

能動データベースによる情報エージェントの協調制約

4K-6

木内 光 中田 優作 小西 修

高知大学理学部数理情報科学科

780 高知県高知市曙町 2-5-1

{hikiuchi, yunakata, konishi}@is.kochi-u.ac.jp

1 はじめに

今日、インターネットの普及に伴いネットワーク分散環境に多くの情報源が存在している。しかし、これらの異種分散情報源は半構造データなども含んでおり、データ定義が不明瞭である。この状況下で利用者が目的の情報を適切に探し出すことは非常に困難である。こういった問題を解決する一つの方法として、我々はアクティブ・メディアーション・システム HI-AMS を開発している [2]。HI-AMS は、能動データベースをプラットフォームとしており、ユーザ・インタフェース・エージェント (UI エージェント、質問・検索法)、アクセス・エージェント (A エージェント、アクセス・集約)、及び情報統合、評価エージェント (情報解析・学習・探索戦略)、そして各エージェントを調停し、協調促進するためのメディアータ・エージェント (M エージェント) から構成されるエージェント・ベースの仲介者システムである [1]。このアーキテクチャの特徴として、M エージェントが情報源を選択すること、新しい情報源の開拓と登録ができること、そしてエージェントの追加というスケラビリティが挙げられる。このシステムはダイナミックかつオープンな環境に対応、スケラビリティ、情報エージェント、そして能動データベースといった考えに基づいて開発されている。本稿ではシステム内部のエージェント協調に着目する。エージェント間の通信は要求とその要求の採否と見ることができる。ここに許可・禁止・義務の判断、すなわち義務論理を入れることがエージェント間の協調促進に有効であることとを証明する。

2 義務論理

義務論理とは様相論理の一つであり、「しなければならぬ (義務)」、「してはいけない (禁止)」、「しても良い (許可)」の3つの状態をモデル化したものである。義務論理モデルの3つの状態を HI-AMS の各エージェントの状態を表すものとして用いる。この状態での義務論理モデルでは、エージェントの初期状態は「許可」であり、処理依頼のメッセージを受け取ると「義務」状態に移行し、処理の主体も移される。処理が終了すると、依頼主に処理結果等のメッセージを返して「許可」状態に移行し、初期状態に戻る。もし、エージェントが処理依頼のメッセージを受け、「義務」状態に移行したが依頼された処理に失敗した場合や、一貫性制約を破る可能性が存在する場合は「禁止」状態にして、それ以後の処理依頼の受付や行動を行わないようにする。問題解決後、M エージェントが「禁止」状態から「可能」状態に戻し、初期状態に戻る。エージェントの要求間の関係はメッセージ・トランザ

クションとして捉えられ [3]、エージェントからの要求に対して、必ず、その対応エージェントは答えなければならないとする。例えば、エージェント A からエージェント B に要求がある場合、次のように記述される。

obligation(has_request(B) ---> has_response(A))

これは、もし B が request を持つならば、A は response を持たねばならないということを述べている。ここに、エージェント B の has_request(B) は、必ず A から B への request という事実があるということが前提になり、次のように記述される。

[AtoB(request)]has_request(B)

これは A が B へ request を送るという事実は、B が request を持つということに帰着する。

3 エージェント・トランザクション

HI-AMS を構成する各エージェントは、ユーザからの質問に答えるために互いに協調しながら作業を進める。このとき、各エージェントの調停を行い、協調を促進させる役割を担う M エージェントは、各エージェントの状態を管理する。各エージェントは何らかの状態変化が起こる度に、M エージェントに自分の状態を通知する。これにより、各エージェントはメディアータを介してより豊かなエージェント間の協調が行える。具体的にエージェント群の振る舞いの簡単な流れを示す。

- (1) 情報源エージェントは、ドメイン制約も含めたドメイン情報をメディアータに通知する。
- (2) UI エージェントからの質問が、A エージェントに送られる。
- (3) A エージェントは、その質問を M エージェントに送る。
- (4) M エージェントは、各ドメイン情報と質問マッチングをとり、最適の情報源エージェントを A エージェントに知らせる。
- (5) A エージェントは、その情報源エージェントに質問を発生し、結果を受け取る。

ここに、エージェント群の振る舞いは次の2点で制約される。

- (1) エージェントは、自己の外へのアクションについては、必ず M エージェントに問い合わせなければならない。M エージェントは、それに必ず答えなければならない。また、メディアータは問い合わせという事実がないのに応答することは禁止される。
- (2) エージェントは、M エージェントから紹介された他のエージェントへのみ、要求することを許可される。

