

サーキュレーションプロトコルによる 分散協調エージェントシステム

新井 敏正* 中沢 実* 服部 進実* 山村 尚一**

*金沢工業大学工学部情報工学科

**株式会社PFU

ワークステーションとネットワークからなる分散処理環境において、複数の自律エージェントが競争・協調場を生成し、利益極大化を図るため、他のエージェントの状況を把握すべく、機能単位の回覧板を用いたサーキュレーションプロトコルと、それを用いて問題解決を行うためのグループ型対話状態遷移図を提案している。本方式では、主に問題の動的な割当、2次的な問題の波及に対する制御を効率的に行うことを実現している。さらに実装例において、複数のメンバー間でスケジュールの自動合成をおこない、効果的な分散協調問題解決の実現について述べている。

Distributed Cooperative Agent System
by the Circulation Protocol

Toshimasa ARAI* Minoru NAKAZAWA * Shimmi HATTORI* Shoichi YAMAMURA**

* Faculty of Engineering, Kanazawa Institute of Technology
7-1 Ohgigaoka Nonoichi Kanazawa-south Ishikawa 921

** PFU Limited,
Unoke-cho Kahoku-gun Ishikawa 929-11

In distributed processing environment with workstations and networks, the circulation protocol with function oriented circulation board and group conversation transition diagram with a language/action model are proposed to solve the problem and to get the maximum benefits, creating competition and coordination field of intelligent agents. In an example of the implementation model, automatic scheduling of office workers is applied, realizing dynamic allocation of problem and efficient control of secondary diffused problems.

1. まえがき

近年の計算機環境において、ワークステーションやパソコンの性能向上、小型化、低価格化などが進み、LANやISDNなどのネットワークを用いたソフトウェアツールなどが続々と発表されてきている。それと同じくして、通信網の高度化を目的として、インテリジェント・ネットワーク (IN) や電気通信管理網 (TMN) の検討が盛んに行われている。⁽¹⁾

一方、計算機による問題解決においては、プロダクションシステム、論理システム、フレームシステム、オブジェクト指向システムなどの知識処理技術が現実のものとして開発されつつある。

しかし上記の通信ネットワークに知識処理を使用したシステムに関し実用的システムについてはまだ模索段階が大部分と思われる。

そこで、このような通信ネットワークにおいて問題解決を取り扱う枠組みとして、分散協調エージェントシステムが目ざされている。⁽²⁾

ここでの分散協調のコンセプトに関しては、次のような研究が行われているが、課題も残されている。

現状の動向

ハードウェアの自律分散化は、その制御ソフトの自律分散化を要請する。全体を管理し、制御するマスターは存在せず、各ノードはトークンと呼ばれるメッセージが伝播し、ある特定されたネットワーク情報を獲得できる。トポロジーやパラメータが時間的に変動しつつあるネットワークにおける最短経路計算を行う分散アルゴリズムが研究されている。また、天災地域で障害や緊急停止を受けたネットワークが運転回復を行うとき、自動的にマスターを定め、所定の回復手順に従ってネットワークを正常運転へ立ち上げる問題もよく研究されている。

課題

LANやISDNの端末は、それがワークステーションであったとしてもユーザの意図を反映させなければならない。1つのユーザ集団はLANやISDN上に競争・協調場を形成しグループ活動を行い、各ユーザは自律的、非リアルタイム的、分散的環境で競争と協調を繰り返しながら利益極大化を図っていくと

いった研究はまだ模索中と考えられる。

本論文では、このような上記の課題を解くために、機能ごとに回覧板を用いて分散環境における他のエージェントの状況を把握するためのサーキュレーションプロトコルと、問題解決を行う回覧板でのグループ型対話状態遷移図を提案している。実装例としてオフィスにおける業務の効率化や複数ユーザ間のグループワーク支援を行うため分散計算機環境において、図1のように回覧板を通信における媒体とする非リアルタイム型のメッセージ通信系と、業務処理そのものに対応するデスクワーク系とを有機的に結合したパーソナルな秘書代行システムを一つのエージェントとし、それらが、グループとして回覧板形式のサーキュレーションプロトコルを用いてお互いに協調しあいながらスケジューリングに関する問題を解いていく分散協調エージェントシステムの構築を行った。

