

回覧板プロトコルを用いたスケジュール調整エージェント構築に関する一考察

2 A F - 7

中村 伸明

沢村 一

新潟大学大学院自然科学研究科 新潟大学工学部情報工学科

1 はじめに

我々の日常生活でスケジュール調整という問題は、大切な時間を浪費し面倒な作業である。スケジュール調整を行なうエージェントとして、ユーザの空き時間、希望などを考慮し機械的に計算するものは存在する。しかし、ユーザの好み、カレンダーの空き時間を考慮し、他のエージェントと通信を行ない、交渉する能力をもち、本当の意味で自動的に作業をおこなってくれるものはまだ存在しない [1]。契約ネットプロトコルを用いてスケジュール調整を行なう方法もあるが、得られる合意は2エージェント間のものであるため、他エージェントの表明を参照し自分の意志を決定できないなどの問題がある。

本稿で報告するエージェントは以下の特徴をもつ。他エージェントの表明を参照し意志決定を行なうプロトコルとして回覧板プロトコルを使用する。これによって、 n エージェント間の合意を扱うことが可能となる。また他エージェントの合意内容によって必要に応じて自らの意向を表明することも可能となる。この場合のスケジュール調整は、他の出席者の意志を参照し、自分の意志を決定する形で行なわれる。また、Java 言語を用いて実装したため、Java が使用できる環境であればどのプラットフォームでも実行可能である。

スケジュール調整エージェント構築の概要を報告する。

2 回覧板プロトコル

エージェント間で合意を形成するためのプロセスを交渉 (Negotiation) と呼ぶ。交渉は合意形成のための代表的な手法であり、契約ネットプロトコルによるものなどがある [2]。契約ネットプロトコルは、タスク割当など2エージェント間の合意を扱う場合に使用される。エージェントは合意候補の中から自分の意志を表明し、ホストとなるエージェントが全エージェントの表明をもとに一つの合意案を導く。

回覧板プロトコルは、 n エージェント間の合意を扱うことが可能なプロトコルである。合意形成を行なうためにエージェント間で回覧板を巡回させ、各巡回で意志表明を行なう。以下に、本稿で実装する回覧板プロトコルを示す。

- ホストとなるエージェントは、はじめに合意案を候補の中から1つ選択する。次に、その合意案を送る相手 (複数可) を決定する。回覧板に合意案と、合意案を送る相手を記入し、あるエージェントに回覧板をまわす。
- 合意案を受けとったエージェントは、ホストの案および、他のエージェントの表明を参照し、その案に対する自分の意志表明を回覧板に記入する。次のエージェントに回覧板をまわす。
- 最後にホストが戻ってきた回覧板をチェックする。合意に至れるなら全エージェントに合意の確認を行ない、駄目ならば、制約を緩め再度回覧板をまわす。

3 エージェントの構成

本稿ではエージェントを、3つのクラスで構成する。それぞれコミュニケーションクラス、知識クラス、処理クラスと呼ぶ。

3.1 処理クラス

処理クラスは、エージェントの問題解決部分を担当する。スケジュール調整における効用の評価、合意案の選択・評価を主に行なう。また、スケジュール調整における過去の履歴評価なども行なう。

3.2 コミュニケーションクラス

コミュニケーションクラスは、ユーザとの対話、他エージェントとの通信を行なう。通信はJava 言語のSocket クラスを使用しTCP/IP により行なう。エージェント間通信では、回覧板プロトコルに従いメッセージ文字列の生成・送信、受信したメッセージのチェックなどが行なわれる。回覧板には、回覧番号、回覧をまわすべきエージェント名、スケジュール調整に関する記述が含まれる。

3.3 知識クラス

知識クラスは、処理クラスとコミュニケーションクラスで利用されるさまざまな知識や情報を提供する。スケジュール調整の知識として、ユーザの空き時間などが記されたカレンダー、ユーザの調整に関する希望、

A consideration on modeling Scheduling Agent using Circulation Board protocol, Nobuaki Nakamura, Graduate School of Science and Technology, Niigata University, Hajime Sawamura, Dept. of Information Engineering, Faculty of Engineering, Niigata University

過去の調整履歴などが格納・管理される。また、エージェント間で通信をするために、自分自身の情報や他エージェントが存在するホスト名などネットワークに関する知識が格納される。

