

修士論文

拡張BDIアーキテクチャに基づく
合理的エージェントの設計

五十嵐 新女

2001年1月31日

奈良女子大学大学院
人間文化研究科 情報科学専攻

拡張 BDI アーキテクチャに基づく 合理的エージェントの設計*

五十嵐 新女

概略

本論文では、マルチエージェント環境において、権利委譲を受けた人間の目標を達成することを主眼において、人間または他のエージェントと協調して問題解決を行う合理的エージェントの実現方法を提案する。我々の目標は、コミュニティウェア構築技術の開発とその構築ツールの提供である。コミュニティウェア構築技術の課題として取り上げる合理的エージェントは、人間から権利委譲された仕事に対して、与えられた目標を達成するために、まず、自分の信念を用い熟考してプランを選択し、選択したプランから、更なる熟考を引き起こすような心的状態として、意図を形成する。その後、どうにかして意図を持続しようとするコミットメントの支配の下に、意図の前提条件を満たす時が来れば、その意図を遂行する。本論文では、この信念、目標、意図に基づく合理的エージェントに適用するエージェント内部アーキテクチャの基礎に、Rao と Georgeff の BDI アーキテクチャを採用する。その上で、エージェント間のコミュニケーション実現に関する BDI アーキテクチャの問題点を挙げ (1) エージェント間のコミュニケーションをプラン本体に自然に表現できるプラン (2) プランに、あるプランを将来実行するという未来指向的意図の形成条件の追加 (3) 意図の持続性のタイプを決定するコミットメント戦略をエージェントの置かれた状況や心的状態に応じて、動的に選択できる意図の持続機能、などの拡張を提案し、実装方法について述べる。これら拡張 BDI アーキテクチャの有効性を応用事例で示す。

*奈良女子大学大学院 人間文化研究科 情報科学専攻 修士論文, 2001 年 1 月 31 日.

Design of Rational Agents and Extended BDI Architecture *

Yoshime IGARASHI

Abstract

In this paper, we propose an extended belief-desire-intention(BDI) architecture, on which rational agents can interact and communicate with other agents – both human and artificial – in multi-agent environments. Our purpose is to implement the rational agents on the BDI architecture proposed by Rao and Georgeff based on the Bratman's theory of intention.

According to this theory, we can describe this type of rational agent as the following: – the rational agent deliberately selects plans by using their own beliefs to attain given goals, forms intentions from the selected plans as states of mind that constrain future deliberation, and performs speech acts with an intention when the precondition of the intention is satisfied.

First, we propose the extended BDI architecture, Then, we will show how effective it is by demonstrating a case based on such an BDI system that was extended: (1) to naturally describe communications with agents in the form of the action tree within the plan itself. (2) to add conditions of agents' formalization of future-directed intention to a plan and (3) to enable agents to dynamically decide on a commitment strategy depending on the situation and the mental state in which they are.

*Master's Thesis, Department of Computer Science, Graduate School of Human Culture, Nara Women's University, January 31, 2001.

目次

1	はじめに	3
1.1	背景と目的	3
1.2	本論文の構成	5
2	BDI アーキテクチャ	6
2.1	BDI Logic	6
2.2	BDI アーキテクチャ	7
3	BDI アーキテクチャの問題点	9
3.1	プランの問題点	9
3.2	コミットメント戦略の問題点	10
4	拡張 BDI アーキテクチャ	10
4.1	プラン本体の拡張	11
4.2	意図形成条件の追加	15
4.3	動的コミットメント戦略	15
4.4	拡張 BDI アーキテクチャの基本設計	16
5	拡張 BDI アーキテクチャの実装方法	20
5.1	拡張 BDI アーキテクチャの内部構成	20
5.2	環境知覚器・行為実行器	20
5.3	BDI 部	22
5.4	実装の現状と今後の課題	30
6	応用事例と検証	31
6.1	研究所見学案内システム	31
6.2	意図的な合理的エージェントの基本設計	33
6.3	FIPA	35
6.4	各エージェント間のコミュニケーション	36
6.5	研究者秘書エージェントの情報フィルタリング事例	37

7	考察と課題	41
7.1	拡張 BDI アーキテクチャの実装	41
7.2	心的状態の整合性	41
7.3	意図形成条件を用いた対話プランの推論	42
7.4	動的コミットメント戦略を用いた協調	42
8	おわりに	43
9	謝辞	44

目 次

1	契約ネットプロトコルのサブプラン例	12
2	拡張 BDI アーキテクチャ	14
3	意図スタックと意図フレーム	17
4	拡張 BDI アーキテクチャの内部構成	21
5	研究所見学案内システム	32
6	エージェント間のコミュニケーション	38
7	秘書エージェントの心的状態	49
8	秘書エージェントのプランライブラリ	50
9	C-MAP の概要	51

1 はじめに

1.1 背景と目的

インターネットに代表される世界規模のコンピュータネットワークやモバイルコンピューティング技術の発展に伴い，コミュニティウェア [1, 2, 3]，つまり，広く分散した人々の間でのコミュニティ形成，興味や知識の共有，社会的活動等を支援するためのシステムの研究の必要性が認識されはじめている．角ら [4] は，C - M A P と呼ぶ実世界コンテキストを利用したモバイルアシスタントの実現を目指したプロジェクトにおいて，コミュニティウェアの具体例を示しつつその方向性を提案している．

そこで，我々は，コミュニティウェアの構築技術の開発を目指し，

- (1) ヘルパーエージェント¹ [5]
- (2) 合理的エージェント [6, 7, 10]
- (3) エージェント間のコミュニケーション [11, 12]
- (4) 秘書エージェント² の情報フィルタリング
- (5) パーソナルエージェントの情報検索と学習
- (6) 秘書エージェントの丁寧な要求拒否応答

などを課題として取り上げ，図 5 の研究所見学案内システムをテストベッドとして，これら課題に対する実現手法の提案に取り組んでいる [13, 14, 15, 16, 19, 20] ．

課題 (1) に対して，利用者とエージェントの対話を仲介し自然言語対話をそのエージェントと分散処理するヘルパーエージェントを提案した．仲介されたエージェントの対話プランは，FIPA [31] に準拠した基本通信行為とプロトコルなどの言語行為 [28] から構成され，プロダクションシステムの前提条件を起動条件と

¹コミュニティウェアの一つとして，複数の人間と対話能力を持ち人間を助けるという社会的役割を持つエージェント．

²我々は，personal agent には 2 つのタイプがあると捉え，利用者の意図を実現するために利用者の個人情報積極的に活用してサービスを行うエージェントは“パーソナルエージェント”，権利委譲を受けた人間の目標を達成することに主眼において利用者の個人情報に依存したサービスを行うエージェントは“秘書エージェント”と，それぞれ呼ぶ．本論文は秘書エージェントに焦点を当てる．

したスクリプトを用いて記述することを提案した [13] . 現在 , その対話制御について実装を終了している [16] .

しかし , 動的に変化する状況を不完全にしか捉えられない環境下では前提条件のみで未来を完全に予測するのは不可能である . Bratman [21] の意図の理論によれば , 人間は , そのような環境下でも目標を達成するために , まず , 熟考して選択したプランを意図として形成する . その後 , その意図の支配の下に , そのプランを実施するための実践的な手段やサブプランを推論したり , その時が来ればそれらプランを意図的に実行して目標を達成しようと努力する . Rao ら [23] はこの意図の理論を熟考型エージェントの内部アーキテクチャに適用した BDI アーキテクチャを提案している . 本論文は , 権利委譲を受けた人間の目標を達成することを主眼において , 人間または他のエージェントと協調して問題解決を行う意図的な合理的エージェントの実現に BDI アーキテクチャを用いる場合 , Rao と Georgeff が提案した BDI アーキテクチャの問題点を挙げ (1) エージェント間のコミュニケーションを自然に表現できるプラン (2) あるプランを将来実行するという未来指向的意図の形成条件 (3) エージェントの置かれた状況や心的状態に応じて , 意図の持続性のタイプを決定するコミットメント戦略を動的に選択できる意図の持続機能 , などを提案し , これら拡張 BDI アーキテクチャの実現手法を提案する .