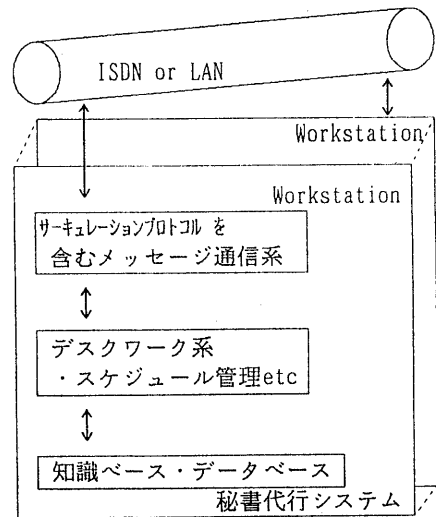


図1. 分散協調エージェントシステム

以下、2章においては、分散協調問題解決の現在の動向とその問題点、要求について考察する。3章では、サーキュレーションプロトコルによる分散協調問題解決CPDAI (Circulation Protocol for DAI) を提案する。また、4

章では、実際に分散環境においてCPDAIで複数人の競合するスケジューリング問題を解くことにより、提案したCPDAIの有効性を示し、その評価を述べる。

2. 分散協調問題解決

分散協調問題解決は、⁽³⁾ 協調して問題を解決するために、インテリジェントな自律エージェントがグループの相互作用を取り扱う人工知能のサブフィールドである。分散協調問題解決は人工知能技術と分散処理技術の両方に関与し、分散処理の一般的な領域とは異なる。なぜなら、分散協調問題解決はデータとそのデータに関する手続きが係わるため、データに対する協調ではなく、データ+手続きの「プロセス」に対する協調を含むからである。一方、分散処理は異なる集合でほとんど独立したタスクを完成するために計算エージェントのネットワークを調和させるものである。

分散協調問題解決のゴールは、オフィスにおける会議などに参加する人々のスケジューリングの調整のような、一つの仕事を解くために、何人かで行うような協調エージェントのグループを作り出すことにあると考える。

もし問題を独立する部分に分けることが出来るならば、分配することによって、より素早く解決法を見出すことが出来る。しかし、本論文では、ある問題を解くのにお互いの状況が競合する場合の解決法や、またそのときにエージェント同士がどれだけ協調しあえるかを問題とする。つまり、我々は問題を完全に分割することはほとんどの場合、不可能と考えている。そしてもし可能であったとしても、複数に独立している問題の解を統合することは難しい。例としてはある一つのサブ問題において選択もしくは委託が他のサブ問題の内容を制約するとき起こる。

最後に、分散協調マルチエージェントシステムはどのようにして一つのシステムが、他のシステムの状況についてまた自分自身について推論するかということが問題である。以上をまとめると次のような要求があると言える。

- (1) 動的な問題割当て
- (2) 知識の分割に応じた問題割当て
- (3) 問題のタイプによる実行制御

- (4) 大局的な問題解決制御
- (5) 問題の2次的な波及に対する制御
- (6) 効率的な問題解決

従来の回覧板プロトコル⁽⁴⁾においては、上記の要求での(1)、(2)、(3)、(4)を満たすような解決方法についてある程度述べられているが、分散協調問題に関しては、「あるサブ問題の生成、変更、更新が他のサブ問題への変更、更新を新たに発生させ、直接関係する問題以外に波及する複合ネットワークを構成することが考えられる。」⁽⁵⁾と言った場合の2次的の波及問題に対する制御方法に関する点や、全エージェントに対して回覧板が巡回するのでエージェントの数の増大に応じてメッセージ巡りに時間がかかるという効率性に関する点が考慮されていない。

