

ルールによる知識獲得のためのソフトウェアエージェント構築技法*

多賀谷 寿子 中所 武司†

明治大学大学院理工学研究科基礎理工学専攻情報科学系‡

{tagaya,chusho}@cs.meiji.ac.jp

1 はじめに

インターネットの普及に伴い、ユーザのネットワーク環境は急速に複雑化している。ネットワークを介したユーザの知的作業を代行するソフトウェアとして、ソフトウェアエージェント [1] が注目されている。ユーザの意図に沿った支援をさせるためには、ユーザが自らソフトウェアエージェントに知識を教える必要がある。

本稿では、ソフトウェアエージェントにユーザの知識をルールとして与える技法およびオブジェクト指向技術による実現方式について述べる。ルールをルールオブジェクトとして実現する方式により、ユーザ自身が動的にルールを追加し、次第にユーザに特化させることを容易にした。

2 ルールベースのエージェント構築

2.1 ソフトウェアエージェントの構成

ソフトウェアエージェントは、図1に示すように、意思決定機構、実行制御部、他のエージェントと対話するための通信部、ユーザと対話するためのUI、ユーザから知識を獲得するためのルールエディタから成る。

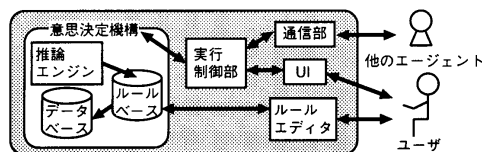


図1: ソフトウェアエージェントの構成

ユーザからの知識獲得を容易にするため、意思決定機構に、記述容易性および柔軟性の高いルールベースを利用する。ユーザは、ルールエディタを用いて動的にルールを追加、変更することによりソフトウェアエージェントを構築していく。

*Construction of Software Agents for Knowledge Acquisition with Rules

†Hisako Tagaya and Takeshi Chusho

‡Computer Science Course, Major in Sciences, Meiji University

2.2 ルールオブジェクトの実現方式

オブジェクト指向技術を適用したルール実現方式としては、オブジェクト指向モデルの下で、規則的知識の記述のためにルール群でメソッドを定義する方式 [2] などがあるが、本研究では、ルールの動的な追加、変更を目的として、ルールを自ら照合、実行するオブジェクトとして実現した。

そのルールオブジェクトが照合、実行のために必要なオブジェクトを参照する方式としては、以下の2つが考えられる。

1. 直接参照する方式
2. データベースを介して間接的に参照する方式

1の方式では、推論エンジンは各々のルールがどのオブジェクトの参照を必要としているかを認識して、その参照をルールオブジェクトに渡さなければならないので、処理が複雑になる。

2の方式では、データベースはオブジェクトの名前と参照先をテーブルとして持ち、ルールオブジェクトはデータベースから必要なオブジェクトの参照を得ることができる。そこでこの方式を採用することにより、推論エンジンがルールオブジェクトにデータベースへの参照を渡すだけで、推論実行を可能とした。

2.3 推論の実行方式

意思決定機構は、推論エンジン (InferenceEngine)、ルールベース (RuleBase)、データベース (DataBase) から構成される。ルールベースはルールオブジェクト (Rule) の集合であり、各ルールオブジェクトは条件部オブジェクト (Condition) と実行部オブジェクト (Execution) を有する。

推論方式は前向き推論で、競合解消にはユーザにとって最も理解しやすいと考えられるファーストマッチを採用している。

2.2節に示した2の方式による意思決定機構の推論の流れをUMLの記法を用いて図2に示す。推論エンジンはルールベースのルール取得 (getRule) メソッドにより