具体的には , BDI アーキテクチャのプラン本体は , 3.1 節で述べるように , エージェント間のコミュニケーションを自然に記述することが出来ない . そこで , 言語行為 [28] (たとえば , FIPA [31]) を用いて遂行されるエージェント間のコミュニケーションにおいて発生する条件分岐やメッセージを待つ同期などを表現できるようにプラン本体の拡張を提案する . また , プランには , 状況や心的状態によって異なる , 同じ目標を達成するようなプランの部分集合を用意した方が良い場合が考えられる . そこで , プランのスロットに (未来指向的) 意図形成条件の追加を提案する .

また , Rao と Georgeff が BDI logic を用いて形式化した 3 種類の *blind* , *single-mind* , *open-mind* などのコミットメント戦略は , エージェントの意図の持続性のタイプとして「エージェント単位に設定する」場合だけでなく , 3.2 節で述べるように , 各エージェントの心的状態や自らの置かれた状況に応じて動的に選択した方が良い場合が考えられる . そこで , 自らの置かれた状況や心的状態に応じて , コミットメント戦略を動的に選択できるよう BDI アーキテクチャの拡張を

提案する．次に，コミュニティウェア構築ツールの提供を目標に，提案する拡張 BDI アーキテクチャの実装方法について述べる．最後に，研究所見学案内システムにおける秘書エージェントの情報フィルタリング事例を用いて，この提案手法が，マルチエージェント環境において人間または他のエージェントと協調して問題解決を行うために，対話プランを意図として形成し，時が来ればその対話プランを構成する言語行為を意図的に遂行する合理的エージェントの基本設計に有効であることを示す．現在，動的コミットメント戦略の処理を残し，拡張 BDI アーキテクチャの実装は終了している．研究所見学案内システム他，これを適用したシステムの構築も概ね問題なく進行している．

1.2 本論文の構成

本論文は，2 章で BDI Logic と BDI アーキテクチャを概説する．3 章で BDI アーキテクチャの問題点を述べる．4 章で，前章の問題を解決するための拡張 BDI アーキテクチャを提案する．5 章で合理的エージェント構築ツールとして拡張 BDI アーキテクチャの実装方法を述べる．6 章で提案手法の有効性を示すために意図的な合理的エージェントの事例を述べる．7 章で提案手法に対する考察と今後の課題を述べる．最後に 8 章でまとめる．なお，本論文を読むに当っては一階述語論理と時相論理の初歩的な知識を仮定している．

2 BDIアーキテクチャ

本章では、BDI Logic と BDI アーキテクチャをそれぞれ概説する。

2.1 BDI Logic

BDI logic [24, 32] は、エージェントの過去から未来に至る信念・目標（願望）・意図などの心的状態や行為などを自然に記述することができる時相論理体系の一つである。分岐時間命題時相論理の一つである CTL* [34] に、信念・目標・意図といった心的状態を表す様相オペレータ BEL, GOAL, INTEND を導入し、さらに、述語論理に拡張した体系である。時相オペレータとしては X (nexttime), F (sometime), U (until), E (ある未来で), A (あらゆる未来で) などを持つ。Rao と Georgeff は Bratman [21] が分析した心的状態の整合性に関する公理や意図の持続性のタイプに関するコミットメント戦略などを、以下のように BDI logic を用いて自然に表現している [24]。 ϕ は BDI logic の論理式一般、 α は [24] で導入される O-formula、 e はイベントである。

- 心的状態の整合性公理

- (A1) $\text{GOAL}(\alpha) \supset \text{BEL}(\alpha)$
- (A2) $\text{INTEND}(\alpha) \supset \text{GOAL}(\alpha)$
- (A3) $\text{INTEND}(\text{does}(e)) \supset \text{does}(e)$
- (A4) $\text{INTEND}(\phi) \supset \text{BEL}(\text{INTEND}(\phi))$
- (A5) $\text{GOAL}(\phi) \supset \text{BEL}(\text{GOAL}(\phi))$
- (A6) $\text{INTEND}(\phi) \supset \text{GOAL}(\text{INTEND}(\phi))$
- (A7) $\text{done}(e) \supset \text{BEL}(\text{done}(e))$
- (A8) $\text{INTEND}(\phi) \supset \text{AF}(\neg \text{INTEND}(\phi))$

- コミットメント戦略

- (A9A) $\text{INTEND}(\text{AF } \phi) \supset$
 $\text{A}(\text{INTEND}(\text{AF } \phi) \text{ U } \text{BEL}(\phi))$

(A9B) INTEND(AF ϕ) \supset
A(INTEND(AF ϕ) U (BEL(ϕ) \vee \neg BEL(EF ϕ)))

(A9C) INTEND(AF ϕ) \supset
A(INTEND(AF ϕ) U (BEL(ϕ) \vee \neg GOAL(EF ϕ)))

たとえば, (A5) は, 直感的には「エージェントが ϕ という状態の達成が目標であるならば, エージェントは ϕ という状態の達成が目標であることを信じている」という意味である.

(A9A), (A9B), (A9C) は, それぞれ *blind*, *single-mind*, *open-mind* と呼ばれるコミットメント戦略である. 直感的には, それぞれ「エージェントが ϕ という状態をあらゆる未来でいつか達成するという意図を形成しているならば, あらゆる未来で, エージェントは ϕ という状態を達成したと信じるまでその意図を持続する」, 「エージェントが ϕ という状態をあらゆる未来でいつか達成するという意図を形成しているならば, あらゆる未来で, エージェントは ϕ という状態を達成した, または, ϕ という状態は達成不可能だと信じるまでその意図を持続する」, 「エージェントが ϕ という状態をあらゆる未来でいつか達成するという意図を形成しているならば, あらゆる未来で, エージェントは ϕ という状態を達成したと信じるか, または, ϕ という状態を目標から撤回するまでその意図を持続する」という意味である.

2.2 BDI アーキテクチャ

BDI アーキテクチャ [23] は, 限られた計算資源環境において動的に変化する環境を知覚し合理的に問題解決を行うために事前にプランを立てる熟考型エージェントの内部アーキテクチャである. エージェントの信念・目標・意図をそれぞれ陽に表現する動的なデータ構造 B, G, I やプランライブラリ, イベントキュー, およびそれらデータ構造を参照・更新するインタプリタから構成される.

インタプリタは以下のループ構造である.

```
initialize-state();  
do  
  options := option-generator(event-queue, B, G, I);  
  selected-option := deliberate(options, B, G, I);
```

```
update-intentions(selected-option,I);
execute(I);
get-new-external-events();
drop-successful-attitudes(B, G, I);
drop-impossible-attitudes(B, G, I);
until quit.
```

プランライブラリは、複雑なプランに関する信念を B から分離したもので、タイプ、連想 (invocation) 条件 (プラン形成条件)、前提条件 (プラン実行条件)、add/delete list (プラン実行の成功時に B に加除される信念)、本体 (実現手段) などをスロットとして持つプランの集合である。プラン本体は、基本行為 (実行可能な行為) とサブゴール (当プランの部分目標となるプラン) を辺に持つ木構造である。

3 BDI アーキテクチャの問題点

本章では、前章で概要を述べた BDI アーキテクチャのエージェント間のコミュニケーション実現に関する問題点を述べる。

3.1 プランの問題点

従来の BDI アーキテクチャ [23] のプラン本体には、基本行為とサブゴールしか記述できないため、これらの“条件分岐”³や他のエージェントからのメッセージを待つ“同期”などを表現することができない。たとえば、エージェント間のコミュニケーションをプランとして記述しようとした場合、4.1.1 で示す実例のように、交換されるメッセージを連想条件に持つプランを複数に分けて記述するしかなく、エージェント間の本来連続した言語行為の関連が不明瞭となるばかりでなく不自然な記述になるという問題点がある。

