

(d) 情報

モバイルエージェントによるスケジュール調整に関する研究

木下 慎[†] (鳥取大学大学院工学研究科知能情報工学専攻)
川村 尚生[†] (鳥取大学工学部知能情報工学科)
菅原 一孔[†] (鳥取大学工学部知能情報工学科)
[†]{kinosita,kawamura,sugahara}@ike.tottori-u.ac.jp

1 はじめに

近年、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が注目されている。このようなソフトウェアはグループウェアと呼ばれ、グループ内での情報の共有、交換を支援するための機能を持つ。

このようなシステムの構築する為の技術として、モバイルエージェント技術が適していると考えられる。モバイルエージェントとは、ネットワーク上を自由に移動する能力を持った自律的ソフトウェアである。

現在、我々はモバイルエージェントフレームワーク Maglog[1] を開発中である。

本論文では、Maglog の応用として、グループウェアの一つであるスケジュール調整アプリケーションを取り上げ、その実現について述べる。

本論文の残りは以下のように構成されている。2 節で Maglog について説明し、3 節でスケジュール調整アプリケーションの概要を述べ、4 節でその実装について述べる。5 節で本論文をまとめる。

2 Maglog の概要

Maglog において、エージェントのプログラムは Prolog で記述され、エージェントはエージェントサーバというプロセス内で動作する。エージェントサーバ内にはフィールド [1] というオブジェクトが存在し、エージェントはデータやプログラムを格納することができる。

エージェントは他のエージェント、もしくはユーザにより、

```
create(-ID, +AgentName, +Goal)
```

という組み込み述語により生成される。

ID は生成されるエージェントの識別子であり、AgentName は生成されるエージェントのプログラムを指定するアトムである。Goal は生成されるエージェントの初期ゴールである。

以降、組み込み述語の説明において、引数の前に付く '-' は出力引数、'+' は入力引数、'?' は入力引数、出力引数のどちらでもよいことを示す。また create/3 のように、述語名と引数の数により組み込み述語を省略して表記する場合がある。

フィールドはエージェントの環境を表しており、エージェントの動作は現在そのエージェントが属している環境、すなわちフィールドによって左右される。エージェントは、

```
in(+Goal, +FieldID)
```

により、FieldID で示されるフィールド内の手続きをスコープに入れ Goal を実行する。

エージェントは、

```
fassert(+Clause, +FieldID)
fretract(+Clause, +FieldID)
fclause(?Head, ?Body, +FieldID)
```

などの組み込み述語を用いて、フィールドへの Prolog 節の追加、削除、読み出しを行なうことができる。

フィールドはまた、タブルスペースとしての機能を持っており、エージェントが fretract/2, fclause/3 を実行した際、削除、読み出しの対象となる節がフィールドに存在しない場合、ブロックされる。

この機構を用いた例として、エージェントによる生産者/消費者モデルを図 1 に示す。Prolog のパターンマッチング機構とタブルスペースの組み合わせにより、エージェント間の同期、柔軟なメッセージ通信が実現できる。

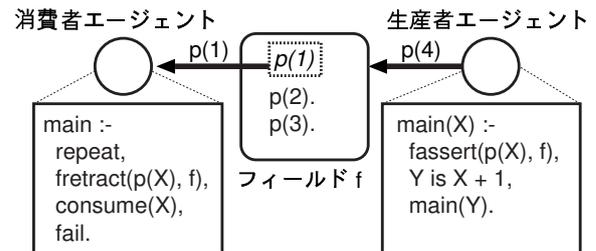


図 1: エージェントによる生産者/消費者モデル

in/2, fassert/2 など引数となる FieldID は、ホスト情報を含むことができる。すなわち、fassert(Clause, FieldName@HostAddress) とすることができ、この場合、エージェントは HostAddress で示されるコンピュータ上へ移動し、FieldName で示されるフィールドに Clause を追加し、元のコンピュータ上に戻る。FieldName はフィールドの名前となるアトムであり、FieldID が単に FieldName である場合には、ローカルホストであると解釈される。

Maglog での、エージェントの移動は強マイグレーションと呼ばれる。強マイグレーションとは、移動後も移動前の状態からプログラムの実行が再開できる移動方式である。この移動方式の性質により、Maglog ではホストをまたいだユニフィケーションやバックトラックが可能である。例えば、

```

common(X) :-
    in(data(X), field@host1),
    in(data(X), field@host2).