Cooperative Agents on Top of the Active database

Hikaru Kiuchi

Dept. of Information Science, Kochi University

2-5-1 Akebono-cho, Kochi 780 Japan

また、M エージェントは能動データベースのフロントエンドに存在して各エージェントの状態をテーブル上に記述して管理する。この状態を管理するために二つのテーブルを用意している。テーブルの一つは義務論理のためのものであり、次のようなテーブルによってエージェントの状態を把握することができる。

```
agent [name,state]
```

もう一つのテーブルはエージェントの動向を逐一記述するための、いわゆるログであり次のようなテーブルである。

```
agent_log [name,operation,from,to,ope_state]
```

このテーブルはエージェント name において、処理 operation をエージェント from から依頼された、またはエージェント to へ依頼して、その処理の状態が ope_state であることを表現している。このエージェントの状態を管理するテーブルと能動データベースの ECA ルール機構によって、義務論理における一貫性制約が表現できる。本研究で実際に実装したルールの例を次に挙げる。

```
trigger ud_ua_state
Event :insert on agent_log
Action:after(execute procedure state_ud_pl())
```

```
procedure:state_ud_pl()
select agent_log.name,agent_log.operation,
agent_log.from,agent_log.to
into n,ope,fr,t from agent_log
where agent_log.name = 'UIAgent' and
agent_log.operation = 'send-query' and
agent_log.ope_state = 'start';
select agent.state
into to_s from agent where agent.name = t;
select agent.state
into pre_ope_s from agent
where agent_log.name = 'UIAgent' and
agent_log.operation = 'get-query';
if pre_ope_s != 'commit' then
update agent
set state = 'KINSHI' where agent.name = 'UIAgent';
elif to_s = 'KYOKA' then
update agent
set state = 'KINSHI' where agent.name = 'UIAgent';
end if;
```

UI エージェントがユーザから受けた要求を A エージェントに送信する以前に、その旨を M エージェントに伝える。そのとき、上述のルールは起動するものである。UI エージェントは A エージェントにメッセージを送る以前にユーザから要求を受けていないといけなく、A エージェントが「許可」状態でなければメッセージを送信することはできない。これらの制約が破られると UI エージェントの状態は「禁止」状態となり、その後の行動を禁止する。

4 実験

本研究では、HI-AMS において各エージェントの状態管理を担う能動データベースとして Infomix 社の Universal Server を用い、エージェントとしては M エージェント、UI エージェント、A エージェントを使用し、これらの通信間の協調を実験した。全てのエージェントは米国スタンフォード大学機械工学科で開発中の Java Agent Template を利用しており、これらはエージェント間知識操作言語である KQML によって通信を行う。UI エー

ントも A エージェントも最初の初期状態は「許可」状態であり、他のエージェントからのメッセージを受けられる状態になっている(図 1)。ユーザから要求を得ると、UI エージェントの状態は「義務」状態へと移行する。M エージェントは各エージェントからメッセージを受け取る度に、エージェントのログテーブルへ書き込んでいき(図 2)、エージェントの状態は 3 章で示したようなルールによって変化していく。M エージェントが受けたメッセージへの返答は、常に agent テーブルと agent_log テーブルを参照することによって要求している操作が commit なのか abort なのかを答える。M エージェントは、ope_state の状態と、各エージェント自身の状態を参照しながらエージェント間の協調促進を行う(図 3)。

name	state
UIAgent	KYOKA
AAgent	KYOKA

図 1 各エージェントの初期状態

name	operation	from	to	ope_state
UIAgent	get_query	user	UIAgent	commit
UIAgent	send_query	AAgent	UIAgent	start

図 2 各エージェントのログ (1)

name	operation	from	to	ope_state
UIAgent	get_query	user	UIAgent	commit
UIAgent	send_query	AAgent	UIAgent	commit
AAgent	get_query	UIAgent	AAgent	commit
AAgent	ask_url	AAgent	MAgent	start

図 3 各エージェントのログ (2)

5 おわりに

本稿では、HI-AMS におけるエージェント間の協調に義務論理を用いることの有効性を実証した。能動データベースの特徴である ECA ルール機構によって義務論理における制約を表現し、これによってエージェント間のより柔軟な協調促進が可能になることを実験によって確認した。今後、情報源エージェントと HI-AMS の間での義務論理導入と、HI-AMS 内の各エージェントがエージェント群を形成した場合の協調促進について検討する。

参考文献

- [1] Bayardo Jr. R.J. and et al., "InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments," SIGMOD'97, pp.195-206, May 1997.
- [2] 小西 修, "異種情報源統合のためのアクティブ・メディエーション・システム-HI-AMS:High Intelligent - Active Mediation System-", Mem.Fac.Sci.Kochi Univ.(Inform.Sci), 17, March 1997.
- [3] H.Weigand, et al., "Interoperable Transactions in Business Models - A Structured Approach", Advanced Information Systems Engineering, LNCS 1080, pp. 193-209, 1996.