

モバイルエージェントに基づく会議日程調整システムに関する研究

A Mobile Agent-based Meeting Arrangement System

浜田 裕介[†] 本村 真一[†] 川村 尚生^{††} 菅原 一孔^{††}

Yusuke HAMADA[†] Shinichi MOTOMURA[†] Takao KAWAMURA^{††} Kazunori SUGAHARA^{††}

[†] 鳥取大学 大学院 工学研究科 ^{††} 鳥取大学 工学部

1 概要

近年、コンピュータネットワークの発達に伴い、グループでの共同作業を効率的に進めるためのグループウェアが注目されている。グループウェアの機能の1つとしてスケジュールの調整が挙げられる。既存のグループウェアでのスケジュール調整は、会議の参加者全員のスケジュールが共有され、あらかじめ正しく入力されていることが前提となっている。また、全員に都合の良い日時がなかった場合に、交渉を行う機能を有するグループウェアは存在していない。

そこで、我々はモバイルエージェント技術に基づき、事前の予定入力が不要で、日程調整において必要であればモバイルエージェントが参加者と交渉する会議日程調整システム [1] を開発している。しかし、本システムではシステムに参加する全てのノードがグローバルIPアドレスを持つコンピュータ(以下グローバルノードと呼ぶ)であるか、NAT(Network Address Translation)ルータの内側にあるコンピュータ(以下ローカルノードと呼ぶ)である必要があった。なぜなら、グローバルノードから、ローカルノードへエージェントが移動できないという問題があったからである。

そこで、本研究では新たなエージェントの移動方式を実装を行い、この問題を解決し、システムに参加するユーザノードがグローバルノードでも、ローカルノードでも参加できるように改良した。

2 会議日程調整システム

2.1 会議日程調整システム概要

本システムは、我々の研究室で開発したモバイルエージェントフレームワークである Maglog[2] を用いて実装している。本システムで会議を招集するには会議情報の決定、すなわち会議の参加者、開催目的、開催日の許容範囲、所要時間の決定を行う必要がある。

会議日程調整が行われる際の利用者の操作と提示される情報を次に示す。会議の招集者と被招集者の操作は会議情報の決定以外は同じである。

1. 会議の情報、すなわち会議の開催目的、開催日の許容範囲、所要時間が提示される
2. 提示された会議開催日の許容範囲の予定を入力する
3. 入力した予定の一部を変更できないかの依頼が提示される
4. 予定の変更を承諾、もしくは拒否する。

5. 会議の開催日時あるいは会議の開催に失敗したことが提示される。

上記の3は交渉が必要な場合に提示され、場合によっては複数回提示されることもある。上記3が提示されない場合は、当然上記4も行われない。

2.2 会議日程調整システム構成

本システムの概要を図1に示す。また、本システムの構成は以下の通りである。

- 日程調整を行う日程調整に関するエージェント
- 日程調整に関する情報を管理する日程調整サーバ
- ユーザとのやりとりをするユーザインタフェース

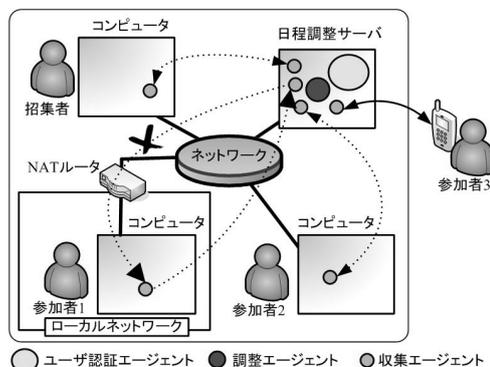


図 1: 会議日程調整システム概要図

日程調整に関するエージェントは収集エージェント、調整エージェント、結果通知エージェントの3つで構成されており、それぞれ予定の収集、予定の調整と交渉、結果通知を行うエージェントである。そして、これらのエージェントが複数ホスト間を移動しつつ、連携し予定収集や交渉を行う。

日程調整サーバは1つだけ存在し、ユーザのログイン情報、ログイン処理、ログイン済みのユーザのノードのIPの管理を行うユーザ認証エージェントが動作している。また、日程調整サーバは日程調整に関するエージェントの移動の起点となっており、日程調整サーバ上で日程調整に関するエージェントが日程調整に必要な情報の交換を行う。