```

とすれば、ネットワークに存在する複数のフィールドから共通するデータを探すことになる。プログラマは、コンピュータ間の移動、バックトラックなどを全く意識する必要は無く、Prolog プログラムの延長として自然にプログラムが記述できる。

3 スケジュール調整アプリケーション

会議の日時を定める場合、参加者ごとに希望する日時は異なるため、参加者同士で交渉し、最も都合の良い日時を決定する必要がある。この作業を人間の代わりに行なうエージェントを考える。

会議等の召集者はまず、エージェントに召集したい会議の日時の許容範囲と、その時間数、および召集したい参加者のリストを与える。例えば、「範囲は8月14日から8月19日の間、時間数は2時間、AさんとBさん」といった情報を与える。

次にエージェントは、会議に召集される各参加者から、指定された日時の範囲内のスケジュール情報を得て、全員が会議に参加できる日時を探す。全員が参加できる日時が見付からない場合、各参加者に対して、あらかじめ決められている予定の変更を交渉し、会議の日程を決定しようとする。

この時、予定の変更を申し込む人数や時間数が少ない日時を優先するといった判断基準をエージェントに持たせることによって、スケジュール調整という煩雑な作業を人間の代わりに行なうシステムを構築することができる。

4 Maglog によるスケジュール調整アプリケーションの実装

スケジュール調整アプリケーションを Maglog で実装するに当たって、図2に示すような構成のアプリケーションを考えた。

本アプリケーションの利用者は、あらかじめエージェントサーバと秘書エージェントを用意する。秘書エージェントとは、利用者のスケジュールを管理し、利用者の代わりにスケジュール調整エージェントとの情報交換を行なうエージェントである。スケジュール調整エージェントは、会議の召集者によって生成され、参加者のホスト間を移動し、秘書エージェントに対し、スケジュール情報の問い合わせや予定変更の交渉を行ない、会議の日時を決定する。

スケジュール調整エージェントと秘書エージェントの情報交換はフィールドを介して行なわれる。秘書エージェントは `fretract/2` を繰り返し実行し、図1のように、フィールドをキューとして利用する。

実装したスケジュール調整エージェントのアルゴリズムは以下の通りである。

1. 会議の召集者によって生成され、召集したい会議の日時の許容範囲と、その時間数、および参加者リストを受け取る。召集したい会議の日時の許容範囲は、2つの `date(Month, Day, Hour)` 項で、参加者リストはそれぞれの参加者の秘書エージェン

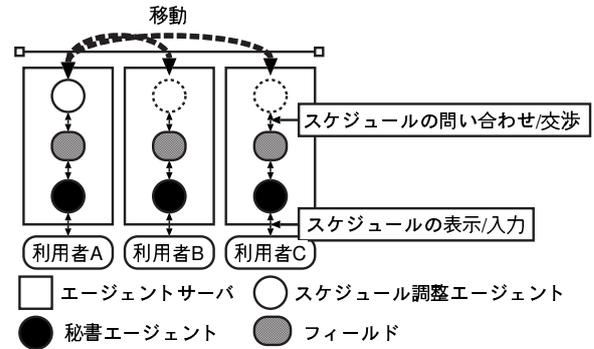


図2: スケジュール調整アプリケーションの構成

トがメッセージを待っているフィールドのIDのリストとして表される。

2. 全ての参加者(召集者含む)が、召集者が設定した日時の許容範囲について、会議に参加できない時間をそれぞれの秘書エージェントに尋ねる。調整エージェントはこの情報を `busy(Date, Field)` 事実節として保持する。Dateには先の `date/3` 項が入る。
3. 時間の許容範囲内全ての時間に関して、欠席者の人数、のべ欠席時間数を計算し、解候補リストを生成する。このリストは `data(Date, Hours, Fields, NumOfAbsent)` 項のリストである。
4. 解候補リストを欠席者の人数、のべ欠席時間数の順で降順にソートする。
5. 解候補リストのそれぞれのデータについて、Fieldsが示す参加者に対して、会議を開くために予定をキャンセルしてもらえるかどうかを尋ねる。
6. Fieldsに含まれる参加者全員がキャンセルを受け入れる。もしくはFieldsが空リストである場合、その時間が会議の時間として参加者全員に通知し、終了する。
7. 会議時間が決定するまで繰り返し、リストが空になった場合は会議の開催に失敗したことを参加者全員に通知し終了する。

このアルゴリズムでは、のべ欠席時間数が少ない日時、同じ時間数であれば欠席者の人数が少ない日時が優先される。このスケジュール調整エージェントのプログラムサイズは約150行である。

5 おわりに

本論文では、Maglogによるスケジュール調整アプリケーションの実現に関して述べ、Maglogが、スケジュール調整などのグループウェアの構築に適していることを示した。

参考文献

- [1] Takao Kawamura, Shin Kinoshita, Kazunori Sugahara, and Tsuyoshi Kuwatani. A logic-based framework for mobile multi-agent systems. In Henry Hexmoor, editor, *Proceedings of International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, pp. 754–759, 10 2003. Boston, Massachusetts, USA.