

エージェントネットワーク Socia の AgenTalk による実装

4P-3

梶原史雄[†] 桑原和宏[‡] 石田亨[†]

[†]京都大学工学部情報工学教室 [‡]NTT コミュニケーション科学研究所

1 はじめに

AgenTalk^[1] はネットワーク上の分散計算機環境において、マルチエージェントシステムを容易に実現するプロトコル記述システムである。具体的には計算機ネットワーク利用者を支援するエージェント群を想定し、その間の通信および協調的動作を記述することを目的としている。我々は AgenTalk を用いて、エージェントネットワーク Socia を実現し、ワークステーションを用いた電子会合に適用した。会合を支援するエージェントは以下の3種のプロセスからなる。

- (1) 顔の探索・追跡を行なうプロセス。
- (2) 無予約型の会合スケジューリングを行なうプロセス。
- (3) 映像、音声を用いたマルチメディア会議ツールの起動終了を行なうプロセス。

これらのプロセス群を有機的に結合してエージェントを構成し、さらにエージェント間のスケジュール調整を実現するために、いかに AgenTalk を用いたかを述べる。

2 AgenTalk

AgenTalk では、各エージェントは以下の手順により、受信したメッセージに対応するメッセージ処理手続き（以下、メッセージハンドラと呼ぶ）を起動する。

- (1) まずエージェントの名標は Post-Office に登録されており、メッセージを送信する際には相手方エージェントの名標を指定する。Post-Office は名標と IP アドレスとの対応付けを行ない、メッセージを配信する。
- (2) メッセージは TCP 結合を通じて、プロセスごとに存在する AgenTalk Library と Post-Office との間で送受信される（図1参照）。この図に示すように、AgenTalk では必ずしもエージェントとプロセスは対応せず、1プロセス内に複数のスレッドを生成し、各スレッドをエージェントと対応させることも可能である。

(3) AgenTalk Library の pattern matcher は、受信したメッセージがどのエージェントのどのメッセージボタンに一致するかを判断し、そのメッセージハンドラを起動する。AgenTalk の特徴は以下の通りである。

- メッセージの記述にクラス継承が使用できるので、既存のメッセージを利用して新たなメッセージを容易に作り出せる。
- メッセージボタンを介してメッセージハンドラを起動するので、自由度の高い記述ができる。

AgenTalk の主要な機能は現在、Allegro Common Lisp を用いて実現されており、SUN, SGI のワークステーション上で動作している。また簡易な機能は C で記述したプログラムでも利用可能である。

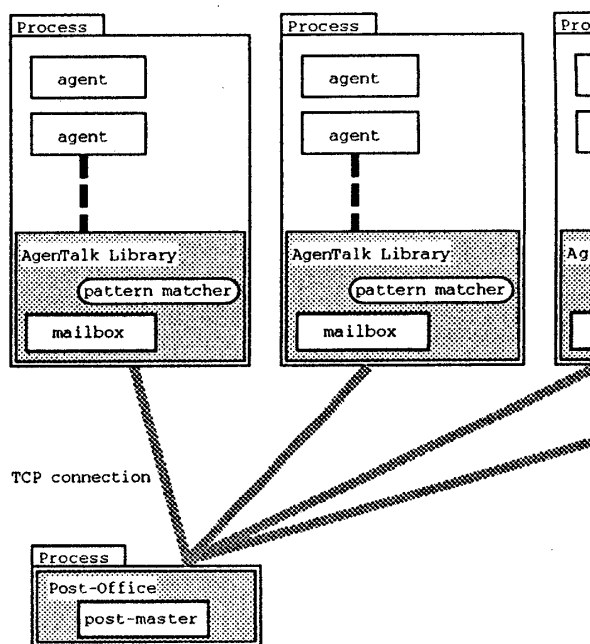


図1: AgenTalk の構成

3 電子会合支援への適用

3.1 システム構成

Socia による電子会合支援は、図2のように利用者各々を支援するエージェント（以下、会合エージェントと呼ぶ）

Implementing Socia Agent Network with AgenTalk

Masao Kajihara[†] Kazuhiro Kuwabara[‡] Toru Ishida[†]

[†]Department of Information Science, Kyoto University

[‡]NTT Communication Science Laboratories

とマルチメディア会議ツールから構成される。図2に電子
 会合支援システムの論理構成を示す。現在のところ、マル
 チメディア会議ツールはLBL, XeroxPARCが開発した音
 声会議ツール vat と映像会議ツール nv を用いている。

