

L-074

モバイルエージェントに基づく会議日程調整システム Meeting Arrangement System Based on Mobile Agent Technologies

本村 真一[†] 影本 憲五[†] 川村 尚生[‡] 菅原 一孔[‡]
Shinichi MOTOMURA Kengo KAGEMOTO Takao KAWAMURA Kazunori SUGAHARA

1. はじめに

本稿では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ついて述べる。通常我々が会議の日時を定める場合、会議の招集者は日程調整を次のような手順で行っている。

1. 会議の参加者、会議を開催する日時の調整期間を定める。
2. 会議の各参加者に対してスケジュールの都合を尋ねる。
3. 各参加者の都合を照らし合わせ、参加者全員の都合のよい日時を探す。
4. 都合のよい日時があれば、参加者全員にその日時を通知する。
5. 参加者全員に都合のよい日時がみつからなかった場合、各参加者に都合をつけてもらえないか個別に交渉する。誰と交渉するかは招集者の判断による。

会議日程を調整する手段としては、相手に直接会う、メールを用いる、電話をかけるなどがあげられる。上記2, 4はメールを用いることにより比較的簡単に行えるが、上記5の交渉するという手続きは、交渉がいつも成功するとは限らないし、誰と交渉すればよいかを考える必要があるため、上記のいずれの手段を用いても時間と労力を必要とする。

そこで、次の手段としてグループウェアを用いることが考えられる。グループウェアとは、コンピュータネットワークを利用し、グループでの共同作業を効率的に進めるためのソフトウェアであり、情報の共有やコミュニケーションの効率化を図り、グループによる協調作業を支援するために用いられる。既存のグループウェアとして、サイボウズ Office, IBM ロータス Notes/Domino, マイクロソフト Exchange Server などがある。これらのグループウェアを用いて会議日程調整を行う場合、会議の参加者全員がスケジュールを共有することが必要であり、全てのスケジュールを正しく入力していることが前提である。そのため一部でも予定の入力が不足している場合には日程を調整することができなくなる。またこれらのグループウェアには全員のスケジュールが空いている日を探す機能はあるが、全員に都合のよい日がない場合の日程調整機能はなく、調整のための交渉は人間が行わなければならない。

本研究ではモバイルエージェントを用いることで、会議日程調整作業のうち人の意志決定の必要のない上記2, 3, 4を自動化し、人の意志決定の必要のある上記5につ

いてもできる限り自動化を行なうことで、招集者が行なう会議日程調整作業の簡素化を目指す。

2. 会議日程調整システム

本システムを用いて会議を招集したり会議に参加するためには、事前にメールアドレスとパスワードを登録しておく必要があり、これらの情報を用いて利用者の識別を行なう。本システムの利用者はログイン手続きを行なうことでオンラインになる。そして会議の招集者はオンラインの利用者の中から会議への参加者を選択する。また、会議を招集するためには会議情報、すなわち会議の参加者、開催目的、開催日の調整期間、会議の所要時間の入力を行う必要がある。

次に、本システムを用いて会議日程調整を行なう際の参加者の操作と表示される情報を次に示す。会議の招集者と参加者の操作は会議情報の入力以外は同じである。

1. 会議の情報、すなわち会議の開催目的、開催日の調整期間、会議の所要時間が表示される
2. 表示された調整期間の予定を入力する
3. 入力した予定の一部を変更できないか尋ねる交渉画面が表示される
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、あるいは会議を開催する日時の調整に失敗したことが表示される

上記3の交渉は必要のない場合もあれば、複数回必要となる場合もある。上記2において会議参加者が行なう予定の入力はGUIにより簡便に行なうことができる。また本システムでは、利用者全員の予定をあらかじめ入力しておく必要はなく、招集された会議参加者が調整期間の予定だけを入力すれば良い。予定は1時間単位で入力し、以下に示す4つの中から選択する。

1. 会議に参加できる時間
2. 会議に参加できない時間
3. 交渉の余地のある時間
4. 会議の開催を希望する時間

ただし、上記4の会議の開催を希望する時間は、会議の招集者だけが選択できる。また、上記1の会議に参加できる時間は入力する必要がなく、入力されていない全ての時間を参加できる時間とみなす。

[†]鳥取大学大学院工学研究科

[‡]鳥取大学工学部

3. 会議日程調整システムの実装

会議日程調整システムは、我々の研究室で開発しているモバイルエージェントフレームワーク Maglog[1] を用いて実装している。Maglog とは Java 上に実装されたモバイルエージェントフレームワークであり、Prolog から Java へのトランスレータである PrologCafé[2] を拡張して実装されている。