以上の要求から、次章では(1)～(6)の機能を実現する問題割当て方式を提案する。

3. サーキュレーションプロトコルによる分散協調問題解決

本論文におけるサーキュレーションプロトコルは、Winograd T. の対話状態遷移図の拡張モデル(グループ型対話状態遷移図)を考案し、それに基づいて各機能ごとに複数の回覧板を用いて、問題解決を行っていくことが最大の特徴である。ここでは、各エージェントの構成、サーキュレーションプロトコルの機能、そのデータフォーマット、多人数を想定した問題解決のためのグループ型対話状態遷移図について述べる。

3.1 エージェントの構成

各エージェント内の構成は、図2のように、大きく分けて2つの部分からなっている。

●メッセージ通信系

メッセージ通信系の最下位層には、UNIXにおいて、標準装備で着信保証機能を持っているTCP/IP等の通信プリミティブを用いての1対1の基本的通信機能を有する。その上に、グループ間通信を行うための回覧板を意識した機能的サーキュレーションプロトコルを配備している。

●問題処理系

問題処理系においては、機能的サーキュレーションプロトコルからの他のエージェントの要

求条件を含む環境状況の情報と、知識ベースからの情報から次なる行動を決定するための回覧板を想定した拡張対話状態遷移図、プロダクションルールとその推論機能が格納され対話状態遷移図やアプリケーションに情報を提供する知識ベース部がある。

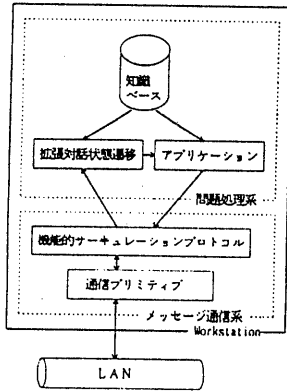


図2. 各エージェントの構成

3.2 サーキュレーションプロトコル

基本的な通信プロトコルである1対1のメッセージ通信の上に、グループ間通信を行うために機能的サーキュレーションプロトコルを考案した。このプロトコルは、図3に示す様に、各エージェント間を通信媒体である回覧板が一つの問題に対し、複数枚回覧する仕組みになっている。この場合、問題の状況により回覧されるエージェントは回覧板単位に異なることがあることを示している。

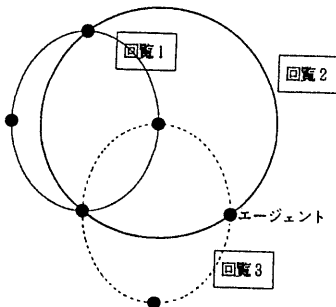


図3. 回覧板順序の制御

回覧板の単位は機能によって分担作成され、そ

のメッセージの意味や内容に沿うように回覧順序を作成して回覧させるものである。例えば、問題の割当を行う際には関連する全てのエージェントに問題の割当メッセージを回覧させる必要があり、その通知メッセージと共に、全エージェントへの回覧順序を作成し回覧させるものである。また、その回覧結果によって、割当可能なエージェントが決定したならばそのエージェントにのみ問題を割り当てるように回覧順序を組んで回覧板に載せて回す。この時に問題の解決がより容易になる回覧順序をその時の問題提起エージェントによって作成する事が出来る。回覧板に載せる情報は図4のような内容になっている。回覧板に対する手続きは、問題提起のエージェントが、「問題内容と終了条件」、「回覧の順序」を提起し、各エージェントが、自分の意思を「各エージェントの処理結果」に入れる。従って、回覧を受けたエージェントは、自分以外のエージェントの処理状況すなわち環境情報を把握することができる。

