

モバイルエージェント技術を用いた会議日程調整システムにおける 一時的なシステム停止の実現

Meeting Scheduling System Based on Mobile Agent Technology

大西 貴之[†] 川村 尚生[†] 笹間 俊彦[†] 菅原 一孔[†]

Takayuki Onishi[†] Takao Kawamura[†] Toshihiko Sasama[†] Kazunori Sugahara[†]

[†] 鳥取大学大学院 工学研究科 情報エレクトロニクス専攻

1 はじめに

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為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つとして日程調整が挙げられる。既存の日程調整機能を持ったグループウェアは、参加者全員が事前に全ての予定を入力している事が前提となっており、それらの予定が共有されるものが一般的である。また参加者全員に都合の良い日時が無かった場合、予定変更の交渉を行うグループウェアは存在しておらず、交渉に関しては招集者が自ら行う必要があった。

そこで我々はモバイルエージェント技術を用いて、会議日程調整システム [1] を構築している。これは必要最低限な予定のみの入力だけで済むだけでなく、日程調整において、必要に応じて、指定した日時の予定を空けて貰えないかといった予定変更の交渉を行う。

本システムには全エージェントの移動の中心となる日程調整サーバが存在し、これを常時起動させておく必要があるが、メンテナンス等によりこれを停止させたい場合が考えられる。

本研究ではこれに対応する為、従来のシステムに一時的なシステムの停止を行う機能を実装する。

2 会議日程調整システム

2.1 会議日程調整システムの概要

一般的に会議を開催するには、開催を希望する期間、所要時間、参加者、開催目的等の情報が必要である。そして参加者全員の予定から全員に都合の良い日時を探す事により日程調整を行う。

本システムを用いて会議日程調整を行う際、招集者が設定する会議情報を以下に示す。

- 開催を希望する期間、所要時間を設定する。
- 開催目的等を記述する。
- 参加者を選択する。

また参加者の操作を以下に示す。予定変更の交渉に関しては、必要な参加者に対して必要な回数行われる。

1. 予定入力要求に応じて予定を入力する。
2. 予定変更の交渉が来た場合、それに返答する。
3. 日程調整の結果通知が掲示される。

システム側での開催日時の適用方法を以下に示す。

1. 参加者全員の予定から、開催可能な日時の候補を全て探索する。
2. 開催可能な日時の候補を、交渉する人数の少ない順にソートする。
3. ソートされた日時の候補を先頭から適用していく。

2.2 会議日程調整システムの構成

本システムの概観を図1に示す。

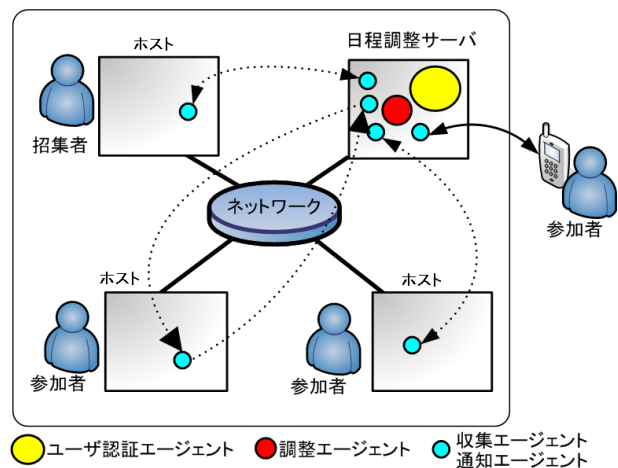


図 1: 会議日程調整システム概観

本システムでの情報の受け渡しはエージェントにより行っており、日程調整作業は日程調整に関するエージェントが行う。

日程調整に関するエージェントは、予定の収集を行う収集エージェント、予定変更の交渉を行う調整エージェント、結果の通知を行う通知エージェントで構成される。日程調整サーバ(以下サーバ)は1台のみ存在し、ユーザ情報及び会議情報を管理する。