また、プランのスロットには、このプランを将来実行するという未来指向的意図の形成条件としては連想条件しかなく⁴、しかも、サブゴールではないトップレベルの目標の追加 (g-add) は外部イベントとして外部から与えられるか、または、心的状態として目標 G に記述されていることが仮定されている。このため、他のエージェントからのメッセージや自らの心的状態などの状況にマッチした目標や意図の形成をエージェント自身で決定するようなプラン記述ができない。このようなプランは、相手の (言語行為を含む) 行動から相手が遂行しようとしているプランを推論し、それに基づいて相手の行動の裏にある意図を読み取ることが必要となる状況に必要となると考えられる。特に、人間との自然言語対話システムなどでは、人間から明示的に目標が与えられることは少なく、文脈から対話プランの推論を行う必要があるが、現状の連想条件のみでは対応できない。

³従来のプラン本体は木構造であるが、CTL* の未来への時間分岐を表現しているに過ぎない。

⁴Rao らの BDI アーキテクチャは、まさに実行しようとしている、または、現在実行している現在指向的意図を主な対象としていると捉えている。そのため前提条件は現在指向的意図の実行条件と捉えている。

3.2 コミットメント戦略の問題点

Rao と Georgeff [24] が形式化した *blind* , *single-mind* , *open-mind* などのコミットメント戦略は、「エージェントのタイプとしてエージェント単位に設定する」場合だけでなく、各エージェントの心的状態や自らの置かれた状況に応じて動的に選択した方が良い場合が考えられる。

たとえば、彼らの用いた「喉の乾きを潤したい」という目標を達成する事例 [23] において、(1) マラソンをした直後の状況では、“すぐに”「喉の乾きを潤したい」ので、サブゴールとして「ソーダ水を飲む」というプランを *open-mind* なコミットメント戦略で持続する意図として形成する。この意図を実施するための実践的なサブプランとして、「冷蔵庫からソーダを取り出す」または「ソーダ水をコンビニまで買いに行く」を推論し、はじめに、「冷蔵庫からソーダを取り出す」を選択して実行したが、冷蔵庫にはソーダ水がないのでこのサブゴールは失敗する。次に、「ソーダ水をコンビニまで買いに行く」というサブプランは、“すぐに”「喉の乾きを潤したい」を満足しないので実行に移さず却下し、当初の「ソーダ水を飲む」というサブゴールを捨てる。次のサブゴールとして「水を飲む」というサブプランを *open-mind* なコミットメント戦略で持続する意図として形成し、その意図を実行するのが合理的である。

一方、(2) 恋人が訪ねてきている状況では、冷蔵庫にソーダ水がなければ「ソーダ水をコンビニまで買いに行く」というサブプランを実行して“必ず”「ソーダ水を飲む」という意図を持続させる *single-mind* なコミットメント戦略を選択するのが合理的である。

4 拡張 BDI アーキテクチャ

本章では、前章の BDI アーキテクチャの問題点を解決するために、プラン本体の拡張、意図形成条件の追加、および、動的コミットメント戦略を提案し、それらの実現手法として、BDI アーキテクチャをベースに変更・拡張した「拡張 BDI アーキテクチャ」の基本設計を記述する。なお、本論文では、Agent Communication Language として、信念、願望、意図に基づく言語行為として基本通信行為が規定された FIPA [31] を使用する。FIPA の詳細については 6.3 で述べる。

4.1 プラン本体の拡張

本節では、言語行為を用いて遂行されるエージェント間のコミュニケーションにおいて発生する条件分岐やメッセージを待つ同期などをプラン本体に表現できるように、プラン本体を、以下の「文」を辺とする木構造に拡張する。

```
文 ::= 単文 [ $\langle$ add/delete-list $\rangle$ ]  
    | { 文; 文; ... 文 }  
    | if 条件 then 文 [else 文]  
    | (待ち条件1 ?; 文1 | ... | 待ち条件n ?; 文n)  
単文 ::= 基本行為 | サブゴール  
待ち条件 ::= イベント
```

上記の文定義の最後の待ち条件などを記述する文は「待機文」と呼び、イベントキューに到着した他のエージェントからのメッセージなどの外部イベントまたは内部イベントと待ち条件_{1~n}の順に検査を行い、最初に真となった待ち条件_nに対応する文_nを選択して実行するという文である。

4.1.1 拡張プラン記述例

拡張したプラン本体の記述例として FIPA-contact-net [31] のサブプラン記述例を図 1 に示す。この例は、マネージャ M からの「タスク告知」に対して「入札」するサブプランである。このエージェントが「提案内容」を添付して「入札」を行い、マネージャ M からの「落札」を待つ。そして、マネージャが落札した場合で、その連絡があまりに遅いときは cancel する。そうでないときはマネージャに「終了報告」を行うプランである。また、落札されなかったときは *fail* して、このサブプランは失敗する。本論文では、FIPA の言語行為は：で始まる基本行為として記述する。

この事例を、従来の BDI アーキテクチャで記述すると、

- 「propose(M, 提案内容)」を本体にもち $\langle done(propose) \rangle$ を add list に持つプラン
- accept-proposal イベントが来てかつ前提条件 too-late が成立しているときに選択され、cancel を本体に持つプラン

```
Type: call-for-proposal  
Invocation: g-add(:propose(M, 提案内容))  
Precondition: true  
Add List: {done(提案内容)}  
Body:  
  :propose(M, 提案内容)(done(propose))  
  (:accept-proposal ?;  
    if too-late then cancel  
    else :inform(done(契約内容))  
  | :reject-proposal ?; fail)
```

図 1: 契約ネットプロトコルのサブプラン例

- 同様に, 前提条件 \rightarrow too-late が成立しているときに選択されて inform を本体に持つプラン
- reject-proposal イベントが来たとき選択されて *fail* を本体に持つプラン

の4つのプランに分けて記述する必要がある。これらが他のエージェントとの交渉を行う一連の言語行為であることは明示的に表現できないため, どのような意味があるのかもわかりにくい。これに対して, 拡張 BDI アーキテクチャを用いてプラン本体に記述するとエージェント間の相互作用を自然に表現することができる。

4.1.2 拡張 BDI アーキテクチャの実現方式

以下, 我々が提案する拡張 BDI アーキテクチャを図 2 に示し, プラン本体の拡張に伴う BDI アーキテクチャの実現方式を述べる。まず, BDI アーキテクチャの構成要素に, プラン本体の待機文を登録する「待ちリスト」を追加する。次に, 以下の 3 個の関数をそれぞれ次のように拡張または追加する。option-generator は待ちリストに登録されている待機文の待ち条件と単一化できる外部イベントが来れば, その文をプラン本体に持つ意図を次に実行する意図の候補に加える。update-intentions はプラン本体の各文を以下のように評価する。

- 単文 (add/delete list が付いている文を含む) の場合は, その文を次に実行する文として意図 I を更新する。
- 複文の場合は, その各文を再帰的に評価する。
- if 文の場合は, 条件の真偽に応じて, then 部/else 部のいずれかの文を再帰的に評価する。
- 待機文の場合は, 単一化した待ち条件に対応する文を再帰的に評価する。そして, 該当する待機文を待ちリストから削除する。また, 全ての待ち条件に単一化しない場合は何もしない。

最後に, 意図として形成されたプラン本体の次に実行する文が待機文の場合は, その待機文を一回だけ待ちリストに登録する。ただし, その意図が達成不能になっている場合は, 待ちリストから該当する待機文を削除する。