Maglog は、エージェント、エージェントサーバ、フィールドからなる。エージェントは Prolog インタープリタを実行するスレッドであり、他のコンピュータに移動し、そのまま実行を継続する機能を持つ。エージェントが実行するプログラムは Prolog または Java によって記述される。フィールドとは、エージェントが実行できるプログラムを収めたデータベースのようなオブジェクトであり、データ交換用共有スペースとして利用できる。エージェントサーバは、エージェントの実行環境である。また、フィールドを保持しており、エージェントが移動する際に窓口となる。

エージェントは、フィールドに対してデータを読み書きすることで他のエージェントと通信を行う。フィールドからのデータの読み込みの際にはマッチングが行われ、マッチング可能な項が読み込まれる。読み込みを実行するエージェントは、マッチング可能な節がない場合、マッチングが可能な節が書き込まれるまでブロックされる。ただし、ブロックされない動作も可能である。

3.1 会議日程調整システム構成

本システムは、ユーザインタフェースプログラム、調整エージェント、ユーザ管理エージェントの 3 種類のプログラムから構成されている。ユーザインタフェースプログラム及び調整エージェントは全ての利用者のコンピュータ上で動作し、ユーザ管理エージェントはユーザ管理用コンピュータ上だけで動作する。それぞれのエージェントはエージェントサーバ上で動作するため、調整エージェントが動作するエージェントサーバを調整エージェントサーバ、ユーザ管理エージェントが動作するエージェントサーバをユーザ管理エージェントサーバと呼ぶ。

調整エージェントは利用者のコンピュータ間、すなわち調整エージェントサーバ間を移動して会議日程調整を行うプログラムであり、会議ごとに生成され日程調整が終わると消滅する。ユーザ管理エージェントは利用者の情報管理を行うプログラムである。ユーザインタフェースプログラムは利用者との対話、調整エージェントとの通信、およびユーザ管理エージェントとの通信を行なうためのものである。

調整エージェントは利用者の調整エージェントサーバを移動して会議日程調整を行うが、ユーザ管理エージェントとは通信を行わない。つまり調整エージェントの移動は Peer-to-Peer である。調整エージェントとユーザインタフェースとの通信には、調整用エージェントサーバ内のフィールドが使われ、ユーザ管理エージェントとユーザインタフェースとの通信には、ユーザ管理用エージェントサーバ内のフィールドが使われる。

3.2 ユーザインタフェース

ユーザインタフェースプログラムは Java の Swing により GUI として実装している。またユーザインタフェースと対をなす調整用エージェントサーバには、desk_field と work_field と呼ばれる 2 種類のフィールドが存在する。desk_field は調整用エージェントサーバ内にひとつだけ存在し、work_field は複数存在する。desk_field は会議ごとに生成された調整エージェントが受付を行うフィールドであり、受付を行うことで会議ごとにユニークな名前の work_field が作られる。desk_field では受付だけが行われ、実際のユーザのスケジュールの受渡しは work_field で行われる。つまり最初に desk_field を用いてユーザインタフェースと通信を行った調整エージェントは、次からは特定の会議専用の work_field を用いて通信を行う。

スケジュールの受渡しについて、会議の情報が work_field に書き込まれると、ユーザインタフェースはユーザに予定入力用のウィンドウをユーザに提供する。ユーザインタフェースは入力された 3 つの、会議に参加できない時間、交渉の余地のある時間、会議の開催を希望する時間をそれぞれ busy(Date), tentative(Date), priority(Date) 項として work_field に書き込む。Date には date(JulianDay, Hour) が入る。JulianDay は年月日を表すユリウス通日表現のアトム、Hour は時間を表すアトムである。

3.3 調整エージェント

調整エージェントは会議開催日の調整期間のはじまりとおわり、一日のうちで会議を開催できる時間の範囲のはじまりとおわり、会議の所要時間、会議に対するコメント、会議の招集者、会議の参加者の情報を与えられ生成される。会議開催日の調整期間のはじまりは date(JulianDay, Start) 項、おわりは date(JulianDay, End) 項で表される。一日のうちで会議を開催できる時間の範囲のはじまりとおわり、会議に対するコメント、会議の招集者はそれぞれアトム Start, End, Comment, Convener で表される。会議の参加者 (招集を含む) はそれぞれの調整用エージェントサーバ内の desk_field のリストで表される。生成された調整エージェントが日程調整を行なうアルゴリズムを次に示す。