日程調整に関するエージェントはサーバとユーザのホストとの間を移動し、サーバにて会議情報を共有する事により連携している。

また本システムの利用者は、ユーザインタフェースにより予定の入力、予定変更の交渉に対する返答、日程調整の結果の取得を行う。

2.3 日程調整に関するエージェントの動作

招集者により会議日程調整が開始されると、会議情報を持った調整エージェントが1つ生成され、これが日程調整作業を開始する。

調整エージェントは、まず参加者の人数分収集エージェントを生成する。全ての収集エージェントから予定を受け取ると、受け取った全ての予定から参加者全員に都合の良い日時を探す。参加者全員に都合の良い日時が存在した場合、また参加者全員に都合の良い日時が無く交渉を行う余地も無かった場合、日程調整の結果を参加者に通知する。参加者全員に都合の良い日時は無いが交渉を行う余地が存在した場合、調整エージェントは交渉可能な日時の中から、交渉する人数の少ない日時から順に適用し、交渉が必要な参加者のホストへ移動して交渉を行う。交渉により開催日時が決定すると、決定した開催日時を参加者に通知する。一定時間内に交渉に対する返答が得られなかった場合、参加者にメールを送信する事により、交渉に対する返答を求める。更に一定時間が経過すると他の日時に対して交渉を行う。他に交渉可能な日時が無かった場合は日程調整に失敗し、その結果を参加者に通知する。

収集エージェントは、調整エージェントにより参加者の人数分生成される。生成された収集エージェントは収集を担当する参加者のホストへ移動し、予定を収集する。予定を収集するとサーバへ移動し、収集した予定を調整エージェントに渡す。一定時間内に予定が収集できなかった場合、参加者にメールを送信する事により予定を収集する。更に一定時間が経過すると、予定が収集できなかった参加者の処理を招集者に尋ねる。招集者はこの参加者を欠席にするか、日程調整を失敗させるか返答する。一定時間内に返答を得られなかった場合、招集者にメールを送信する事により返答を求める。ここで一定時間が経過すると、日程調整の中止を調整エージェントに伝える。

通知エージェントは、調整エージェントにより参加者の人数分生成され、日程調整の結果を、参加者のホストへ移動して通知する。通知する参加者がログアウトしている場合、メールを送信する事により日程調整の結果を通知する。その後、参加者がログインした際に再度日程調整の結果を通知する。

日程調整の中でメールを送信しているが、メールにはURLが記載されており、参加者はそのURLにアクセスする事で外部からシステムにアクセスできる。

3 システム中断/再開機能

従来のシステムでは、全エージェントの移動の中心であるサーバを常時起動させておく必要があり、メンテナンス等でこれの停止が必要となっても、一時的でさえ停止させる事が出来ない。なぜなら、サーバでは不定期にエージェントの移動が発生する事に加え、サーバとユーザのホストとの間をエージェントが移動している途中に通信経路が途切れると、そのエージェントは消失し、システムに不整合が起こるからである。そこで、不整合を起こさないサーバの停止を可能にする

為、システム中断/再開機能を追加する。

エージェントの動作環境であるエージェントサーバを停止させるには、まずそこで動作しているエージェントを全て停止させなければならないが、エージェントはそれぞれ処理の内部状態を保持している為、これを保存する事によりエージェントの停止を実現できる。エージェントサーバの停止は、エージェントサーバにて動作しているエージェントを全て停止させた後、エージェントサーバの管理する情報を保存する事により実現できる。よって、エージェント及びエージェントサーバに対して、主記憶上のプログラムの内部状態をファイルとして二次記憶に保存する永続化機能が必要となる。

サーバを停止させる事は、この章の冒頭で述べた理由により極めて難しい。そこでサーバの停止を容易にする為、ユーザのホストにて動作中のエージェントをサーバに集結させる手法を用いる。更に新たなエージェントの生成を禁止する事でエージェントの移動が発生しなくなるので、この状況であれば不整合を起こさずにサーバを停止させる事が出来る。また本システムではユーザ認証処理においてもエージェントを生成するので、どのユーザも本システムを利用していない状況にする必要がある。よってログイン中の全ユーザをログアウトさせる処理が必要となった為、その処理を行うログアウト機能も追加する。

