

マルチエージェントシステムのための通信プロトコルとその効率分析

2 A F - 4

佐藤 義顕

沢村 一

新潟大学大学院自然科学研究科 新潟大学工学部情報工学科

1 はじめに

マルチエージェントシステムでは、エージェント間の協調 (cooperation), 交渉 (negotiation) などの相互作用をいかに効率良く実現するかが非常に重要である。一般的にエージェントとは、外部環境との相互作用に基づいて、状況を把握したり、他の処理体と協力しながら、ある目標に向かって自律的に問題解決を進めていく処理体と捉えることができる。ここでエージェント間ではどのメッセージをどのタイミングで送信したら良いか、またメッセージを受信したエージェントはどのように振る舞えばよいかという相互間の取り決め、すなわち通信プロトコルが問題となる。

分散人工知能の研究分野では、契約ネットプロトコル [1] など、エージェント間の協調、交渉を実現するための数多くの通信プロトコルが提案されている。これらのプロトコルは応用目的に応じて拡張などが行われ、また実行環境や実装言語なども様々である。つまり、応用目的の違いによって効率のよい通信プロトコルを選択しているといえる。また、最近ソフトウェアエージェントについての研究が盛んになり、応用目的に応じていろいろな通信プロトコルが必要になってきている。ここで、エージェントが存在する環境や状況、あるいは解決しようとする問題によって効率的な通信プロトコルを選択し実行することが可能であれば、マルチエージェントシステム全体の効率を上げたり、応用範囲なども広げることが可能になる。しかし、既存のマルチエージェントシステムは動作する環境や通信プロトコルに依存し、通信プロトコルそのものを性質や効率の立場で比較することは難しい。

本論文では、ネットワーク環境でマルチエージェントシステムにおける種々の通信プロトコルの実行を行うことが可能なソフトウェアエージェント、つまりエージェントの知識、通信部分の設計方法を提案し、それに基づく実装について述べる。

2 エージェントの機能

契約ネットプロトコルなどの通信プロトコルの効率を比較するには、まずプロトコルを実行するためのエージェントのシステムを実装する必要がある。我々が実現したエージェントの特徴的な機能は以下の通りである

- エージェントは、システム上の通信や制御に依存しない、知識を定義できる機能を持つ。

- 学習能力。問題解決の過程においてエージェントの自己知識や社会知識は強化される。エージェントの動作や契約はこのエージェントの経験によって改善されていく。
- 通信部分を知識部分、処理部分から分割する。これにより、問題の種類、状況の変化に応じて様々な通信プロトコルを選択して実行することができる。

3 エージェントの構造

エージェントの構造は機能別に大きく3つのレベルのクラスに分解する。それは知識クラス、コミュニケーションクラス、処理クラスである。

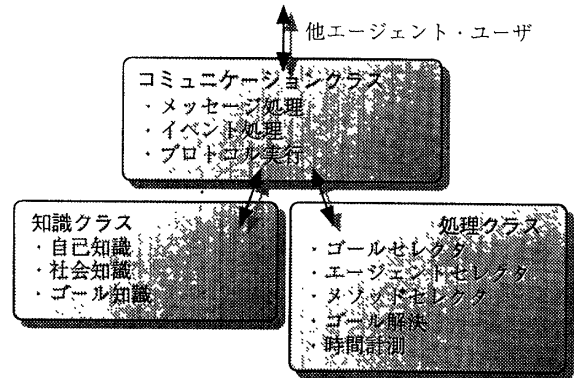


図 1: エージェントのモデル

3.1 知識クラス

自己知識: これはネットワークアドレスなどの自分自身の情報や、自分の知識、つまり自分が解決することのできるゴールの種類の集まりによって記述される。解決すべきゴールを受信した場合、エージェントはそのゴールを解くための自己知識を有するか否かをチェックする。

社会知識: これはエージェントが他のエージェント群、つまり社会に対して持っている知識のことである。契約ネットプロトコルの場合、エージェントが自分の持っている知識によって与えられたゴールを解決できない時、他のエージェントにそのゴールを契約によって渡す。もし他のエージェントについての情報を持っている場合、直接そのエージェントに対して契約を申し出るが、情報を持っていなかった場合、すべてのエージェントに向けて誰かこのゴールを解決することができないか?と尋ねるメッセージを送る。この「どのエージェントがどのゴールを解決することができるか」という情報が社会知識である。