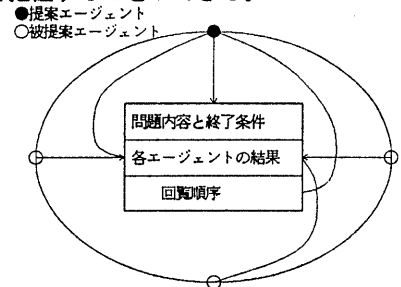


図4. 回覧板の制御

3.3 問題解決プロトコル

分散協調問題を解くことは、適度の競合と協調が大切である。特に協調して問題を解く場合は、全体の目的を達成するために、多少は優先度の低いエージェントが犠牲にならざるを得ない。ここで我々は、こういう分散協調問題を解くために、Winograd T. の対話状態遷移図⁽⁸⁾の問題提案から問題受理までの過程を、多人数によっても可能になるような提案に対する同意度による重み付き拡張対話状態遷移図（グループ型対話状態遷移図）を提案する。（図5）この状態遷移図は回覧板の各エージェントの結果の状態によって行動を決定していく過程を示しており各エージェントがこの回覧板に関する

状態遷移図のアーチの部分を生産ションルの形で保有している。

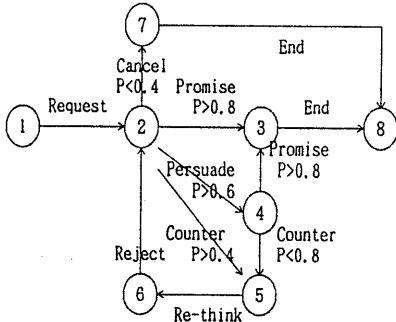


図 5. グループ型対話状態遷移図

それぞれのノードについてその番号に対応させて説明すると、以下ようになる。

- ①問題提案一回覧板 (Request)

問題提案エージェントが被提案エージェント全員に対して、回覧板を用いて、問題の提案を行う。
- ②被提案者回答 (select)

被提案エージェント (複数) の回答を元に、問題の同意度 (P) を求め、それぞれの値に従って、次の行動に移る。
- ③問題同意一回覧板 (Promise)

②の結果で、ほぼ同意がとれたので契約を行う意味で、同意をえたエージェントに対して、回覧板を回して、確認をとる。
- ④説得一回覧板 (Persuade)

②の結果で、同意がえられなかったエージェントに対して、更に上位エージェント (エージェント間において重要度が高い) の意見をいれて、説得を行う。
- ⑤再提案 (Counter)

②の結果で、問題の提案に対してあまり同意度が高くないので、②において同意していなかったエージェントに対して最も重要度が高いエージェントの提案によって、再び回覧順序と問題提起がされる。
- ⑥受入拒否一回覧板 (Reject)

⑤に続いて、それまでに同意を得ていたエージェントに対して問題の取消を行うことを求める回覧板、このあとに、②にいて再び提案を行う。

⑦問題取消一回覧板 (Cancel)

②の結果で、問題の提案に対してほとんど同意がとられず、再提案を行っても解決の見通しが立ちそうにないので、問題の取消を行う。

⑧問題終了一回覧板 (End)

問題の割当の終了及び確認事項を各エージェントに示す回覧板

このグループ型対話状態遷移図に示されている P は、問題におけるその時点での協調の評価値である。評価値の設定は、後に実装例で示すが問題によって設定を行うものである。評価値 P と行動との対応は、現状では主観的なもので、問題の状況との関係で定量化することが今後の課題である。また、この様に複数の回覧板プロトコルを用いて問題解決を行うことによって、組織階層構造を用いた意思伝達、説得などや、動的な問題 (一度決定した解の変更などの割り込み等) に対しても、次の 2 種類の方式の回覧板を用いて、並列的に機能させることができる。(図 6)

