

## 7F-7 分散協調型問題解決におけるプロトコル

平田 謙 中森 勝 井内 稔 山崎 晴明  
山梨大学

## 1. はじめに

人間社会においてグループで問題解決をはかることは一般的なことである。グループで問題を解決することは1人で問題を解決するときとくらべ、より柔軟に大きな問題に取り組むことができる。また1人では解決できない問題でもグループで取り組むことによって解決できることがしばしばある。このとき人間は意識せずに協調をおこなっている。グループで問題解決を行うとき、協調のための情報は会議や個々のコミュニケーションで交換される。これらの問題解決方法をシステム化しコンピュータ上に実現したものを分散協調型問題解決という。分散協調型問題解決では複数の問題解決エージェントを協調させることにより問題の解決をはかる。協調は通信路により相互接続されている個々のエージェントが、必要な情報や推論の中間結果を交換することによりおこなわれる。我々は分散協調型問題解決における協調という動作をプロトコルの立場から研究し試作をおこなった。

## 2. 分散協調型問題解決

分散協調型問題解決とはAIと分散処理が重なった分野である。個々のエージェントは疎結合のネットワークを通して協調をおこない問題解決をはかる。<sup>1)</sup>そのために推論、情報の交換すなわち協調プロトコルの必要が生じる。

このような協調プロトコルのひとつに契約ネットプロトコル<sup>1)</sup>がある。このプロトコルではタスクの報知、入札、落札という形でタスクを個々のエージェントに割り当てる。これに対し我々は個々のエージェントが推論の中間結果を質問、応答、提案という形で他のエージェントに知らせるといった協調プロトコルの研究と試作をおこなった。

## 3. 協調プロトコル

我々は図1に示すシステムの試作をおこなった。エージェントに3台、プロトコルマネージャおよびユーザインターフェイスに1台のPCを割り当てる。個々のエージェントには異なる知識表現ベースの推論機構を持つものをそれぞれ試作した。それらの知識表現は意味ネットワーク、フレーム、プロダクションシステムの3つであり、各エージェントはその知識表現の特徴を活かした推論をおこなう。また個々のエージェントは通信路によりプロトコルマネージャに相互接続しており、プロトコルマネージャを介して他のエージェントと推論の中間結果を送受信する機能を持つ。プロトコルマネージャは協調プロトコルを用い、エージェント間の協調動作を支援する。このシステムにおいて、プロトコルマネージャはプロトコルの集中管理をおこなっている。これは複雑なプロトコル管理を分散化しておこなうことがPCという制約上困難なためである。したがって、現在のシステムでのプロトコルマネージャは分散協調プロトコルのシミュレータとして位置づけされる。

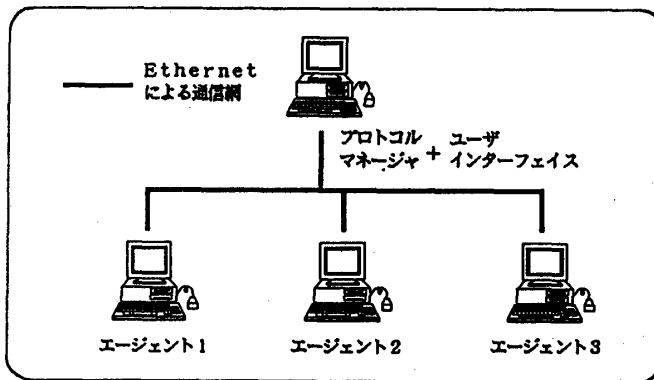


図1. システムの構成

## 3.1. 異なる知識表現エージェントによる協調

我々のシステムにおいて個々のエージェントはそれぞれ異なる知識表現ベースによる推論をおこなう。したがって協調動作において各エージェントの推論の結果を他のエージェントで扱うため、推論過程の表現に共通の表現形式が必要となる。

