

協調プロトコル記述言語 AgenTalk の機能拡張の検討

3Bb-1 浅山和典 篠原拓嗣 桑原和宏[†] 大坪基秀[†] 西村俊和 石田 亨

京都大学工学部情報工学教室

[†]NTT コミュニケーション科学研究所

[†]NEC C&C 研究所

1 はじめに

マルチエージェントシステムのプロトコル記述言語として提案している AgenTalk[1] は、KQML や Cool とは異なり、通信網プロトコル記述に用いられてきた拡張状態遷移機械に基づくアプローチを採っている。

AgenTalk では個々のプロトコルを拡張状態遷移機械として記述し、これをスクリプト (script) と呼ぶ。ひとつのスクリプトは、状態、初期状態、状態遷移規則と遷移動作、変数、及びスクリプトにローカルな関数の組によって構成される。

ただし、利用者がプロトコルを容易に記述、蓄積、再利用できるよう、プロトコル間の継承機能を導入している [2]。継承の対象はスクリプトの構成要素全てで、状態とその状態からの遷移規則を継承の最小単位としている。この機能により、分散人工知能の分野で提案されていたマルチステージネゴシエーションプロトコルを、契約ネットプロトコルの拡張として記述するのに成功している。

本報告ではこの AgenTalk の継承機能を利用して、今後需要が増えると思われるネットワーク内での予約スクリプトの記述に AgenTalk を適用し、その可能性と問題点を検討し、言語機能拡張の方向を探る。

2 AgenTalk によるスクリプト記述実験

予約スクリプトの記述へ AgenTalk を適用する際の問題点を明らかにするために複数の記述実験をおこなった。

2.1 旅行代理店スクリプト

京都大学では、ホテルの予約を題材にして記述実験をおこなった。旅行代理店と顧客が対話によってホテルを決定し予約をおこなう際の、旅行代理店のスクリプトを記述する。図 1 がその記述例である。このスクリプトでは一連の動作を状態遷移として記述している。

また、このスクリプトに従い顧客の入力を受け付ける WWW インタフェースを記述した。顧客は画面上の選択肢から受たいサービスを選ぶ。プログラムは

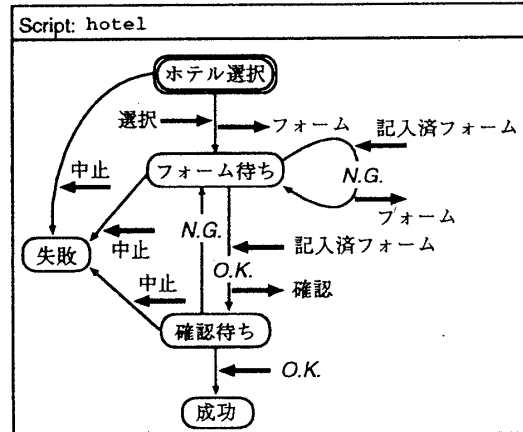


図 1: 旅行代理店スクリプト

Perl で記述してあるがその動作は拡張状態遷移機械と同等である。表示される一つの画面が状態遷移機械の一つの状態に、顧客が選ぶサービスが遷移条件と遷移動作に、それぞれ対応する。Perl プログラム内部では、各サービスに対応する手続きが呼ばれ、必要な処理 (遷移動作) をすると共に次の画面を生成 (状態遷移) する。

2.2 知識問い合わせスクリプト

NEC では、知識問い合わせスクリプトを AgenTalk で記述した。これは、例えば利用者が旅行の手配をエージェントに指示した場合に、そのエージェントが持つ知識で解決できなければ、ネットワーク上の他のエージェントから知識を収集し解決を試みるというものである。

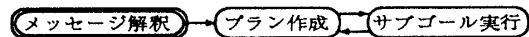


図 2: 知識問い合わせスクリプト

このスクリプトは、タスク要求メッセージの解釈、プランの作成、サブゴールの実行、の 3 状態からなる (図 2)。推論機構はサブゴールの実行に失敗するとバックトラックして別のプランを作成する。

2.3 AgenTalk の問題点

記述実験を通して AgenTalk の問題点が明らかとなり、次のような機能が必要であることが分かった。

Extending the AgenTalk Coordination Protocol Description Language

Kazunori Asayama, Takuji Shinohara, Kazuhiro Kuwabara[†], Motohide Otsubo[†], Toshikazu Nishimura, Toru Ishida
Department of Information Science, Kyoto University
Kyoto 606-01, Japan

[†]NTT Communication Science Laboratories

[†]NEC C&C Research Laboratories.