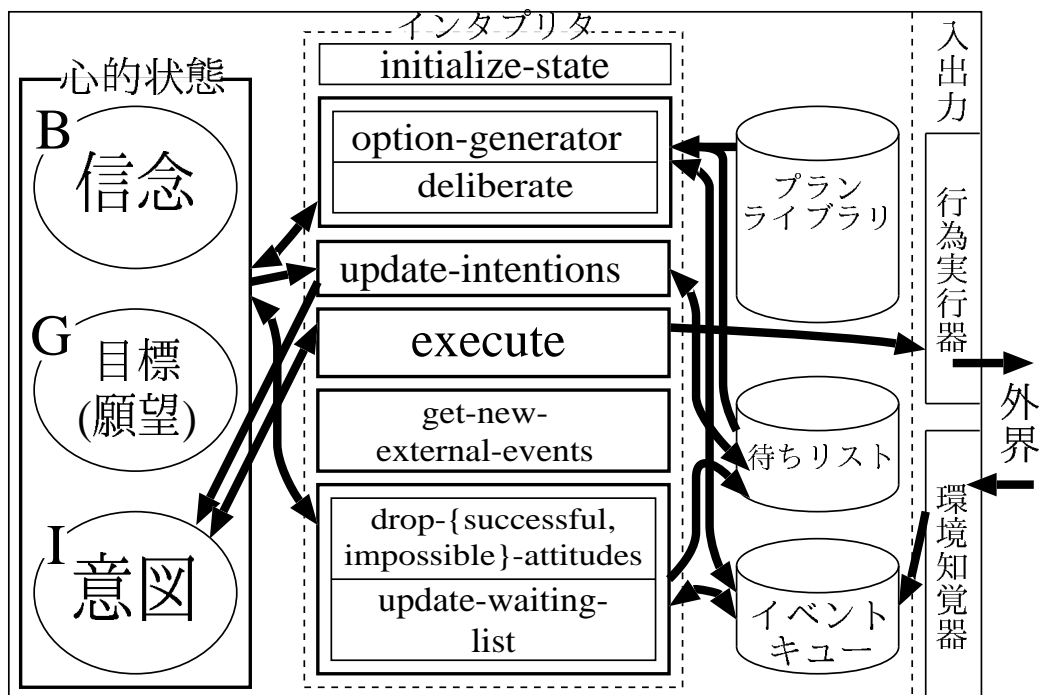


図 2: 拡張 BDI アーキテクチャ

4.2 意図形成条件の追加

他のエージェントからのメッセージ，自らの環境，および，心的状態などの状況にマッチする目標やその未来指向的意図をエージェント自身の判断で形成するために，プランのスロットに，意図形成条件 (FormCondition) を追加する．意図形成条件は，BDI logic を用いて表現し，意図形成の処理は 4.4 節の option-generator で行う．

4.3 動的コミットメント戦略

本節では，エージェントの置かれた状況や心的状態に応じて，意図の持続性のタイプを動的に決定するコミットメント戦略の実現方式を提案する．

4.3.1 コミットメント戦略の決定メカニズム

本論文では，エージェントの心的状態や自らが置かれた外部状況に応じて，形成する意図をどのコミットメント戦略で持続するのが合理的であるかという決定メカニズムについては今後の課題とする．たとえば，自然言語処理をベースとして，マラソンを終えた直後の状況では，“すぐに”「喉の乾きを潤したい」という目標の“すぐに”に注目して，この目標を達成するために形成する意図は blind で持続し，サブプラン (手段) は，“すぐに”実現できるサブゴールならどれでも良いと解釈して，open-mind で持続するように決定するようなメカニズムが考えられる．

4.3.2 戦略決定の単位と設定方式

コミットメント戦略は，プランおよびサブプラン単位に形成される意図を単位に決定する．そこで，図 3 のように，プランおよびサブプランに対して形成される意図スタックの中に，コミットメント戦略の属性を追加して，戦略を設定する．図 3 は，恋人が訪ねてきている状況で，冷蔵庫にソーダ水がなく，「ソーダ水をコンビニまで買いに行く」というサブプランを実行しようとしている例である．「ソーダ水を飲む」のコミットメント戦略には single-mind が，「ソーダ水をコン

「ビニまで買いに行く」のコミットメント戦略には `open-mind` が、それぞれ設定されている。

また、コミットメント戦略の持続の処理は、次節の `drop-successful-attitudes` , `drop-impossible-attitudes` において述べる。

4.4 拡張 BDI アーキテクチャの基本設計

本節では、以上に述べたような変更・拡張を施した「拡張 BDI アーキテクチャ」の基本設計を提案する。

心的状態 B, G, I やプランの意図形成条件、前提条件、`add/delete list`、および、プラン本体の `if` 文の条件などは BDI logic を用いて表現する。

拡張 BDI アーキテクチャのインタプリタは、以下のループ構造とする。

```
initialize-state();
do
  options := option-generator(event-queue,waiting-list, B, G, I);
  selected-option := deliberate(options,B, G, I);
  update-intentions(selected-option,B, G, I, waiting-list);
  execute(I);
  get-new-external-events();
  drop-successful-attitudes(B, G, I);
  drop-impossible-attitudes(B, G, I);
  update-waiting-list(I, waiting-list);
until quit.
```

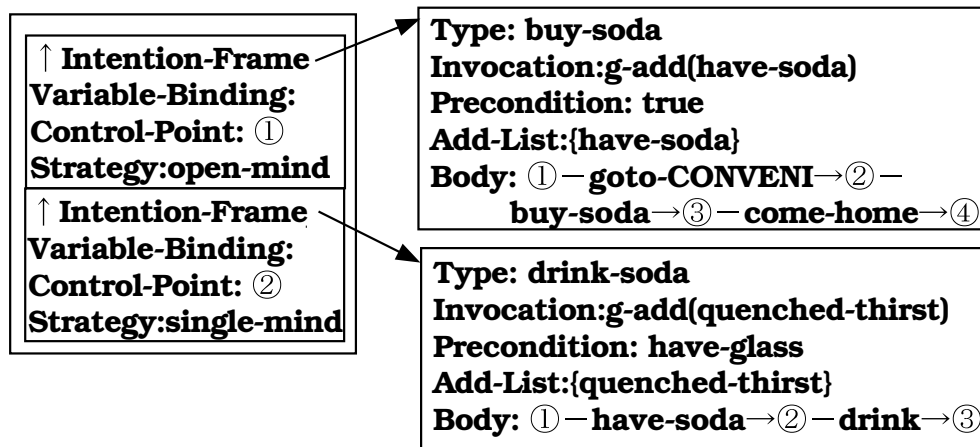
各機能は、2.1 節の心的状態の整合性公理を満足するように心的状態を更新する。ただし、信念管理については、TMS [35] または ATMS [36] を用いて行われるべきであるが今後の課題とする。

4.4.1 option-generator(意図の形成)

`option-generator` は、`event-queue` のイベントを解析して、`g-add` イベント（目標の追加）の場合は、その目標を G に追加する。そして、連想条件とマッチし、さらに、意図形成条件が B から真と推論できるプランを未来指向的意図として I に形成する。また、`waiting-list` に登録された待機文の待ち条件と単一化できるイ

意図スタック

意図フレーム



捨てられた意図フレーム

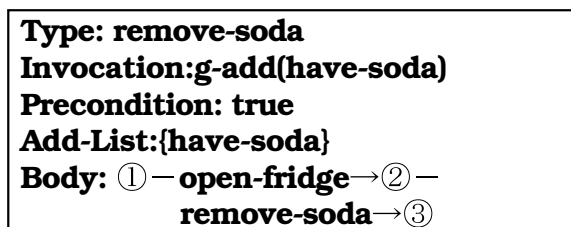


図 3: 意図スタックと意図フレーム

ベントの場合は、その文を本体に持つ意図を次に実行可能な現在指向的意図 [21] の候補として options リストに追加する。さらに、上記および既に形成されている未来指向的意図の前提条件が B から真と推論できる意図を options リストに追加する。

4.4.2 deliberate(意図の選択)

deliberate は、B, G, I やプランを選択するためのプランであるメタレベルプラン [23] を使って option-generator で選択された options リストの中から次に実行する意図を一つ selected-option に選択する。ただし、選択基準は個々のアプリケーションに依存する。

4.4.3 update-intentions(意図の更新)

update-intentions は、deliberate で選択された意図の本体から 4.1.2 節で述べた評価方法を用いて、次に実行すべき基本行為またはサブゴールを決定し、本体の次に実行すべき行為を指し示すように、その意図の制御情報を更新する。そして、waiting-list に登録された待機文の待ち条件を使用した場合は、その待機文を、waiting-list から消去する。また、次に実行すべき本体がサブゴールの場合は、このサブゴールを g-add 内部イベントとして event-queue に加え、このサブゴールが終了するまで、選択された意図を I 中の意図スタックにプッシュしておく。