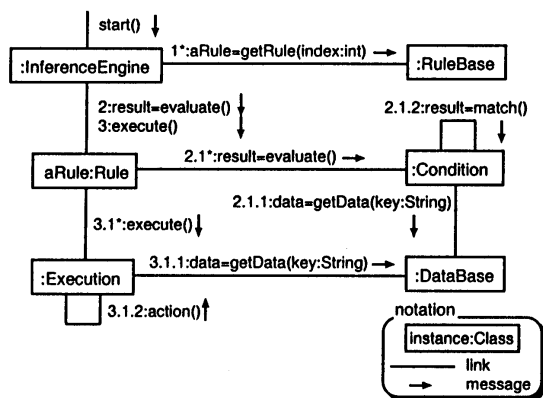


図 2: 推論の流れを表すコラボレーション図

ルールの参照を得る (図 2 のメッセージ 1)。推論エンジンはルールの評価 (evaluate) メソッドを呼び出し (2), ルールは条件部の評価メソッドを呼び出す (2.1)。条件部はデータベースのデータ取得メソッド (getData) により, 必要なオブジェクトの参照を取得し (2.1.1), 照合 (match) を行う (2.1.2)。推論エンジンは条件部の成立したルールに対して, 実行 (execute) メソッドを呼び出し (3), ルールは実行部の実行メソッドを呼び出す (3.1)。実行部はデータベースから必要なオブジェクトの参照を取得し (3.1.1), 実行 (action) を行う (3.1.2)。

後述するスケジュール調整エージェントシステムの“スケジュールが空いているならば 参加する”というルールを例に挙げ, オブジェクトへの参照方式を説明する。

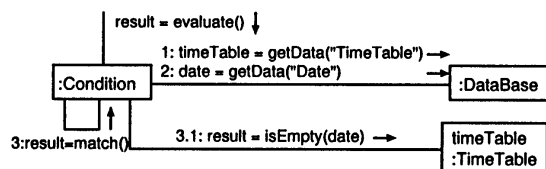


図 3: 照合の流れの例

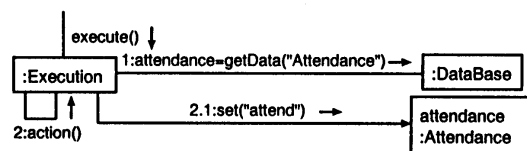


図 4: 実行の流れの例

条件部は, 図 3 に示すように, データベースからスケジュール帳 (timeTable), 日付 (date) の参照を受け取り (図 3 のメッセージ 1, 2), 照合を行い (3), スケジュー

ルが空いていれば true を返す (3.1)。実行部は, 図 4 に示すように, データベースから出欠席 (attendance) の参照を受け取り (1), 実行を行い (2), 参加するという意思を登録する (2.1)。

3 スケジュール調整エージェントシステムへの適用

2 節で提案したエージェントモデルをスケジュール調整エージェントシステムに適用した。会議予約において, 参加者エージェントは日程の候補に対して参加する/参加しない/保留するの意思を返信し, 主催者エージェントは, 参加者の出席状況によってどの日程で予約するかを決定する。

ユーザは, 様々なルールを図 5 に示すルールエディタを用いて, エージェントに与える。“スケジュールが空いている”などの条件文および“参加する”, “参加しない”などの実行文をユーザが組み合わせてルールを生成し登録する。

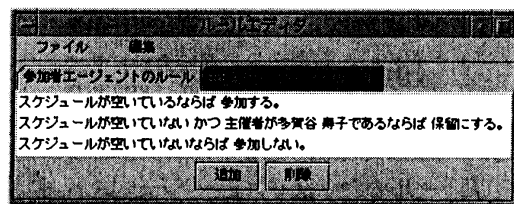


図 5: ルールエディタ

ルールをオブジェクトとして実現することにより, 図 5 の 3 番目のルールを“スケジュールが空いていないならば 保留にする。”のように変更したり, 動的に新しいルールを追加することができた。

4 おわりに

本稿では, ソフトウェアエージェントにルールとしてユーザの知識を与えるエージェントモデルを提案し, その実現方式と適用結果について述べた。

参考文献

- [1] 西田 豊明: ネットワーク社会とエージェント—擬人化された人工システム— 情報処理学会誌, Vol.38, No.1, pp.10-16(1997).
- [2] 中所 武司: ソフトウェア危機とプログラミングパラダイム—“わかりやすさ”の追求—, 啓学出版 (1992).