2.3 日程調整に関するエージェントの動作

本システムでは、会議を招集しようとする時、調整エージェントが招集者のノード上で生成される。生成

された調整エージェントは日程調整サーバに移動し、収集エージェントを参加者ごとに生成する。生成された収集エージェントはそれぞれの担当する日程調整サーバ上の情報を参照し、参加者のノードに移動し、予定を収集する。予定の収集が終わると、収集エージェントは日程調整サーバに戻り、収集した予定を調整エージェントに渡す。調整エージェントは、全ての予定を取得すると、参加者全員に都合の良い日時を探す。もし、全員に都合の良い日時がない場合、調整エージェントは適切な日時において任意の参加者と交渉を行う。適切な日時は、交渉が必要な時間数、交渉が必要な参加者数が少なくなるように選ぶ。招集者が開催を希望している時間も考慮する。日程調整の結果が決まると、調整エージェントは、参加者ごとに、結果通知エージェントを生成する。生成された結果通知エージェントは、担当する参加者に日程調整の結果を通知する。

もし、システムにログインしていない参加者がいた場合は、日程調整に関するエージェントは日程調整サーバ上で一定時間待機し、それでもシステムへログインしてこない場合、参加者にメールを送信する。メールには URL が記載されており、参加者は URL にアクセスすることにより、日程調整サーバを通じて外部からのシステムへの参加が可能となる。

しかし、それでもなお参加者が日程調整に一定時間の内に参加しなかった場合、収集エージェントが予定の収集を行っている場合は招集者に参加者を欠席にするか、日程調整を中止するかを聞きに行くという作業を行う。この際、参加者がシステムにログインしていない場合の動作は、前述した通りに行い、それでも招集者が一定時間の内に参加しなかった場合は、日程調整を中止し、結果の通知を行う。調整エージェントが予定の交渉を行っている場合は、日程調整を中止し、結果の通知を行う。結果通知エージェントの場合は、この様な場合でも無期限で待ちつづける。

2.4 実行例

実際にユーザが予定を入力している例を図 2 に示す。ユーザは GUI を用いて簡単に予定を入力することができる。PREFERABLE は会議の開催を希望する時間を表し、招集者だけが入力できる。BUSY は会議に参加できない時間、TENTATIVE は交渉の余地のある時間、FREE は会議に参加できる時間を表す。BUSY と入力されている時間では会議は開催されず、必要であれば TENTATIVE と入力されている時間で交渉が行われる。

3 問題点

グローバルノードとローカルノードの通信についての概略を図 3 に示す。

一般に、ローカルノードからグローバルノードへ通信を開始しようとした場合、間にある NAT ルータがローカルノードからグローバルノードへ向けたパケットの送信元 IP アドレスを NAT ルータが持っているグローバル IP アドレスに書き換え、その際に、元の送

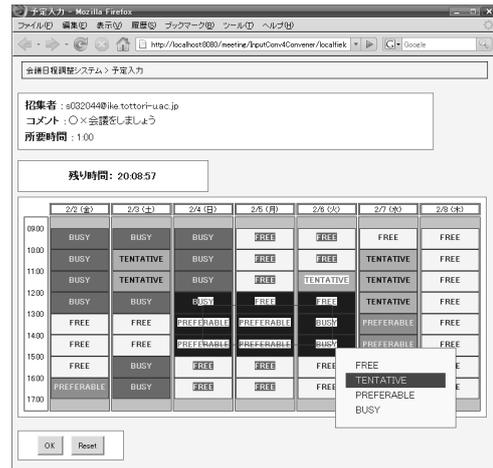
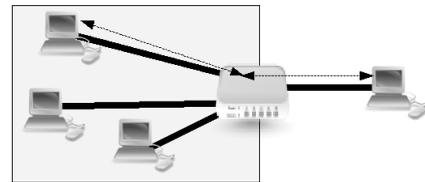


図 2: ユーザの予定入力例

ローカルノードからグローバルノードへの通信



グローバルノードからローカルノードへの通信



図 3: グローバルノードとローカルノードの通信

信元 IP アドレスと送信先 IP アドレスの対応を記憶する。そして、グローバルノードにパケットが届くと、グローバルノードはその応答をパケットの送信元 IP アドレス、つまり NAT ルータに向けて送信する。グローバルノードから送信された応答のパケットが NAT ルータに届くと、先ほど記憶しておいた IP アドレスの対応を参照して、送信先 IP アドレスをローカルノードの IP アドレスに書き換え、パケットをローカルノードに再送し、応答のパケットがローカルノードへ届く。そして、ローカルノードからグローバルノードへの通信が確立される。

逆に、グローバルノードからローカルノードへの通信を行う場合、ローカルノードが持っているプライベート IP アドレスではそもそも通信が行えない。このため、ローカルノードが持っているローカル IP アドレスではなくローカルノードとグローバルノードの間にある NAT ルータのグローバル IP アドレスを使用して通信を行う必要があるが、通信を開始しようとしても、NAT ルータには送信元 IP アドレスと送信先 IP アドレスの対応付け、つまり NAT ルータに届いたパケットをどのローカルノードに送ればいいかという情報が