また、update-intentions では、4.3.1 節および 4.3.2 節にて述べた動的コミットメント戦略の決定とそれに伴う意図の更新を行う。

4.4.4 execute(意図の実行)

execute は、update-intentions で更新された制御情報を使って、次に実行する意図の本体が基本行為であれば、その基本行為を実行する。サブゴールであれば何もしない。

4.4.5 get-new-external-events

get-new-external-events は、環境知覚器に、event-queue への外部イベントの書き込み命令を発行する。

4.4.6 drop-*-attitudes(心的状態の更新)

drop-successfull-attitudes および drop-impossible-attitudes は、達成・撤回したと信じる目標および対応する意図を、それぞれ G および I から取り除く。これは、意図 I に形成されている全ての意図に対して設定されているコミットメント戦略が 2.1 節の blind, single-mind, または, open-mind によって、それぞれ、 $BEL(\phi)$, $BEL(\phi) \vee \neg BEL(EF \phi)$, $BEL(\phi) \vee \neg GOAL(EF \phi)$ を証明することで、目標の達成・撤回を推論する。また、サブゴールが終了した時は update-intentions でプッシュした意図を意図スタックからポップする。

4.4.7 update-waiting-list

update-waiting-list は、4.1.2 節で述べたように待機文を waiting-lists へ登録する。

4.4.8 拡張アーキテクチャの動作例

4.1.1 節の例文を本体に持つプランを意図として実行する拡張アーキテクチャの一連の動作概要を述べる。まず、execute において、先頭の propose を実行して、次に実行する文を次の待機文にする。そして、update-waiting-list において一回だけ待機文を待ちリストに登録して応答を待つ。ただし、その意図が達成不能になっている場合は、待ちリストから該当する待機文を削除する。次に、待ちリストの待機文に対応する他のエージェントからのメッセージが外部イベントとして到着して option-generator と deliberate により選択されると、update-intentions が待機文を評価する。たとえば、accept-proposal が来ていれば、too-late の真偽を判定して cancel か inform を I に加え、それを execute で実行する。

5 拡張 BDI アーキテクチャの実装方法

本章では、コミュニティウェア構築技術として、合理的エージェント構築ツールを提供するために、4章で提案した拡張 BDI アーキテクチャの実装方法について述べる。これをアーキテクチャとするエージェントを、拡張 BDI エージェントと呼ぶ。

5.1 節で拡張 BDI アーキテクチャの内部構成、5.2 節で環境知覚器・行為実行器について、5.3 節で BDI 部について、それぞれ述べる。

5.1 拡張 BDI アーキテクチャの内部構成

拡張 BDI アーキテクチャ（以後、エージェントと略す）を BDI 部と環境知覚器・行為実行器で構成する（4 図）

エージェントへの入力イベントであり、出力は基本行為である。これらが具体的に何であるかは個々のエージェントやマルチエージェントシステムとして構築するシステムの仕様で定義する。例えば、人間と対話可能なエージェントの入（出）力には自然言語（音声入力やテキスト入力）が採用でき、マルチエージェント環境にて他エージェントと相互作用可能なエージェントの入出力には ACL（例えば、FIPA）が採用できる。

BDI 部は、環境知覚器・行為実行器から得た環境情報と心的状況に基づき、自分の行動を決定する。環境知覚器・行為実行器は、BDI 部らの命令に従って外界との入出力を行う。動的な環境の変化に対応するため、BDI 部と環境知覚器・行為実行器はそれぞれ並列に処理が行えるよう実装する。

以下に、各要素の意味付けと機能の具体的な実装方法について述べる。

5.2 環境知覚器・行為実行器

環境知覚器・行為実行器で行う処理は以下の 4 つである。

- (1) 外界からのメッセージを受信
- (2) BDI 部から命令があれば、受信したメッセージを、直接あるいは BDI 部での受信メッセージ形式に変換し、BDI 部へ出力

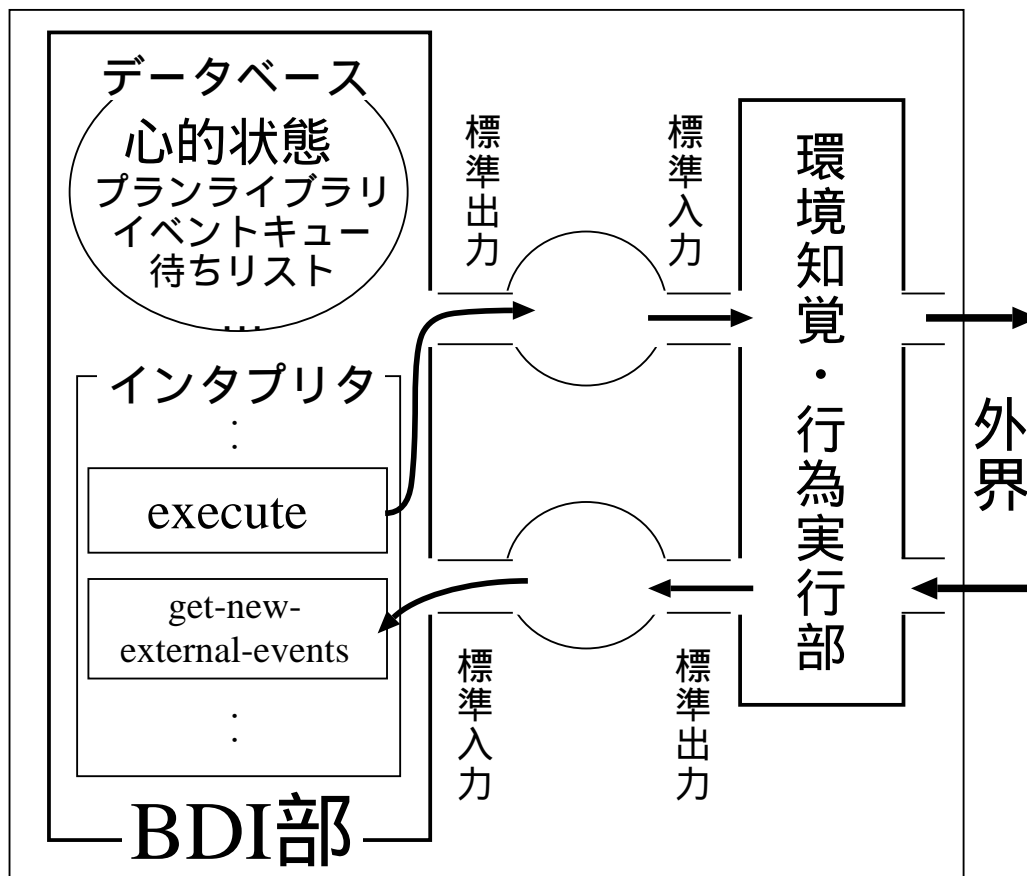


図 4: 拡張 BDI アーキテクチャの内部構成

(3) メッセージ ID の管理

(4) メッセージ形式の互換性を保つための変換フィルタ

(2) は BDI 部 と同期をとっていることを意味する．エージェント—外界間とエージェント内部（環境知覚器・行為実行器 — BDI 部間および BDI 部内）のメッセージ形式の違いを吸収できることから (4) を補う．例えば，エージェント間のコミュニケーションに XML を用いたメッセージを使用する場合，これを BDI 部 で使用しやすい形式（例えば，述語形式）に変換するためのフィルタを実装する．

5.3 BDI 部

本節では，BDI 部の実装環境，入出力，データの内部表現，インタプリタの実装方法について述べる．本節を読むにあたっては初歩的な Prolog プログラミングの知識を仮定する．また標準的な Prolog 系処理系で予め提供されている組み込み述語を本文中で断りなく使用する．

5.3.1 実装環境

実装環境には，Prolog 系処理系 CGAEA [33] を採用する．CGAEA はセルと呼ばれるデータ構造が複数扱え，多重世界が容易に記述できる．例えば，このエージェントを適用して複数のユーザと対話可能な自然言語処理システムを構築する場合，個々のユーザに関する信念を各セルに対応させれば，対話履歴や個人履歴の管理が容易に扱える．