●水平型回覧板

階層の同一レベル間のメッセージ通信を行う。

●垂直型回覧板

異なるレベル間のメッセージ通信を行う。

前者は問題の提起を行うときに使用されたりする。後者は主に下位エージェントへの説得や割り込みを行うためのもので職位、重要度に関する知識が不可欠である。

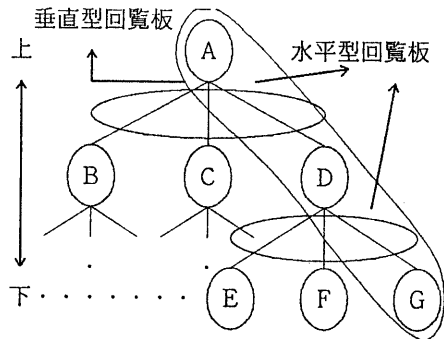


図 6. 階層構造における回覧板

4. 実装例（スケジュール）

以上に述べたサーキュレーションプロトコルによる分散協調マルチエージェントシステムを、オブジェクト指向言語Smalltalk-80を用いて、複数台のUNIXワークステーションPFU-A Station230上で試作した。実装例は、スケジュール自動管理システムで、着信メールで通知される、場所・日時などの情報に基づいて、行動計画のスケジューリングを行う。問題となるのは、通知された内容が、スケジューリングの上で衝突した場合の対処法であり、その対処法としては、秘書が行っているボスの行動計画のスケジューリングそのものを機械化する。デスクワーカーの職位・行動習性・優先度選択基準などを知識ベース化し、AI技術を適用することにより、自動化を行うことを目的とする。このためには分散処理環境において、複数のワークステーションがスケジュール合成を行うために、ワークステーション相互が協調しあひながらスケジュールを決定するシステムについて述べる。⁽⁵⁾

4.1 データベース/知識ベースの構成

分散協調問題解決においては、データベースおよび知識ベースの存在は、入力に対するそのエージェントにとっての意思に当たり、重要なファクターである。スケジュール自動合成においてはスケジュールの衝突時の優先順位付けを行うものである。優先度情報の種類としては、

- ①個人重要度データ
- ②スケジュール重要度データ
- ③場所と場所との所要時間データ
- ④職位重要度データ
- ⑤人物と職位の関係データ
- ⑥個人重要度、スケジュール名重要度、職位重要度—スケジュール重要度生成ルール
- ⑦提案スケジュール重要度、既決スケジュール重要度—中間仮説生成ルール
- ⑧既存スケジュールの決定からの経過日数、中間仮説—行動決定生成ルール

なお、上記のデータおよびルールは、個別データと共通データに分かれる。

●個別データ

各ユーザの知識や職位に基づく行動習性を表すものであり、各ユーザ個別のもので各ワークステーションにより内容が異なる。上記で言え

ば、①、②、③が、個別データにあたる。

●共通データ

すべてのユーザにとって共通のものであり、企業組織構造や人事データなどのデータベース的なものである。上記で言えば、④、⑤が、共通データにあたる。

●知識ベース

上記のデータから、優先度情報を定量化するルールなどを格納する。共通データと同じように、全てのユーザにとって共通のものである。上記で言えば、⑥、⑦、⑧が、知識ベースにあたる。

4.2 スケジュール管理システムの動作

スケジュール管理システムの実際の動作は次の3つの処理を並行に行っている。

(1) サーキュレーション通信系

他のエージェントからの回覧板を受信し、そのメッセージを保管したり、他のエージェントに、回覧板として情報を送信する。

(2) スケジュール合成処理

(1)の処理で保管したメッセージを取り出して、スケジュールを承認、変更、削除などを知識ベースを用いて、スケジュールを自動的に決定する権限をもつ。また、結果を(1)に送ることを行う。