3.1 永続化機能の設計

永続化機能には以下に示す機能が必要である。

中断機能 処理の実行中、任意の状態中断し、その状態をファイルとして二次記憶に保存する機能

再開機能 二次記憶に保存されたファイルを用いて、中断時の状態に復元する機能

永続化機能を実現するには、エージェントに対してはエージェントの処理の内部状態、またエージェントサーバに対してはエージェントサーバの管理する情報の永続化が必要となる。

3.2 システム中断/再開機能の設計

一時的なシステム停止を実現するには、サーバの中断を行うシステム中断機能、サーバの再開を行うシステム再開機能が必要である。

システム中断機能の処理を以下に示す。これによりシステムに関する情報が全て保存されるので、サーバを停止させても不整合は起こらない。エージェントがサーバに集結し、サーバ上で動作しているエージェント及びサーバの管理する情報を、ファイルとして二次記憶に保存する様子を図2に示す。

1. ログイン及び新規登録処理を一時的に停止させる。
2. ログアウト機能を実行し、日程調整に関する全エージェントをサーバに集結させる。
3. 日程調整中の全エージェントが、エージェント中断機能により自身をサーバの二次記憶に保存するのを監視し、完了するまで待機する。
4. エージェントサーバ中断機能を実行する。

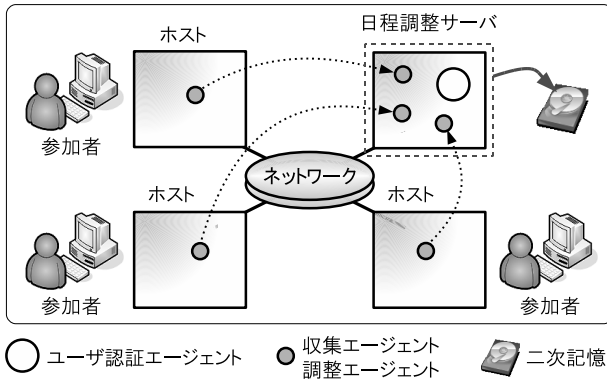


図 2: システムに関する情報の保存

システム再開機能の処理を以下に示す。これによりシステム中断機能を実行する直前の状態に復元される。

1. エージェントサーバ再開機能を実行する。
2. 一時的に停止していたログイン及び新規登録処理を再開させる。
3. 日程調整中であつた全エージェントに対し、エージェント再開機能を実行する。

3.3 ログアウト機能の設計

ログアウト機能は、ログイン中の全ユーザを一定時間でログアウトさせる処理を行う。ログアウト機能の処理を以下に示す。

1. ログイン中の全ユーザの情報を取得する。
2. ログイン中の全ユーザにログアウトを勧告する。
3. 一定時間待機する。
4. 再びログイン中の全ユーザの情報を取得する。
5. ログイン中の全ユーザに対し、以下の処理を行う。
 - (a) ログアウト処理を行う。
 - (b) ログアウトした事をユーザに通達する。

3.4 日程調整に関するエージェントの設計

日程調整作業を行う収集エージェント及び調整エージェントに関しては、サーバとユーザのホストとの間を不定期に移動する為、システム中断/再開機能の追加に伴い一部動作を変更する。

予定の収集中及び予定変更の交渉中においてシステム中断機能が開始された事を知ると、エージェント中断機能により、自身をサーバの二次記憶に保存する。

4 実装

本研究室では Maglog[2] を開発している。Maglog は論理型言語の Prolog で書かれたプログラムを Java の実行環境で実行させる事ができる、モバイルエージェントフレームワークである。Maglog にはフィールドと呼ばれる概念が存在し、エージェントはフィールドを介して他のエージェントとの通信を行う。

本システムは Maglog を用いて実装しているが、これに前章で述べた永続化機能が実装された [3]。よってシステム中断/再開機能を実装する事が可能となった。

システム中断/再開機能の追加に伴い、永続化フィールドを追加する。永続化フィールドは、システム中断機能が開始された事を告示すると共に、会議毎に作成されるフィールドを全て管理する。これにより、日程調整に関する全エージェントを管理できる。システム中断/再開機能は処理の中でこの情報を用いる。