1. リストの先頭から順に desk_field に移動し、各参加者のユーザインタフェースから今回の会議専用の work_field を取得し、新たに work_field のリスト List を作成する。
2. List の先頭から順に work_fiekd に移動し、開催日の調整期間内の予定を取得する。取得した予定は busy(Date, work_field), tentative(Date, work_field), priority(Date) の事実節として保持する。Date には date(JulianDay, Hour) 項が入る。Hour は時間を表すアトムである
3. 開催日時の調整期間で会議を開催できる時間帯の開始時間 Date 項のリストを作成し、リストから busy/2 項の Date 項とマッチするものを取り除いたリストを作成する。

4. このリストの各 Date 項と tentative/2 項の Date 項のマッチングを行い、リストの各 Date 項が示す時間帯で tentative/2 項の数 Weight と work_field の数 NumOfPeople を計算する。また、priority/1 項の Date 項ともマッチするか調べ、data(Date, Weight, People, NumOfPeople, Priority) 項のリストを作成する。People は work_field のリストであり、Priority は priority/1 項とマッチすれば 0 しなければ 1 のアトムである。
5. 作成した data/5 項のリストを Priority, NumOfPeople, Weight の順で降順にソートする。ソートすることによりリストが Weigh, NumOfPeople, Priority の小さい順に整列される。このリストから Date, People 項を取り出した data(Date, People) 項のリストを作る。
6. 作成したリストが空の場合、参加者全員の work_field に会議の開催に失敗したことを書き込む。
7. リストが空でない場合、リストの先頭から data/2 項を取り出し People が示す参加者と Date の時間帯で交渉する。ただし、People が空のリストである場合は交渉する必要がないので、Date が示す時間帯で会議が開催されることを参加者全員の work_field に書き込む。
8. 交渉が成功すれば、Date の時間帯で会議が開催されることを参加者全員の work_field に書き込む。交渉に失敗した場合、リストの次の data/2 項で先ほどと同様に交渉する。
9. 8を繰り返しリストが空になった場合、参加者全員の work_field に会議の開催に失敗したことを書き込む。

3.4 ユーザ管理エージェント

ユーザ管理エージェントは、ユーザ管理用エージェントサーバで実行されている。ユーザ管理用エージェントサーバでは、memberlist_field, online_field の 2 つのフィールドがユーザの登録情報の管理、ユーザの接続情報の管理に用いられ、query_field, answer_field の 2 つのフィールドがユーザインタフェースとの通信に用いられる。

memberlist_field にはユーザの登録情報、つまりメールアドレスとパスワードが、online_field にはオンラインであるユーザの情報、つまりメールアドレスと IP アドレスがそれぞれ格納される。query_field にはユーザからの要求が書き込まれる。具体的には、新規登録要求、登録解除要求、ログイン要求、ログアウト要求、ユーザが持っているオンライン情報の更新要求がある。それらの要求に対する回答が answer_field に格納される。

ユーザ管理エージェントは query_field を常に監視している。そして、query_field にユーザから先の要求が書き込まれると、要求を読み出し、要求に応じて実際の仕事を行うエージェントを生成する。そして、自身は query_field の監視を続ける。生成されたエージェントは memberlist_field あるいは online_field でマッチングを行い要求を満たし、回答を answer_field に書き込む。

4. 会議日程調整システム実行例

図 1 に本システムのメイン画面例を示す。off はシス

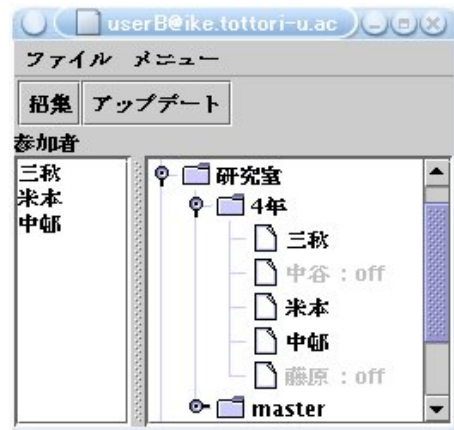


図 1: 会議日程調整システムメイン画面例

テム利用者であるがオンライン状態でない者を表している。招集者は右側のウィンドウに表示されているシステムの利用者の中から、会議に招集したい人をドラッグアンドドロップすることで左側のウィンドウに参加者として追加する。図 1 の左上の招集ボタンを押すことで会議を招集できる。その際図 2 に示す会議情報入力用ウィンドウに、会議開催日を調整する期間の始まりと終わりを指定してコメントを入力する。次に、スケジュール入力用ウィンドウが表示されるのでそこにスケジュールを入力する。