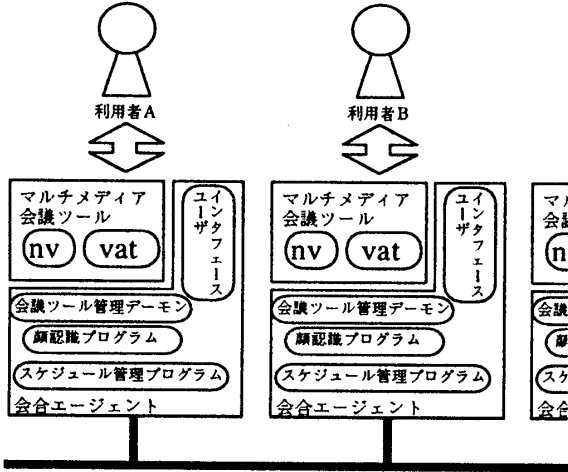


図2: 電子会合支援システムの論理構成

会合エージェントは以下の3種のプログラムで構成さ
 れている。

- (1) スケジュール管理プログラム：利用者相互間の交渉を
 行ない、会合のスケジュールを実時間で決定する。
- (2) 会議ツール管理デモン：各々のワークステーション
 内で nv, vat の起動・停止を管理する。
- (3) 顔認識プログラム：利用者の顔の探索・追跡を行ない、
 利用者の挙動を把握する。

物理的には図3に示すようにワークステーション4台を
 用いたシステムを実装中である。ここでは会議ツール管理
 デモン、顔認識プログラムをCで、スケジュール管理プ
 ログラムをLispで記述している。ユーザインタフェースは
 スケジュール管理プログラムに組み込んでいる。

論理的なエージェントが物理的には複数のプロセスとス
 レッドで構成されていることに注目されたい。この様子は
 AgenTalk を利用することにより、多様な物理構成を持つ
 エージェントを自在に実現できることを示している。

3.2 動作手順

図3に示したシステムでは、会合は以下のような手順で
 進行する。

- (1) ワークステーション3内にあるスケジュール管理プロ
 グラム同士が会合に必要な情報を AgenTalk を介して通信

し合う。

- (2) ワークステーション4の顔認識プログラムがスケ
 ジュール管理プログラムの状況判断を助けるため、
 AgenTalk を介して利用者の挙動を報告する。

- (3) 会合が実現できるようであれば、スケジュール管理
 プログラムがワークステーション1, 2, 4に存在する会議
 ツール管理デモンに、相手方のIPアドレスを含む情報
 を AgenTalk を介して送信する。

- (4) 各ワークステーションの会議ツール管理デモンは、
 受信した情報を基に nv, vat をワークステーション間で
 接続させる。

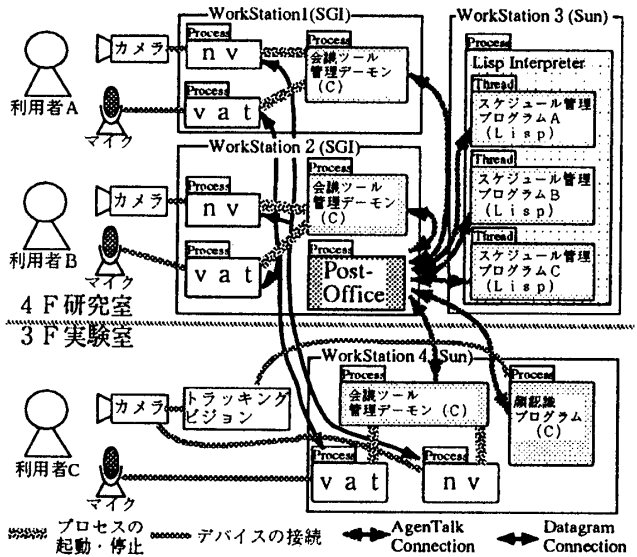


図3: 電子会合支援システムの物理構成

4 おわりに

今回の実装では AgenTalk の柔軟な通信機能を利用し
 て、会合エージェントを複数のプロセスで構成すると共に、
 エージェント間でのスケジュール調整を実現した。現時
 点では、AgenTalk はメッセージ定義機能の提供はしてい
 るものの、エージェント間プロトコルを容易に定義する機
 能は備えていない。現在 AgenTalk の次期バージョンとし
 て、協調プロトコルを階層的に記述するための、拡張有限
 状態機械を基本としたスクリプト言語を開発中である。

参考文献

[1] 桑原和宏, 石田亨, 大里延康, "AgenTalk: マルチエー
 ジェントシステムにおける協調プロトコル記述," 電子通信情報学会
 AI 94-56, pp.1-8, 1995.