5.3.2 入出力

環境知覚器・行為実行器 から BDI 部 への入力形式を以下に示す．

[*Message*, *Intention_id*]

- (1) *Message* : 述語形式
- (2) *Intention_id* : *Message* が待ち条件のメッセージである場合，該当する待機文を含む意図の ID を 環境知覚器・行為実行器 が追加する．

5.3.3 データの内部表現

以下で、信念、目標（願望）、プラン、オプション、意図について具体的な BDI 部内の内部形式を、意味付けと共に述べる（以下、単にこれを内部形式と呼ぶ）
実際には、内部形式は CGAEA が提供するデータ領域（セル）に保存されるデータ形式を指す。そのため、内部形式は一般に Prolog 系処理系で扱いやすい形式にする。

信念・目標（願望）

信念・目標（願望）はエージェントが信じている命題・エージェントの目標を記述するためのものである。

信念・目標（願望）の内部形式は

$$belief(X)$$
$$goal(X)$$

である。

データの更新はインタプリタが行う。

プラン

プランは目的と目的を達成するための手段を記述するためのものである。ただし、実際にプランを登録するのはこの構築ツールを利用するユーザである。ユーザが登録時に記述するプランの形式と、BDI 部内部で使用する以下の内部形式とが同じである必要はなく、前者はより記述しやすさが必要となる。これは今後の課題である。

プランの BDI 部内部形式を以下に示す。

$$plan(ID, Invocation, [FormCondition, PreCondition, AddList, DeleteList, Body])$$

- (1) *Type* : タイプ。
- (2) *Invocation* : 連想条件。
- (3) *FormConditon* : 未来指向的意図の形成条件。条件を要素とするリストである。

- (4) *PreCondition* : 現在指向的意図の候補となる前提条件 . 条件を要素とするリストである .
- (5) *AddList* : プランの目的 . 手段実行の成功時に追加される信念 . 信念を要素とするリストである .
- (6) *DeleteList* : 手段実行の成功時に除去される信念 . 信念を要素とするリストである .
- (7) *Body* : 本体 . 目的の実現手段 . 本体は , 4.1 で拡張し定義した文を各辺に持つ木構造をしている . この内部形式は文を順に並べたリストである .

文の内部形式を以下に示す .

- 基本行為 : ユーザ定義述語
- サブゴール : $subgoal(X)$
- 複文 { 文; 文; } : 文を要素とするリスト [文, ..., 文]
- if 条件 then 文_{then} [else 文_{else}] :

$if(条件, 文_{then}, 文_{else})$ または $if(条件, 文_{then})$

- 待機文 : $guard(Body)$
- ループ :

$do_until(文_{do}, 条件)$ または $until(文_{do}, 条件)$

option-generator で , プランを基にオプションと未来指向的意図を生成し , update-intentions で , オプションを基に現在指向的意図を生成する .

意図

意図はプランのタイプの代わりに (1)1つのエージェントの内部で一意的な ID を持ち , それ以外のプランの引数に (2) 現在どこまで本体を実行したかを示す Current Body と , (3) ガードのメッセージ Message を追加して表す .

以下に内部形式を示す .

intention(*ID*, *Invocation*,
[*FormCondition*, *PreCondition*, *AddList*, *DeleteList*, *Body*],
CurrentBody, *Message*)

- (1) *ID* : ユニークな ID . インタプリタで行う処理のために必要 .
- (2) *Invocation* : 連想条件 .
- (3) *FormCondition* : 未来指向的意図の形成条件 . 条件を要素とするリストである .
- (4) *PreCondition* : 現在指向的意図実行の前提条件 . 条件を要素とするリストである .
- (5) *AddList* : 目的 . 本体実行の成功時に追加される信念 . 信念を要素とするリストである .
- (6) *DeleteList* : 本体実行の成功時に除去される信念 . 信念を要素とするリストである .
- (7) *Body* : 本体 . 目的の実現手段
- (8) *CurrentBody* : 本体をどこまで実行したかを示すもの .
- (9) *Message* : ガードのメッセージ (外部イベント) . `update-intentions` で待機文の処理をするためのもの . 該当する待機文がない場合は空リスト .

オプション

オプションは , プランの引数に (1) 現在どこまで本体を実行したかを示す *CurrentBody* と , (2) ガードのメッセージ *Message* を追加して表す .

オプションの内部形式を以下に示す .

option(*Type*, *Invocation*,
[*FormCondition*, *PreCondition*, *AddList*, *DeleteList*, *Body*],
CurrentBody, *Message*)

- (1) *Type* : タイプ
- (2) *Invocation* : 連想条件 .

- (3) *FormConditon* : 未来指向的意図の形成条件 . 条件を要素とするリストである .
- (4) *PreCondition* : 現在指向的意図実行の前提条件 . 条件を要素とするリストである .
- (5) *AddList* : 目的 . 本体実行の成功時に追加される信念 . 信念を要素とするリストである .
- (6) *DeleteList* : 本体実行の成功時に除去される信念 . 信念を要素とするリストである .
- (7) *Body* : 本体 . 目的の実現手段
- (8) *CurrentBody* : 本体をどこまで実行したかを示すもの .
- (9) *Message* : ガードのメッセージ (外部イベント) . `update-intentions` で待機文の処理をするためのもの . 該当する待機文がない場合は空リスト .

イベントキュー・待ちリスト

イベントキューは先入れ先出しのスタックとして、セルに実現する (*assertz*, *asserta*) . 待ちリストは、`update-intentions` で処理される Action の中の Guard で実現する . それぞれの内部形式を以下に示す .

event_queue(Event)

- (1) *Event* : 環境知覚器・行為実行器 からの入力メッセージ . *Event* はリスト [*Message*, *Intention_id*] である . *Message* が待ち条件に含まれる場合には、該当する意図の ID が *Intention_id* となる . そうでなければ、*Intention_id* は空リストである .

wait_event([Message, Intention_id])

- (1) *Message* : 待ちリストに登録されていたメッセージ
- (2) *Intention_id* : 該当する待機文を本体に含む意図の ID

5.3.4 心的状況の登録方法

この構築ツールのユーザは、エージェントに与える信念、目標、意図、プランを以下の初期設定ファイルに記述し、これらエージェントに登録する。初期設定ファイルは、

- plan_lib.g：プランライブラリ
- belief.g：信念
- goal.g：願望（目標）
- intention.g：意図
- user.g：ユーザ定義述語。プラン本体に記述する基本行為の実体。

である。現時点の実装では、これらのファイルはBDI部内部形式で記述しているが、構築ツールとしての使いやすさを考慮した形式を提案する必要がある。これらユーザ定義に使用する書式とBDI内部形式とが異なる場合には、前者を後者に変換するコンパイラを作成する。これらは今後の課題である。

5.3.5 インタプリタ部の実装方法

以下に、4章の拡張BDIアーキテクチャに基づいたインタプリタ部の実装方法について述べる。

initialize-state

前述の初期設定ファイル群をセル bdi に設定する。

option-generator 意図の形成

意図を形成するために、以下の処理を行う。

- 到着したメッセージが待ち条件のメッセージである場合
（入力メッセージに意図 ID が含まれている（[Message,Intention_id]）場合）
Intention_id を ID として持ち、該当する待機文が本体に存在する意図を待ち条件のメッセージ Message を含むオプションとしてオプションリストの

先頭に追加 (option/5 を *asserta*) する。先頭に追加することで、以降ではこのオプションが優先的に処理される。

- 到着したメッセージが新たなプランの連想条件 (Invocation) となる場合該当するプランの未来指向的意図の形成条件 (Form Condition) が B から真と推論できる場合、未来指向的意図を形成 (intention/5 を *assertz*) する。プランの Type から intention_id を生成し、各エージェント内で意図を一意に扱えるようにする。これらの処理を全プランに対して行う。形成する未来指向的意図を 1 つに決定するかどうかは今後の課題とする。
- この時点で現在指向的意図の候補となる前提条件 (PreCondition) が B から真と推論できる意図を次に実行可能な現在指向的意図のオプションとして、オプションリストの末尾に追加 (option/5 を *assertz*) する。

deliberate(実行プランの選択)

option-generator で形成された意図から、何らかの方法でオプションを 1 つ選択し、オプションリストを削除する。選択方法は、現在の実装ではランダムな選択としているが、別途アーキテクチャの外から与えるメタレベルプランによる熟考で選択することも考えられ、その方法の見当や実装は今後の課題である。

update-intentions(意図の更新)

deliberate で選択した意図を execute で実行するために、意図を更新し、次に実行する文を以下の処理によって再帰的に (最終的には基本行為になるまで) 評価して次に実行する行為 Action を確定する。