図 2: 会議情報入力用ウィンドウ例

以下に実行例として、招集者ユーザ A がユーザ B を参加者とし、会議開催日の調整期間を二日、会議の所要時間を 2 時間とした場合を考える。この時、ユーザ A とユーザ B は表 1 のように予定を入力している。priority は会議の開催を希望する時間を表しており、ユーザ A 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、入力していない空白部分は会議に参加できる時間を表している。ユーザ A は 7 月 5 日の 9 時から 11 時までの間に交渉の余地のある時間を、15 時から 17 時までの間に会議の開催を希望する時間を、7 月 6 日の 13 時から 17 時までの間に会議に参加できない時間を入力している。一方、ユーザ B は 7 月 5 日の 9

表 1: ユーザ A, ユーザ B の予定入力例

予定				
	7/5		7/6	
時間	ユーザ A	ユーザ B	ユーザ A	ユーザ B
9 時	tentative	tentative		busy
10 時	tentative	tentative		busy
11 時				busy
12 時		tentative		bysy
13 時		tentative	busy	
14 時		busy	busy	
15 時	priority	busy	busy	
16 時	priority	busy	busy	

時から 11 時までの間と 12 時から 14 時までの間に交渉の余地のある時間を, 14 時から 17 時までの間と 7 月 6 日の 9 時から 13 時までの間に会議に参加できない日を入力している. この場合, 会議の所要時間が 2 時間であり, busy として入力されている予定については交渉を行わないため, 交渉せずに会議の開催ができる時間帯は存在せず, ユーザ A もしくはユーザ B の予定を変更するために交渉を行なう必要がある. 交渉により会議の開催が可能になるのは 7 月 5 日のみであり, その際に交渉する時間帯, およびその時間帯で交渉する人数, 交渉する延べ時間数を表 2 に示す.

表 2: 会議開催可能な時間帯

会議の時間帯	交渉する人数	延べ時間数
9 時から 11 時	2 [人]	4 [時間]
10 時から 12 時	2 [人]	2 [時間]
11 時から 13 時	1 [人]	1 [時間]
12 時から 14 時	1 [人]	2 [時間]

会議の開催には次の時間帯が優先的に選ばれる.

1. 交渉する延べ時間数が 1 時間であり, 交渉する人数が一人である 11 時から 13 時までの時間帯
2. 交渉する延べ時間数が 2 時間であり, 交渉する人数が一人である 12 時から 14 時までの時間帯
3. 交渉する延べ時間数が 2 時間であり交渉する人数が二人である 10 時から 12 時までの時間帯
4. 交渉する延べ時間数が 4 時間であり交渉する人数が二人である 9 時から 11 時までの時間帯

つまり, 調整エージェントによりユーザ A とユーザ B に対して上記の時間の順番で交渉が行なわれる. 上記 1 での交渉に失敗すると, 次は 2, 3 というように順番に交渉が行われる. 上記 3, 4 では, ユーザ A ユーザ B とともに予定を変更しないと会議は開催できないが, 交渉は同時には行われず, 一人ずつ行われる. 例えば, ユーザ B より先にユーザ A と交渉を行なった場合, ユーザ A での交渉に失敗するとユーザ B とは交渉を行わない.

図 3 は, 表 1 の内容をユーザ A が入力している様子を示す. このように, 本システムでは予定の入力及び予定変更の入力を GUI による簡便な操作で実現している.



図 3: ユーザ A の予定入力画面

5. おわりに

本稿ではモバイルエージェント技術に基づいた会議日程調整システムを提案した.

本システムを用いて会議の招集を行なうと, 招集者は会議の情報を入力するだけで, 調整エージェントが参加者間のコンピュータを移動してスケジュールの収集及び予定変更の交渉を行い, 会議日程の調整が実施される. そのため従来のメールなどを用いた場合と比べて, 招集者が会議日程の調整を行なう負担を大幅に軽減できる. また, 従来のグループウェアとは異なり, 利用者全員が全ての予定を入力しておく必要がなく, 招集された参加者が調整期間内の予定だけを入力すれば良い.

今後の課題として, オンライン状態でない利用者が招集の対象に含めること, 及び必ずしも全員が参加しなくても会議を開催できる調整への対応を検討している.

参考文献

- [1] Takao Kawamura, Shin Kinoshita, and Kazunori Sugahara. Implementation of a mobile agent framework on java environment. In Teofilo Gonzalez, editor, *Proceedings of the IASTED International Conference Parallel and Distributed Computing and Systems*, pp. 589–593, 11 2004. MIT, Cambridge, USA.
- [2] Mutsunori Banbara and Naoyuki Tamura. Prolog cafe: A prolog to java translator system. <http://kaminari.istc.kobe-u.ac.jp/PrologCafe/>.