記録されていないため、パケットが NAT ルータで止まってしまい、ローカルノードに届かないため、通信を開始することができない。

上記の理由により、ローカルノードからグローバルノードへの通信を開始し、グローバルノードからローカルノードへ応答を返すことはできるが、逆にグローバルノードからローカルノードへの通信を開始することは不可能である。本システムのエージェントの移動もノード間の通信によって行うため、同様の理由でローカルノードからグローバルノードへの移動は可能であるが、逆にグローバルノードからローカルノードへの移動は行えない。

4 新たなエージェントの移動方式

前述の問題を解決する方法として、まず、NAT ルータにパケットのルーティング情報を設定する、という方法があるが、いちいちルーティングの設定をあらかじめ行わなければならないため面倒である。この手間を省く方法として、UPnP(Universal Plug and Play)と呼ばれる機能を持った NAT ルータを使用する方法がある。しかし、この機能を利用するためには、NAT ルータとアプリケーションの両方が対応していなければならない、また、NAT ルータどうしが複数つながれている場合には対応できない。

そこで、以下の様な移動処理をある特定のグローバルノードに向けて一定周期で繰り返すことにより、グローバルノードからローカルノードへのエージェントの移動を実現した。移動処理の概略図を図 4 に示す。

1. ローカルノードからグローバルノードへ、移動したがついているエージェントがあれば受け取る、という要求を出す。
2. グローバルノードから、移動したがついているエージェントがあれば、応答を通じてエージェントが移動する。

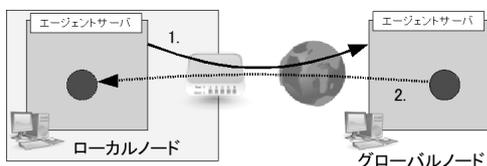


図 4: 新たなエージェントの移動方式の概略図

しかし、このエージェントの移動方式には 2 つの問題点がある。まず、移動処理の周期が短いとシステムの負荷、通信量が大きくなるため、移動処理の周期をある程度長くする必要がある。しかし、周期を長くしてしまうと、エージェントを即座に移動させる、といったことができなくなってしまう。この問題については、本システムではグローバルノードからローカルノードへ即座に移動する必要のあるエージェントが存在しないため問題にならない。次に、移動処理をどのノードに向けて実行するかを必ず指定する必要があるため、

不特定多数のノードと相互にエージェントを移動させられない、という問題がある。この問題については、本システムでは、全てのエージェントが他のノードへ移動する際に必ず日程調整サーバを経由して移動するため、移動処理の通信先ノードを日程調整サーバに設定すればよいと、問題にならない。

5 実装

新たなエージェントの移動方式を使用した会議日程調整システムを実現するため、本システムで使用しているエージェントフレームワーク Maglog 上に新たなエージェントの移動方式を実装した。加えて、会議日程調整システムの各エージェントもこれに対応させるため、必要に応じて従来通りのエージェントの移動方法と、今回実装したエージェントの移動方法を使い分けるように改良を行った。

6 実験

まず、192.168.1.0/24 のネットワークに属するノードを 2 台用意し、うち 1 台を日程調整サーバとした。次に 192.168.2.0/24 のネットワークに属するノードを 2 台用意した。そして NAT ルータを 1 台用意し、WAN(Wide Area Network) 側のアドレスが 192.168.1.0/24 に属し、LAN(Local Area Network) 側のアドレスが 192.168.2.0/24 のネットワークに属するように設定して、192.168.1.0/24 のネットワークに属するノード 2 台を NAT ルータの WAN 側ポートに、192.168.2.0/24 のネットワークに属するノード 2 台を NAT ルータの LAN 側ポートに接続した。この状態では 192.168.1.0/24 に属するノードから 192.168.2.0/24 に属するノードへ向けて通信を開始することはできない。この様な環境で改良後のシステムを動作させ、動作することを確認した。

7 おわりに

本研究では、我々が構築した会議日程調整システムにおいて、グローバルノードからローカルノードへエージェントを移動できないという問題を解決する一手法について述べ、これを実装した。これにより、我々が構築した会議日程調整システムを使用できる環境を増やすことができた。

今後の課題として、参加者の中に必ずしも参加しなくても良い人がいる場合に対応するという検討している。

参考文献

- [1] Yusuke Hamada, et al. Multi-agent-based approach for meeting scheduling. In *Proc. of IEEE Intl. Conf. on Integration of Knowledge Intensive Multi-Agent Systems*, pp. 79–84, 4 2007. Waltham, Massachusetts, USA.
- [2] Shinichi Motomura, Takao Kawamura, and Kazunori Sugahara. Logic-based mobile agent framework with a concept of “field”. *IPSJ Journal*, Vol. 47, No. 4, pp. 1230–1238, 4 2006.