(3) スケジュール管理インタフェース

ユーザが実際にスケジュールを登録、変更、削除などを行う部分。(2)を起動したり、(1)に、回覧板に載せる情報を転送したりする。

3つの処理間の関係は、図7の様になっている。

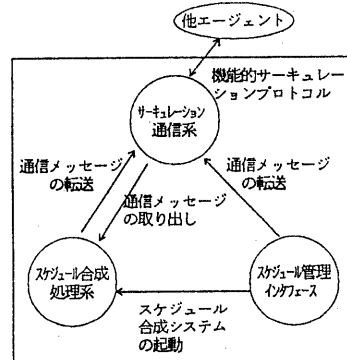


図7. スケジュール自動管理システム

4.3 スケジュール管理システムの機能

スケジュール管理システムの機能は、機能的回覧板通信を使うものをあげると、次のようなものがある。

●共通スケジュールの提案一回覧板メソッド

```
(Agent new) proposeSchedule: aName
                place: aPlace
                myOption: anArray1
                member: anArray2
aName          ・スケジュール名
aPlace         ・スケジュール開催場所
anArray1       ・スケジュールの提案者名と
                スケジュール開催日時
anArray2       ・参加メンバー名
```

複数の参加メンバーによる、会議などの共通のスケジュールの提案である。この回覧板を受け取ったメンバーは、スケジュール合成処理を起動して、自分の意見を回覧板に付加し、次のメンバーに送信する。

●共通スケジュールの削除一回覧板メソッド

```
(Agent new)
                proposeRemovedSchedule: aSchedule
aSchedule      として既に登録してある共通のスケジュールを回覧板を用いて削除する。
```

●空時間収集一回覧板メソッド

```
(Agent new)
                getFreeTimeY: aYear    (年)
                month:      aMonth    (月)
                date:       aDate     (日)
                place:      aPlace    (場所)
                member:     aMember   (メンバー)
```

aMember において、この日(年月日)に、この場所(aPlace)に行き来ができるほどの時間を見つけろといった回覧板を作る。スムーズにスケジュールの提案をおこないたいときに便利である。

●再提案準備一回覧板メソッド

```
(Agent new)
                proposeScheduleAgain
```

この機能は共通スケジュールの提案により他の提案者より発生したスケジュールの再提案を、

そのスケジュールの提案者に要請する回覧板を作る機能である。この回覧板は提案者に送るだけでなく、再提案となったスケジュールの参加メンバーにも送られそのスケジュールの削除を行う。

●参加説得一回覧板メソッド

```
(Agent new)
                proposePersuadeSchedule: aSchedule
                alterProposer: anAgent
                proposeMember: aMember1
                member: aMember2
aSchedule      ・スケジュール名と開催日時
anAgent        ・不参加者に対して説得を行う人
aMember1       ・参加しないメンバー
aMember2       ・参加メンバー
```

この回覧板は共通スケジュールの提案においてそのスケジュールに対して同意を示さないメンバーがいるときに使用する。スケジュールに参加するメンバーにおいて一番スケジュールの重要度が高い参加者と、同意しなかったメンバーにおいてのみ再び回覧板を回し同意を得ることを目的とする。

これらの機能ごとに生成される回覧板を前述したグループ型対話状態遷移図の下において実行される。この実装例での協調の評価値は、次の評価関数のもとで行っていく。

R = 同意した被提案者の職位・人物重要度
 A = 被提案者全員の職位・人物重要度
 $P = R / A$ (評価関数)

また問題実行において、一度決まったスケジュールが優先度順位付けアルゴリズムによってあるユーザが放棄になる場合の2次的な波及に関する場合も、その時点で回覧板をまわし、同様な拡張対話状態遷移図によって解決できることも確かめられた。

4.4 実験

複数マネージャによるスケジュール自動合成でのシミュレーションを行い、サーキュレーションプロトコルにおける複数回覧板の効果を調べた。問題は、5人の場合のスケジュールリングについて、サーキュレーションプロトコル+グループ型対話状態遷移図を使用する本論文の場合と、一斉同報+グループ型対話状態遷移図のみを使用する場合⁽⁵⁾との両方を、同じ環境に