次に実行する文 (リスト Current Body の先頭要素) が

- if(Condition,Then) の場合
条件 Condition が B から真と推論できれば、文 Then を再帰的に評価する。真でなければ、Action は空リストとなる。
- if 文 if(Condition,Then,Else) の場合
条件 Condition が B から真と推論できれば、文 Then を、そうでなければ文 Else を再帰的に評価する。

- `until(Condition,Do)` の場合
条件 `Condition` が `B` から真と推論できれば, `Action` は空リストとなる. 真でなければ, 次に条件をチェックするまでの間に実行する文の後に `until(Condition,Do)` を追加したリストを再帰的に評価し, これによってループを実現する.
- `do_until(Do,Condition)` の場合
`do_until` ループを使い, `until` ループを実現する. 文 `Do` に `until(Condition,Do)` を追加したリストを再帰的に評価する.
- 待機文 `guard(Body)` の場合

`Message` がガードの条件と一致したところで, `Action` が確定し, これを再帰的に評価する.

- 複文 `[Action—_]` の場合
リストの先頭要素が `Action` である. これを再帰的に評価する.
- サブゴール `subgoal(X)` の場合
これを `Action` とする.
- 基本行為の場合
これが具体的に次に実行する行為である. `Action` を文とする.

その後、確定した `Action` がサブゴールである場合はそれを `g-add` 内部イベントとしてイベントキューに加え (`event_queue(X)` を `assertz` し), このサブゴールが終了するまで, 選択された意図を `I` 中の意図スタックにプッシュしておく.

`execute(意図の実行)`

`update-intentions` で確定した次に実行する行為が基本行為 (ユーザが予め定義しておいた述語) ならば, 実行 (`call`) する. サブゴールならば, 何もしない.

`get-new-external-events`

環境知覚器・行為実行器と同期を取るために, 外部イベントの書き込み命令 (適当な文字列) を出力する. メッセージの終端を表す記号 (例えば, 「`[]`」) まで入力を受け、イベントキューに保持する. (`event-queue(X)` を `assertz`)

心的状態の更新 (drop-*-attitudes)

execute での基本行為の実行 (*call*) が成功した場合 drop-successful-attitudes で本体 (Body) の実行を全て終了した意図 (Current Body が空リストである意図) を削除する。DeleteList の内容を信念から削除し、AddList の内容を信念に追加する。これらを該当する意図がなくなるまで繰り返し実行する。

execute での基本行為の実行が失敗した場合には、コミットメント戦略に従い目標・意図を撤回する。

また、サブゴールが終了した時は update-intentions でプッシュした意図を意図スタックからポップする。

5.4 実装の現状と今後の課題

現在、本章で示した実装方法に基づいた合理的エージェント構築ツールの実装は、動的コミットメント戦略の処理を残して、終了している。また、次章で詳細を述べる研究所見学案内システムは、この構築ツールを使用して構築した。他にも現在ホテル検索案内を行う自然言語対話システムを構築中であり、現状では概ね問題なく実行できている。

4章で提案した拡張 BDI アーキテクチャをコミットメント戦略の処理、サブゴール、意図の構造等、より忠実に実装することは今後の課題である。また、構築ツールとしてユーザの使い易さを重視することも今後の課題である。

6 応用事例と検証

本章では、広域情報ネットワーク上で不特定多数の人々の活動を支援するために、マルチエージェント環境において意図的に言語行為を遂行する合理的エージェントの基本設計を通じて、4章で提案した拡張 BDI アーキテクチャの有用性を示す。具体的には、基本設計を提案するための事例として「研究所見学案内システム」の概要を述べ、そこで使用される「研究者秘書エージェント」をモデルとして意図的に言語行為を遂行する合理的エージェントに対する基本的な仕様を列挙し、これらを拡張 BDI アーキテクチャと FIPA を適用して実現する。次に、秘書エージェントの情報フィルタリングの事例を通じて、4.1 節で提案したプランの拡張が、人間または他のエージェントと協調して問題解決を行うために対話プランを構成する言語行為を意図的に遂行する合理的エージェントの基本設計に有効であることを示す。

6.1 研究所見学案内システム

研究所見学案内システムは、不特定多数の利用者に対する“見学受付業務サービス”および研究所の見学者に対する“見学案内業務サービス”を行う。

ヘルパーエージェント [13] は、利用者の言語行為を“行為指示行為”と解釈すると、そのサービスを提供するエージェントの推薦を Directory Facilitator (DF) から受け、そのエージェントに利用者の発話に対する提案を要求する。エージェントは、仲介された発話を自然言語処理して利用者の意図を解釈し、その意図に応じて問題解決した返答を利用者へ通知するようヘルパーエージェントに要求する。このように、本システムでは、エージェント間のコミュニケーションを、相互の目標を達成するために意図的に遂行する言語行為と見なす。また、各エージェントは、DF に広告 (advertise) したサービスの範囲内で利用者からの発話を受け付け必要なら自らの自然言語処理を通じて利用者の意図を解釈し問題解決を行う。

見学受付エージェントは、“見学受付業務サービス”および“見学案内業務サービス”の窓口を行うエージェントで、利用者からの“見学テーマ提案依頼”や“スケジュール調整依頼”などの発話を受け付け、それぞれ「見学テーマ提案エージェント」と「スケジュール調整エージェント」に利用者への提案を要求する。

見学テーマ提案エージェントは、「見学者個人情報エージェント」から得た利用者の興味やシステム利用履歴などの個人情報を基に、情報フィルタリングされた

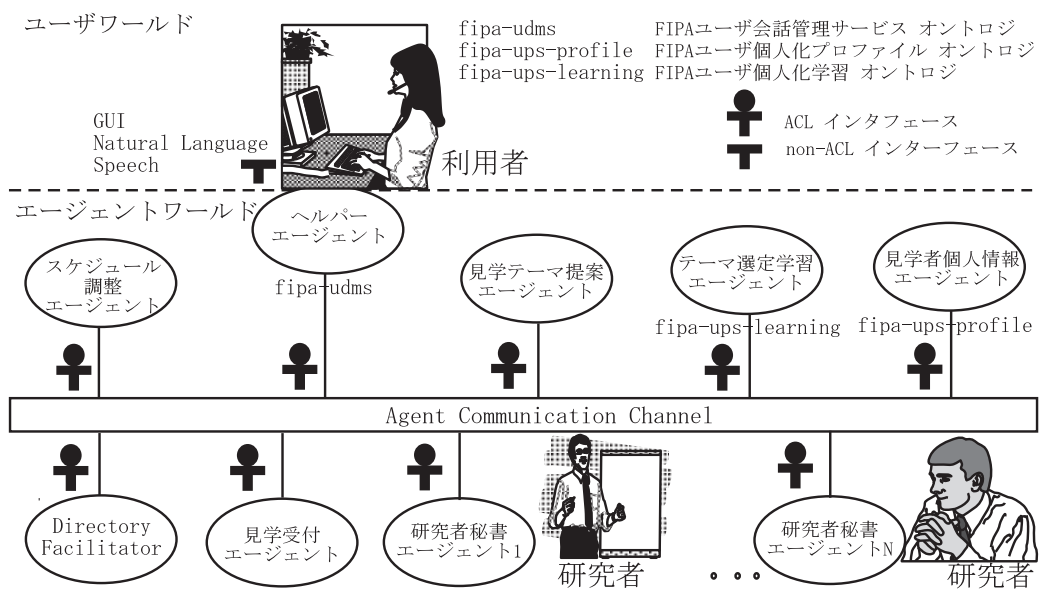


図 5: 研究所見学案内システム

研究テーマと研究者名を「テーマ選定学習エージェント」から得る。次に、これら研究者名に対して DF から得た全ての研究者秘書エージェントに対して、該当する研究テーマに関する見学対応の交渉を契約ネットプロトコル [30] を用いて行う。そして、落札したエージェントからの研究テーマと概要説明を利用者に通知するようヘルパーエージェントに要求する。この処理は、利用者が見学テーマを決定するまで繰り返される。

