

エージェントによる会議予約システムの試作と評価*

2N-2

秋本綾子 井上健 根岸雅子 日野泰臣†

横河電機(株) 中央研究所‡

1 はじめに

ネットワーク上に分散して存在するエージェントが互いに通信しあって問題解決に当たる「ネットワークエージェント」の例題として、会議予約システムを連邦アーキテクチャに基づいて試作した。本稿では、試作したシステムの概要を説明し、エージェントシステムの開発手法や、連邦アーキテクチャの効果等について、考察を行う。

2 システムの概要

2.1 機能

今回試作した会議予約システムの機能は、大きく次の二つに分かれる。

1. 会議室予約（図1左下の画面）

タイムテーブル形式で表されている会議室の1日分の予約状況から、人間が目で見えて空いているところを見つけ、時間を厳密に指定して予約する。

2. 会議アレンジ（図1右の画面）

希望の日時・場所・参加者をおおまかに指定すると、参加者のスケジュールを考慮しながら、条件を満たす日時を割り出し、会議室を予約し、参加者に会議開催通知メールを送るものである。指定の際には、「来週の後半」「午後2時間位」「OHPがあり15人入れる会議室」「Aチームの人」のような曖昧な指定が可能である。また、参加者のスケジュールは合うが会議室が取れない場合に、会議のセッティングだけを行い、目当ての会議室の予約がキャンセルされたらメールで通知する機能も持つ。これらの条件の入力は、ボタンやリストメニューを選択して行うことも出来るし、限定された言い回しのみではあるが自然言語（英語）により行うことも出来る。

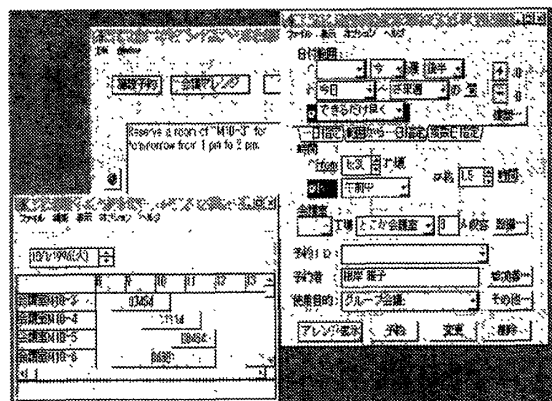


図1: 会議予約システムのGUI画面

2.2 構成

本システムは、独立したプロセスであるエージェントたちが、協調促進器（Facilitator）と呼ばれる一種の黒板を介して通信しあう、連邦アーキテクチャに基づいている。実現に当たっては、米国SRI (Stanford Research Institute) International が開発したエージェントシステム構築支援環境 OAA (Open Agent Architecture) を利用した。OAA では、Facilitator 自体と、Facilitator を介したエージェント同士の通信部分のライブラリが用意されており、我々は、エージェントの機能自体の開発に専念することが出来る。また、複数のプラットフォーム (PC, UNIX) と言語 (C, Delphi, Prolog, Lisp, Visual Basic, Java) をサポートしているため、プログラムの性質や開発者のスキルに応じた柔軟なシステム構成が可能となる。

以下に本システムを構成する9つのエージェントを示す（図2参照）。

- ・会議 GUI: ユーザの指定をシステム内部形式に変換し、実処理を行うエージェントに渡す。
- ・会議コーディネータ: ユーザの要求を解釈し、必要な情報を集めてきて、会議をコーディネートする。様々なエージェントに処理を割り振り、結果をまとめる一種のメタエージェント。
- ・スケジューラ: 会議室や人間のスケジュールの共通の空き領域を探し、所望の数だけリストアップする。

*Prototype of Agent based Meeting Scheduling System

†Ayako Akimoto, Takeshi Inoue, Masako Negishi, Yasutaka Hino

‡Yokogawa Electric Corporation

2-9-32 Nakacho, Musashino-shi, Tokyo, 180 Japan

- ・日付ソルバ、時間ソルバ：日付や時間に関する曖昧な表現を解釈し、正確な日付・時間のリストに変換する。
- ・会議室マネージャ、人情情報マネージャ：会議室や人間といった資源の属性データおよびスケジュールの管理を行う。
- ・自然言語：会議室を予約する簡単な英文を解析し、システム内部形式に変換する。
- ・電子メール：電子メールの送信や、受信の監視などを行う OAA 付属の既存エージェント。