において行った。その場合のスケジュールが決定する過程を図8、図9に示す。なお星印の数が多ほどエージェントの重要度が高いことを意味する。この図よりサーキュレーションプロトコルを使用することによって、各エージェントの状況が回覧板という一つの記憶領域に環境情報として、集めることができ、エージェント間の一対一の基本通信回数が削減できることが明らかである。しかし欠点としては、スケジュールが決定するまでの時間がかかることである。

5. むすび

本論文では、複数のワークステーションでの分散協調型マルチエージェントシステムについて、他のエージェントの状況を把握するための機能的サーキュレーションプロトコルに、グループ型対話状態遷移図を使用して組織階層構造における意思伝達、説得などを用いた分散協調問題解決を行うシステムについて提案した。更に実装例においては、分散された複数のメンバーによるスケジュール問題で、各個人の行動様式や重要度に関する知識ベースとサーキュレーションプロトコルにより機能分散問題解決を行うシステムを試作した。その結果、問題の2次的な波及に関する制御や、効果的な分散問題解決について達成することができた。

今後、更に本システムを用いたより具体的な事例への適用、より汎用的な問題解決が、検討課題である。

参考文献

1. 第2回 NAワークショップ予稿集(1991)
2. 特集：自律分散システム、計測と制御、29-10 (1990)
3. Keith S. Decker.: "Distributed Problem Solving Techniques: a survey. IEEE Transactions on Systems, Man, and Cybernetics, SMC-17(5):722-740, September/October (1987)
4. 北村、小川、北橋：“分散協調問題解決における問題割当てのための一通信方式”、電子情報通信学会論文誌D, J71-D-2 No.2, 439-447, (1988)
5. 服部、蔵本、中沢：“分散協調環境における知的マルチメディア通信システム”、電

子情報通信学会論文誌B, J74-B-1 No.11, 899-908, (1991)

6. Winograd T.: "A language/Action Perspective on the Design of Cooperative work" Computer-Supported Cooperative Work: A Book of Readings, pp.311-331, Kaufmann Publishers(1988)

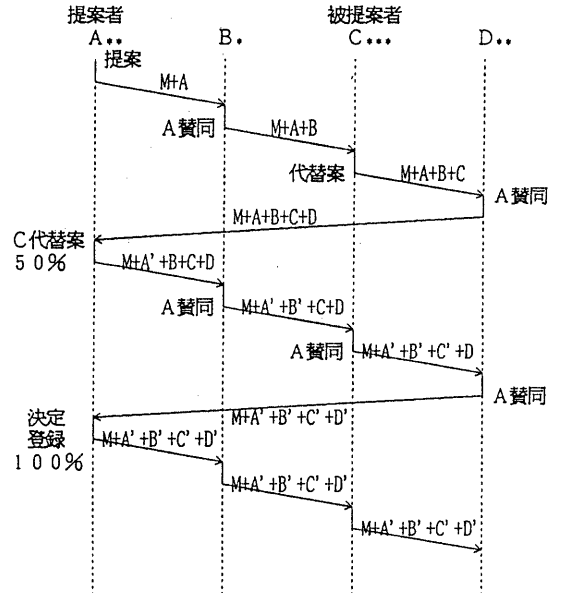


図8. サーキュレーションプロトコル+グループ型対話状態遷移図

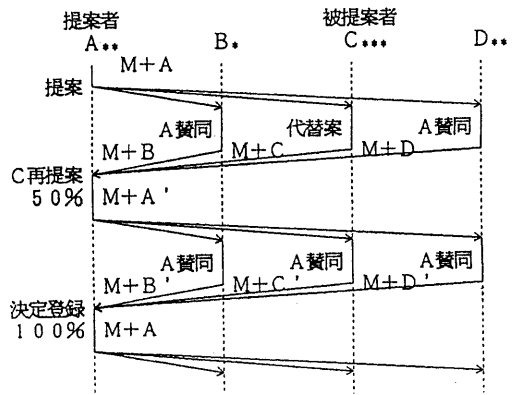


図9. 一斉同報+グループ型対話状態遷移図