4 スケジュール調整に必要なパラメータ

スケジュール調整に使用するパラメータとして、会議を行なう日付、会議の開始時間、会議時間の長さ、会議参加の優先度、各会議に参加すべきメンバーの優先度を考える。

会議 j の時間を表すベクトルを \mathbf{I}^j 、エージェント k が希望する時間を表すベクトルを \mathbf{P}^k 、ベクトルの各成分に対する重みを \mathbf{W}^k とする。各ベクトルはその成分として日付、開始時間、長さを要素とし以下のように定義する。

- $\mathbf{I}^j = (\text{日付}, \text{開始時間}, \text{長さ})$
- $\mathbf{P}^k = (\text{日付}, \text{開始時間}, \text{長さ})$
- $\mathbf{W}^k = (\text{日付の重み}, \text{開始時間の重み}, \text{長さの重み})$
重みの範囲は 0 以上 1 以下とする

上記のベクトルを用いて、あるエージェントが会議に割り当てる優先度、及びホストエージェントが合意案の選択に使用する効用関数を定義する。会議 j の時間と、エージェント k が希望する時間との重み付き距離を式 (1) で定義する。

$$WDist(\mathbf{I}^j, \mathbf{P}^k, \mathbf{W}^k) = \sum_{i=1}^3 w_i^k (I_i^j - P_i^k) \quad (1)$$

ここで右辺の下添字は、ベクトルの第 i ($i = 1, 2, 3$) 成分を表す。

式 (1) の重み付き距離を用いて、エージェント k が会議 j に割り当てる優先度 ($\in [0, 1]$) を式 (2) で定義する。

$$Priority_k(\mathbf{I}^j) = \frac{\sum_{i=1}^3 [\text{成分}_i \text{の幅}] - WDist(\mathbf{I}^j, \mathbf{P}^k, \mathbf{W}^k)}{\sum_{i=1}^3 [\text{成分}_i \text{の幅}]} \quad (2)$$

ここで成分 i の幅は、ベクトルの第 i 成分のスケジュール調整可能な最大値を表す。この関数を使うことにより、エージェント k は会議 j に対し優先度を割り当てることができる。

ホストエージェントが会議 j の合意案を評価するときの効用関数を式 (3) で定義する。

$$Utility_j = \frac{\sum_{i=1}^{\text{Agent の数}} Priority_{i_j}(\mathbf{I}^j)}{\text{Agent の数}} \times 10 \quad (3)$$

ホストエージェントは各巡回に於いて、式 (3) の効用関数の値が減少しないように次回の見覧板の内容を

決定する。各エージェントは、他のエージェントの表明を参照しその時点での会議に対する優先度が最大になるように意志表明する。

5 エージェントの実装

見覧板プロトコルを用いたスケジュール調整エージェントの実装を行なった。プログラミング言語には Java 言語を使用する。このため Java の実行環境さえあれば、どのプラットフォームでも動作可能である。

各エージェントは以下の手順に従いスケジュール調整を行なう。

1. 会議の調整が必要なら、ホストとなるエージェントはユーザに与えられた希望をもとに合意案の候補を探す。ホストエージェントは、参加する可能性のあるエージェントのリストを作成する。見覧板に、会議の予定時間、参加者リストを記入する。
2. 見覧板プロトコルに従い、会議時間を出席者へアナウンスする。
3. 見覧板を見たエージェントは、ホストの提案、他の参加者の意志表明を参考に自分の意志表明を見覧板に記入、次のエージェントに送る。
4. 参加者全員が、ホストの提案を受け入れたら調整終了。
5. ホストの提案が少なくとも 1 エージェントに拒否されたら、エージェントの表明が収束するまで、あるいは規定の回数までスケジュール調整プロトコルを繰り返す。

6 まとめ

本稿では見覧板プロトコルを利用したスケジュール調整エージェントのモデルを提案し、その実装について述べた。今後は、評価の方法として、表明が収束するまでのエージェント間の交渉回数、時間、調整の質などを調べる。また他のプロトコルを用いてスケジュール調整を行なった場合にエージェント間で行なわれる交渉の回数、合意案に収束するまでの時間、スケジュール調整の質などについて比較検討をしていく。

参考文献

- [1] Leonardo Garrido, Katia Sycara, Multi-Agent Meeting Scheduling: Preliminary Experimental Results, ICMAS-96, 1996, pp.95-102.
- [2] 石田 了, 片桐 恭弘, 桑原 和宏, “分散人工知能”, コロナ社, 1996