な手続き (アクション) を持つ実行主体である。

エージェント間の通信はメッセージの送受信によって行い、メッセージは到着した順に処理される。

メッセージの種類は通知 (Inform) と要求 (Request) の二種類に制限する。Inform メッセージはエージェントの内部状態の参照や通知を行うものである。一方、Request メッセージはタスクの実行を要求するものであり、デッドライン等の時間的制約を明示することができる。要求されたタスクは実時間制約を満足するようにスケジューラがスケジューリングを行う。スケジューリングアルゴリズムは最短デッドライン優先方式で行われる。

また、アクションは他のエージェントからは直接参照することのできない private action と、メッセージによって参照できる communicative action の二種類に区別される。communicative action には request action および inform action があり、それぞれ Request メッセージ、Inform メッセージによって参照される。

request action 内ではタスクのスケジューリングを行うために commit action を起動できる。commit action はシステムが持っているアクションであり、アクション内でスケジューラを起動して、タスクの実時間制約を満たすようにスケジューリングを行う。

メッセージの送信は、ある特定のエージェントへの送信だけでなく、ある条件を満たす全てのエージェントに対してメッセージを送信するマルチキャストも可能とする。

*Distributed real-time system based on multi-agent model
Masafumi Negishi†, Yasushi Nakauchi‡ Yuichiro Anzai†
Keio University†, National Defense Academy‡

3.2 言語仕様

MALOC は C++のシンタックスに基づいている。エージェントの宣言では、内部変数の定義とアクションの定義が行われる (図 1)。アクション内で他のエージェントにメッセージを送信する時には (エージェント名)-> (アクション名) と記述する (図 2)。

```
agent printer
{
        "内部変数の定義"
        "アクションの定義"
}
```

図 1: エージェントの宣言例

```
request_print(引数)
{
  if(条件){
    commit(…);
    …
    printer1->inform_accepted(…);
  }
  …
  printer1->inform_reject(…);
}
```

図 2: アクションの記述例 (request action)

4 実装

MALOCは、C++を基本言語とし、C++へのプリプロセッサとして実装されている。エージェントはC++のオブジェクトとして管理される。各エージェントはメッセージキューを持ち、メッセージの処理やタスクの実行を行うマネージャを保持する。main 関数のループ内でシステムの時刻はインクリメントされ、各エージェントのマネージャが逐次的に起動される。

また、ユーザがエージェントとしてメッセージを送信するためのインタフェースを X-Window システム上に Motif を用いて実装した。

5 評価および考察

MALOC の有効性を示すために OA 機器の分散制御シミュレータを開発した。このシミュレータはエージェントとしてユーザの他に OA 機器を扱い、ユーザからランダムに送信されるメッセージを処理する。 また、エージェント間で効率の良い負荷分散を実現するために、ユーザエージェントはタスクの内容 (実時間制約など) をプリンタエージェントに対してブロードキャストし、プリンタエージェントからの入札を待って、その中から最適なものを判断してタスクを依頼することを可能としている。

このシミュレータを用いて評価する項目として、エージェント毎のタスクの受理率、タスクの依頼が行われてから実際にタスクが実行されるまでの応答時間等が挙げられる。

評価方法としては、

1. ユーザエージェントはタスクを依頼する時全てにおいて、ある特定のプリンタエージェントを指定して行う。
2. ユーザエージェントはタスクを依頼する時全てにおいて、まずタスク内容をブロードキャストして、プリンタエージェントからの入札を待ち、最も早く実行されるものにタスクを依頼する。またこの時、入札するエージェントは自分が落札されることを想定して、タスクキューにそのタスクをあらかじめ確保しておく。
3. ユーザエージェントは、タスクを依頼する時全てにおいて、まずブロードキャストして、入札を待ってから落札を行う。しかし、2.の場合と違い、入札するエージェントはタスクキューにそのタスクを確保しない。

等が考えられる。

特定のプリンタエージェントに対してタスクを依頼する場合は、ユーザはプリンタの内部状態を知らないためタスクの受理率は低くなり、ブロードキャストする場合は、最適なエージェントをユーザが選択できるのでタスク受理率は高くなる、と予想される。

一方、ブロードキャストした場合にはエージェント間の交渉により遅れが生じるので、交渉を行わない時に比べて応答時間が長くなると予想される。

また、ブロードキャストされたタスクをキューに確保するか、しないかでタスクの受理率、応答時間にどのような変化があるかも比較する必要があるだろう。

6 まとめ

分散環境において複数の自律エージェントが行う協調作業を支援するシステムを構築するにあたり、エージェント間のコミュニケーションを記述するためのモデル、およびマルチエージェント記述言語 MALOC を提案した。今後 MALOC の有効性を示すには、5 章で述べた評価方法にしたがって評価を行い、さらに現実に即したアプリケーションを開発する必要がある。

参考文献

[1] Y.Nakauchi, K.Kawasugi, T.Okada, N.Yamasaki and Y.Anzai. Human-Robot Interface Architecture for Distributed Environments. In Proc. of IEEE International Workshop on Robot and human Communication(ROMAN'92). 1992

[2] Y.Shoham. Agent-oriented programing. Technical Report STAN-CS-90-1335, Computer Science Stanford University, 1990.