日程調整に関するエージェントは、永続化フィールドを介してシステム中断機能が実行中か判断する。またログアウト機能においても永続化フィールドは使用され、ユーザのログアウトの完了等の確認を行う。

収集エージェント及び調整エージェントに関して、会議毎に作成されるフィールドを使用し、エージェント中断機能により自身をサーバの二次記憶に保存した事を明示する。

システム中断機能は、会議毎に作成されるフィールドに明示されている情報を用いて、日程調整に関するエージェントが保存されたか判断し、全てが二次記憶に保存されたと確認でき次第、エージェントサーバ中断機能を実行する。

システム再開機能は、まずエージェントサーバ再開機能を実行する。その後会議毎に作成されるフィールドから、再開させるエージェントのリストを取得し、その全てに対してエージェント再開機能を実行する。

システム中断/再開機能は、エージェントが実行する手続きとして実装した為、サーバの管理者はエージェントを生成する事によりこの機能を使用する。

5 動作検証

システム中断/再開機能の動作検証を、サーバ用のホストが 1 台、システムを利用するホストが 12 台で行った。表 1 に示す 3 つの状況において、1 度ずつシステム中断/再開機能を実行した。

表 1: システム中断/再開機能の動作検証

1 : 4 人の会議を 2 つ招集	
4 人の会議	予定の収集中
4 人の会議	予定の収集中
2 : 4 人の会議を 2 つ、12 人の会議を 1 つ招集	
4 人の会議	メールによる予定の収集中
4 人の会議	タイムアウトした参加者の処理
12 人の会議	予定の収集中
3 : 4 人の会議を 2 つ、12 人の会議を 1 つ招集	
4 人の会議	メールによる予定変更の交渉中
4 人の会議	メールによるタイムアウトした参加者の処理
12 人の会議	予定変更の交渉中

その結果、システムの利用により移行する可能性のある全ての状況において、また 12 人が利用し、複数の会議の日程調整を並行して行っている状況において、

不整合を起こさずにサーバを停止させる事が出来た。これにより、システム中断/再開機能が正しく実装できている事が確認できた。

5.1 実行例

会議情報入力画面を図3に示す。

図 3: 会議情報入力画面

予定入力画面を図4に示す。図3において選択したテンプレートによって、図4におけるそれぞれの時間帯の開始時刻及び終了時刻が変化する。また図4において、12:00~13:00の黒色で何も表示されていない時間帯は長時間休憩を表しており、この時間帯を跨いでの日程調整も可能である。

	2/2 (月)	2/3 (火)	2/4 (水)	2/5 (木)	2/6 (金)
8:50	BUSY	TENTATIVE	FREE	TENTATIVE	FREE
10:20	BUSY	TENTATIVE	FREE	FREE	
10:30				TENTATIVE	
12:00					
12:00					
13:00					
13:00	TENTATIVE	FREE	PREFERABLE	PREFERABLE	
14:30				BUSY	
14:40	TENTATIVE	FREE	PREFERABLE		
16:10					
16:20	TENTATIVE	FREE	FREE	FREE	BUSY
17:50					

図 4: 予定入力画面

6 おわりに

本研究では、会議日程調整システムに一時的なシステム停止を行う機能を実装した事により、サーバの管理する情報を保持した状態でのサーバの停止が可能となった。これにより、メンテナンス等に伴う電力供給の停止に対応し、サーバの停止している期間を跨いでの日程調整が可能となった。

今後の課題として、サーバの停止中に日程調整の期限が過ぎた会議への対応、停止させたい日時を事前に登録しておく事によるサーバの停止の自動化等が挙げられる。

参考文献

- [1] Yusuke Hamada, Shinichi Motomura, Takao Kawamura, and Kazunori Sugahara. NAT Traversal Method for Multi-Agent-based Meeting Scheduling System. In *Proceedings of the Third International Conference on Internet and Web Applications and Services*, pp. 223–226, 6 2008. Athens, Greece.
- [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.
- [3] Shinichi Motomura, Junya Kishida, Takao Kawamura, and Kazunori Sugahara. Realization of Persistency in a Multi-Agent Framework. In *Proceedings of IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pp. 28–33, 4 2007. Waltham, Massachusetts, USA.