ルール記述 いずれの記述例でも他の状態への遷移よりも同じ状態に留まることの方が多い。また記述もルールに重点がおかれている。ホテル予約(図1)の例では、その中心であるホテルを決定するまでのプロセスは一つの「ホテル選択」という状態の中でおこわれている。この例では AgenTalk の中心的機能の状態遷移モデルを活用できていない。これから、予約スクリプトの記述言語にはルール記述も必要になってくると考えられる。手続き的なスクリプト呼び出し 既存のスクリプトを再利用する方式としては継承機能のみでは不十分で、手続き的な起動を可能とする必要がある。あるスクリプトから他のスクリプトを起動し、終了を待って次の動作に移ったり実行結果によって次の動作を決定したりできれば、プロトコルの部品ライブラリの構築ができる。

ルール単位での継承 AgenTalk では継承は状態単位でおこなわれるため、状態遷移規則の一つをサブクラスで再定義する場合にもその状態全体を定義し直す必要が生じる。状態遷移規則(ルール)単位での継承である。

3 AgenTalk へのルール記述機能の導入

上記の問題点を解決するために、言語機能の拡張の検討を始めている。以下ではまず、ルールの集合とその発火条件を記述し、条件照合によってループを起動する、というルール記述中心のスクリプトの導入を考える。

ルール記述では、プロトコルの進行に関する情報は発火条件に属する変数のみで表す。予め全ての動作を有限状態機械によって指定するのではないため、柔軟な動作順序を実現するスクリプトの記述も可能になる。また、ルール間に順序の制約がないため、実行順序を問わない動作の記述も容易になる。

一方、ルール記述のみからなるスクリプトは、ルール発火の制御を予め指定することが困難で、予期せぬルールが発火しないようにするには発火条件を詳細に定義する必要がある。従って、ルール記述と状態遷移記述を統合して、言語機能として導入することが必要である。

そこで、AgenTalk の拡張として、(1) ルールの集合で状態を構成することにし、動作の大枠を状態遷移記述によって指定する、また(2) 具体的な動作の詳細をルールによって記述可能とする。以下にその詳細を示す。

状態の定義 ルールの集合に対して一つの状態を割り当てる。同じ状態に留まり、対話型で情報を検索するような動作記述には、各動作をルールで表現し、状態遷移をそのルール集合の切り替えとするほうが適している。現状の AgenTalk では、ルール記述のための言語機能が十分とは言えない。

継承 AgenTalk では継承の最小単位が状態であるため、ルール集合全体が継承の単位となる。僅かでも異なるルールを記述したければ継承機能を利用できず、全て

のルールを再定義する必要がある。そこで、継承はルール単位で行なうこととする。ある状態に属するルールの集合に対して、ルールの追加・変更をルール単位で行なえるように拡張し、予約スクリプトのサービスの追加や変更を継承によって容易に実現できるようにする。

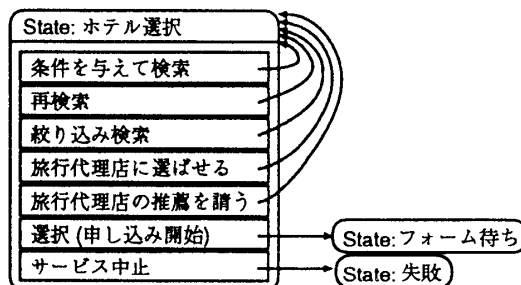


図3: ホテル選択状態を構成するルール集合

```
(define-rule-script hotel ())
(define-action (search-by-condition hotel)
 :precondition
 ((msg search :method "condition"))
 :action ((! search-hotel msg)))
(define-action (start-reservation hotel)
 :precondition ((msg reserve))
 :action ((call-script 'hotel-reservation)))
```

図4: ルール記述の例

上記の方針に基づき、AgenTalk の言語仕様になんたなマクロを導入する形でルール記述によるスクリプトを実装した。これは、現在の仕様で提供されていたスクリプト記述機能を、そのまま継続して使用可能とするためである。図1に示した旅行代理店スクリプトのホテル選択状態を構成するルール集合を図3に示す。また、その記述例(一部)を図4に示す。

4 おわりに

予約スクリプトなどの記述実験によって、ルールの集合を単位として状態を構成し、状態遷移を記述する方式が向いている場面を見出すことができた。このような記述には、現在の AgenTalk の言語機能は十分でなく、ルール記述を導入する必要があると考えている。なお、AgenTalk は <http://www.cslab.tas.ntt.jp/at/> からフリーソフトとして取得できる。

参考文献

- [1] 桑原 和宏, 篠原 拓嗣, 大里 延康, 石田 亨, “協調プロトコル記述言語 AgenTalk の実現,” 信学技報, AI-95-18, 1995.
- [2] K.Kuwabara, T.Ishida and N.Osato, “AgenTalk: Describing Multiagent Coordination Protocols with Inheritance,” *TAI-95*, p.460-p.465, 1995.