### 3.2. システムの問題解決動作

まずユーザインターフェースから問題が入力されると、各エージェントはプロトコルマネージャを通して問題を受け取り推論を開始する。エージェントは、推論の中間結果をプロトコルマネージャ上に送りながら推論を進める。送られた中間結果は、プロトコルマネージャで木構造を形成する。エージェントは定期的に木を参照し、自分の推論を継続するか、他の中間結果に移行するかを判断する。その判断基準は中間結果に付加された評価値でおこなう。エージェントは推論の継続が不可能となったとき他のエージェントに対し「質問」をおこなう。つまり「質問」の属性を持つ中間結果をプロトコルマネージャに送る。また木を参照したときに「質問」を受け取ったエージェントは、優先的に「質問」についての推論をおこない、その推論結果を「応答」としてプロトコルマネージャに送る。これらを繰り返す間に評価値の低い推論過程は淘汰され、すべてのエージェントが評価値の高い方向に向かって推論を進めるようになる。その結果、最終的に一つの結果に収束する。またエージェントは中間結果を積極的に「提案」することにより自身を含むエージェントの推論の継続が不可能となった場合や評価値の低い中間結果にいたった場合、その「提案」を参照することにより新たな推論を始めることができる。

### 3.3. 推論過程の表現

木におけるノードは、推論過程におけるある状態を表す。そのノードの子ノードは、その状態から推論により推移した、つまり推論が1段階進んだ状態を表している。ノードの情報はプロトコルマネージャの中にユニークな番号とともに保持される。このユニークな番号は、そのノードの木における位置のルートノードからの推論の履歴そのものである。例を図2に示す。図中の各ノードは、それぞれ推論の中間結果を表す。各ノードは2つ組  $(m, AG_n)$  で表され、 $AG_n$  はそのノードを木に加えたエージェントの番号を示し、 $m$  は親のノードから見たときにそのノードが、 $AG_n$  で表されたエージェントにとって、自分が木に加えた何番目の子ノードかを表す。図中の二重の枠で囲まれたノードを表す番号は、

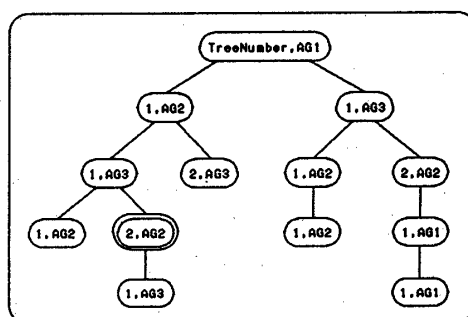


図2. 推論過程の木構造

$(TreeNumber, AG1) (1, AG2) (1, AG3) (2, AG2)$

と表される。この場合木が深くなる、つまり推論が進むにしたがってノード番号の表現に必要な桁数が増えていくことになるが、この桁数を可変長にすると処理の上で困難が生じるため推論の深さを制限する必要がある。その制限と十分な推論の深さとのトレードオフはアプリケーション依存なため一意に定めることはできない。この方法での問題としては、複数の異なるノードが同じ推論の中間結果を持つ場合があり、その中間結果を参照したエージェントが冗長な推論をおこない、最悪の場合システム全体が同一の推論過程を繰り返す可能性があるということである。

木の操作に関しては、エージェントは以下のことをおこなうことができる。

#### Add\_Node:

現在のエージェントの中間結果を木にノードとして加える。これには三つの場合がある。

「提案」は、中間結果を評価値とともに、木に加える。

「質問」は、推論しても評価値が定まらない場合で、評価値が空欄になっている状態でノードを木に加える。

「応答」は、中間結果を評価値とともに、木に加える。「提案」と異なる点は、受け取り相手が指定されていることである。

#### Read\_Node:

他のエージェントの中間結果を受け取る。受け取り後に、その中間結果の続きを推論するか、あるいは自分自身の推論を継続するかは、エージェントに依存する。ただし、受け取ったものが「質問」だった場合は、それを優先して推論することになる。

### 4. おわりに

現在のシステムではプロトコルマネージャが協調プロトコル全体の集中管理をおこなっているためこの部分の分散化を予定している。また同一の中間結果を持つノードの管理についての問題の解決が必要である。

#### 参考文献

- 1) Durfee, Lesser, Corkil: "Trends in Cooperative Distributed Problem Solving", IEEE transaction on knowledge and data engineering, Vol1, No1, March 1989