ゴール知識：これは、それぞれのゴールがどの程度まで解決されているかというラベルを格納するものである。ラベルは以下のものがある。実行する通信プロトコルによっては使用されないラベルもある。ラベル以外にもゴール知識はそのゴールの期限やゴールに入札しているエージェントのリストなどを持っている。

- 通知 : ANNOUNCE
- 未解決 : UNSOLVED
- 解決中 : SOLVING
- 解決済み : SOLVED
- 未契約 : UNCONTRACTED
- 契約中 : CONTRACTING
- 契約済み : CONTRACTED
- 失敗 : FAIL
- 待機中 : WAITING

3.2 コミュニケーションクラス

コミュニケーションクラスは通信を行うためのクラスである。通信にはTCP/IPを利用して行う。

メッセージ処理：これはメッセージをトークンに分解したり、一連のトークンからメッセージの文字列を生成する。

イベント処理：緊急メッセージなどの実現、ユーザとの対話などで使用する。

プロトコル実行：これは通信プロトコルに従って、ゴールの生成、メッセージのやり取りを行う。

例えば、TASK_ANNOUNCEMENTであれば契約のためのゴールの生成、AWARDやDIRECTED_AWARDであれば解決のためのゴールの生成などを行う。

これらのクラスは通信プロトコル別にそれぞれ設計される。知識クラス、処理クラスは通信プロトコルに依存しないので、通信プロトコルに対応したコミュニケーションクラスをロードすることにより、知識クラスや処理クラスが共通なエージェント上で、様々な通信プロトコルを実行することができる。

3.3 処理クラス

処理クラスはエージェントのメインとなる部分である。実際にタスク分割、ゴール解決、それらに必要な処理などを行う。

ゴールセクタ：これはコミュニケーションクラスのプロトコル実行クラスから生成されるゴール群から、それぞれのゴールのステータスや、優先度から一つを選択する。その後、メソッドセクタを起動する。

エージェントセクタ：これはあるゴールを解決することができるエージェント群から、一番能力が高いものを選択する。

メソッドセクタ：ゴールが未解決のラベルをもっている場合、このゴールを解決するためのメソッドあるいはクラスを選択する。

ゴール解決：メソッドセクタによって選択されたクラスをロードし、オブジェクトの生成、実行を行う。

時間計測：これは入札期間の計測や、ゴールの期限切れ、一定時間以上待っても返答の来ないエージェントを調べるためなどに使用される。

4 実装と実験内容

以上で述べたモデルに基づいて実装を行った。実装言語としてはプログラミング言語Javaを使用している。インターネットの発達に伴い、将来このようなネットワーク環境下でJavaによるソフトウェアエージェントの研究、開発が増加するであろう。Javaで実装することにより、TCP/IPを利用してインターネットなどのネットワーク環境に適用でき、プラットフォーム独立な実行可能オブジェクトが生成され、セキュリティなどが保証される機能を付加できるという利点も持つ。現在は[2]で取り上げられている緊急医療補助問題を例題として契約ネットプロトコルによる通信を動作させている。この例題では5つの種類のエージェント(図2)が互いに通信し、事故の種類判断、事故現場の特定、緊急車両の選択を行う。

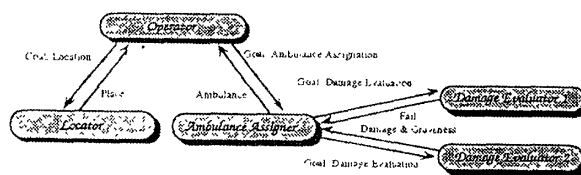


図 2: 緊急医療補助問題

このようなマルチエージェントシステム上で、エージェントの数、動作システムの負荷、ネットワークの負荷の変化による、効率のよい最適な通信プロトコルを実験によって調べる。効率のよい最適なプロトコルの基準としては、通信回数やゴール解決に必要な実時間などによって判定する。

5 まとめ

本稿では様々な通信プロトコルを実行できるエージェントのモデルを提案し、またその実装について述べた。今後は、このエージェントのモデルを使った黑板モデル、回覧板モデル、多段階交渉プロトコルや、統合交渉プロトコルの実装、および実装した場合の通信プロトコル同士の比較方法を検討していく。具体的な応用として、オークションなどの実際に人間社会に存在するプロトコルに適用することを目指す。

参考文献

- [1] R.G.Smith, *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver*. IEEE Transactions on Computers, C-29(12), pp. 1104-1113, 1980.
- [2] Karmelo Urzelai, Francisco J. Garijo., *MAKILA: A Tool for the Development of Cooperative Societies*. LNAI, vol.830, Springer, 1992, pp. 311-323.