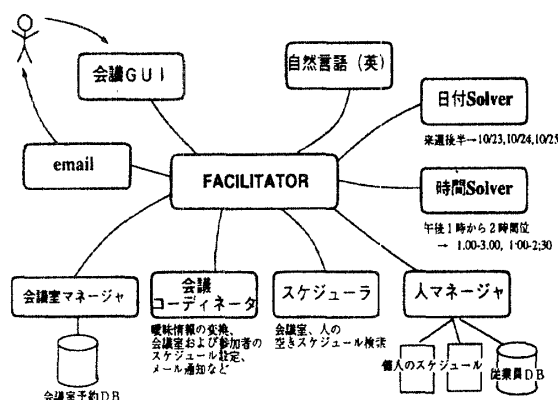


図 2: 会議予約システムのエージェント構成

それぞれのエージェントが、他のエージェントに何らかの処理を依頼する時は、何をして欲しいかを Facilitator に掲示すれば、Facilitator がそれを適切なエージェントにルーティングするため、エージェント自体はどのエージェントがその問題を解くことができるか知らなくてよい。

3 考察

エージェントシステムを構築する場合に、最も基本的で重要な問題は、エージェントの洗い出しであろう。我々は、「シナリオに基づいて役割を考え、エージェントを洗い出す」というオブジェクト指向のユースケースを用いた開発に近い方法をとったが、これは有効であった。注目する単位は、オブジェクトよりも粒度の大きい「役割」であり、実際に人間が受け持つとしたらどうなるかという観点から、独立した一つのエージェントとして存在し得るだけの役割を持つか、逆に複数の役割をまとめ過ぎてはいないかなどに留意した。

連邦アーキテクチャでは、エージェントたちは、機能に注目して相手を特定しない自由な通信が可能である。誰から利用されるか定まっていらないからこそ、特定の相手に対してのみ有効にならないよう、メッセージの形式は慎重に決めなければならない。例えば、今回のシステムのユーザインタフェースには GUI と自然言語があるため、英語の構文からも導きやすい形式になるよう気を付けた。

今回のシステムでは、曖昧さを含まないプリミティブな予約は、会議コーディネータだけでなく会議室マネージャも受け取ることが出来る。そのため、会議コーディネータがつかない時でも、厳密な指定であれば予約が可能である。これは、「誰が」解くかではなく「何を」解くかという、機能に注目する連邦アーキテクチャならではの利点であろう。一方、実際には、会議コーディネータにはメールを出すという付加的な機能もついており、曖昧さを含まない指定であっても会議コーディネータに解いて欲しいという場合がある。しかし、Facilitator は基本的には単純なルーティングしか行わないため、会議室マネージャではなく会議コーディネータを選ぶ仕組みはエージェント側に作りこまなければならなかった。以上のことから考えても、連邦アーキテクチャは機能の異なるエージェントにより構成される場合にもっとも威力を発揮し、同じような機能を持つエージェントが複数存在するような場合には、選択機構を組み込むために負荷が発生してしまう。

4 おわりに

連邦アーキテクチャに基づくネットワークエージェントシステムとして会議予約システムを試作し、評価について述べた。ネットワーク技術等の進展により、今後ますます異機種分散環境におけるアプリケーションの重要性は高くなっていくことが予想される。我々は、今回の試作を足掛かりとして、ネットワーク上に分散する様々なサービスを効率よく連携させ統合する技術であるネットワークエージェントの研究を進め、特に産業分野への適用例を検討していく予定である。

参考文献

- [1] 木下哲男, 菅原研次, 「エージェント指向コンピューティング」, SRC(1995)
- [2] P.R.Cohen, A.J.Cheyre, M.Wang, S.C.Baeg, "An Open Agent Architecture", in AAAI Spring Symposium, pp.1-8, March 1994