スケジュール調整エージェントは、決定された見学テーマに関する研究者秘書エージェントと利用者とのスケジュール調整を繰り返してスケジュールを決定する。研究者秘書エージェントは、研究者の見学対応に対する目標を達成することを主眼において利用者の個人情報に依存した情報フィルタリングを行うエージェントである。詳細は 6.2 節で述べる。システムと利用者の具体的な対話例とシステム内におけるエージェント間のコミュニケーション例をシステムの動作例として 6.4 節で示す。

6.2 意図的な合理的エージェントの基本設計

意図的な合理的エージェントの基本設計として、研究所見学システムにおける研究者秘書エージェントに対する基本設計を提案する。

例えば「他の研究者と議論がしたい」という願望を持つ研究所内の研究者は、自分の秘書エージェントに自分の願望を与え、見学対応（見学日程の調整や研究に関連する情報の提示等）を権利委譲する。つまり、研究者秘書エージェントは、権利委譲を受けた研究者の見学対応に対する目標を達成することを主眼において利用者の個人情報に依存した情報フィルタリングを行う秘書エージェントである。情報フィルタリングの結果、秘書エージェントは利用者に応じて異なるサービスを行なう。

具体的には、見学テーマ提案エージェントから依頼された「見学対応交渉」に対して、研究者の見学対応に対する目標を達成すると判断すれば、承諾する。ただし、見学対応がダブルブッキングしたときは、研究者の信念（価値基準）に従って、どちらかをキャンセルする、というサービスを行うことができる。

たとえば、以下のようなシナリオを考える。

研究者秘書エージェントは、権利委譲を受けた研究者から「他の研究者と彼の研究テーマである C-MAP について議論したい」という目標が与えられ、この目

標を達成するために構築された対話プラン「研究者への見学対応」を意図として形成する。同様に、「博物館など学芸員に対して、C-MAP を売り込みたい」という目標も与えられ、この目標を達成するために構築された対話プラン「学芸員への見学対応」を意図として形成する。その後、見学テーマ提案エージェントから、契約ネットプロトコルを用いた“研究者に対する C-MAP 見学対応告知”があり、“無条件で研究者への見学対応”に「入札」して、見学テーマ提案エージェントからの「落札」を待っている。その間に、見学テーマ提案エージェントから、新たに、“学芸員に対する C-MAP 見学対応告知”があったため、権利委譲者の信念「議論よりは売り込みの方が重要である」に従って、研究者への見学対応より、学芸員への見学対応を優先すると判断して、以前に「入札」した“無条件で研究者への見学対応”をキャンセルして、“無条件で学芸員への見学対応”に「入札」し、見学テーマ提案エージェントからの「落札」を待つ。以上のシナリオから、基本的な要求仕様を列挙する。

- (1) 権利委譲者の目標、信念、プランなどをエージェントに登録できる。
- (2) 権利委譲者の目標が与えられると、その目標を追加し、さらに達成するための最良のプランを熟考して、意図として形成する。この意図は目標を達成するまで持続させることができる。
- (3) 前提条件が満たされる意図を順次選択して実行し、目標を達成すれば、目標および意図を消去する。
- (4) 広域情報ネットワーク上で意味の規定された相互作用および通信を行うことができる。
- (5) 契約ネットプロトコルなどのあらかじめ規定されたエージェント間の相互作用プロトコルや基本通信行為を対話プランの言語行為として登録し環境および心的状態に依存して実行することができる。

(1)(2)(3)(5)の実現手段として 拡張 BDI アーキテクチャを (4)の実現手段として FIPA を採用する。

3.1 節で述べたように、2.2 節で紹介した従来の BDI アーキテクチャは (1)、(2)(3)は実現できるが (5)の実現には適していない。一方 4 章で提案した拡張 BDI アーキテクチャを用いれば、FIPA 準拠の基本通信行為と相互作用プロト

コルを言語行為として、環境や心的状態に依存した対話プランとして実行することができる。

以下で、FIPA の概要と採用理由を述べる。

6.3 FIPA

本節では、FIPA の採用理由、基本通信行為と相互作用プロトコルの分類を述べる。

6.3.1 FIPA の採用理由

Agent Communication Language(ACL)としては、KQML、KIF、Ontolingua [29]などの標準がすでに確立されている。しかし、我々はコミュニティウェアの構築には、広域情報ネットワーク上で相互運用性を保証したエージェントの外部インターフェースの規定が必須であると考え、これらの ACL には規定されていない。そこで、TCP/IP 上で相互運用性を保証するためにエージェントの外部インターフェースを規定した FIPA (Foundation for Intelligent Physical Agents) [31]の ACL を採用する。

6.3.2 基本通信行為と相互作用プロトコルの分類

FIPA では、基本通信行為として、受理、同意、取り消し、実行の申し込み、確信の通知、否定の通知、失敗、通知、真偽の通知、参照の通知、理解不能、提案、真偽の質問、参照の質問、拒否、提案の却下、要求、条件を満たす時点で実施を要求、条件を満たす毎に実施を要求、および、登録の申し込みなどが、形式的に定義されている。また、上記の基本通信行為を組み合わせ、FIPA-request、FIPA-query、FIPA-request-when、FIPA-contact-net、FIPA-iterated-contact-net、FIPA-auction-english、および、FIPA-auction-dutch などの相互作用プロトコルが定義されている。このようにあらかじめ規定された相互作用プロトコルを使うことで、単に注意深くプロトコルに従うだけで、他のエージェントと意味のある言語行為を行うことができる。本論文では、これら FIPA に準拠した基本通信行為と相互作用プロトコルを言語行為として採用する。

6.4 各エージェント間のコミュニケーション

本節では、研究所見学案内システムにおけるエージェント間のコミュニケーションを示すために、利用者とシステムの対話例、および、エージェント間のメッセージ事例を示す。各エージェントは拡張 BDI アーキテクチャに基づいて構築されている。

6.4.1 対話例

利用者 (U) と研究所見学案内システム (S) との対話例を以下に記述する。この対話例で、見学希望者は "相互作用" に関する研究の見学を希望しているが、プロジェクト C-MAP の表示をきっかけに、同プロジェクトの中心人物角さんのメールアドレスをシステムに尋ね、その後、C-MAP の見学を希望することをシステムに告げる。

- U₁ ATR の見学をしたいのですが。
S₁ 少々お待ち下さい.....
S₂ (Web に研究所見学案内システムの概要を出力して) どのようなことにご興味をお持ちでしょうか？
U₂ 相互作用について見学できますか？
S₃ 少々お待ち下さい.....
S₄ 相互作用には、以下の研究があります。1.C-MAP：展示ガイドシステム、2. 社会的相互行為の研究、どの見学をご希望ですか？
U₃ ところで、角さんのメールアドレスを教えてください。
S₅ 角さんのメールアドレスの検索ですね。少々お待ちください.....
S₆ 角さんのメールアドレスは xxxxx です。
S₇ ところで、どの見学をご希望でしょうか？
- U₄ C-MAP の見学をお願いします。
S₉ 少々お待ち下さい。3 階の実験室までお越しく下さい。
U₅ わかりました。ありがとうございました。
S₁₀ ご利用ありがとうございました。

6.4.2 XML のメッセージ事例

図 6 は、対話例の S_3 までに交換されるエージェント間のコミュニケーションの流れを表している。また、システム実装において、エージェント間のメッセージは FIPA ACL に準拠した XML を用いて記述した。

6.5 研究者秘書エージェントの情報フィルタリング事例

本節では、秘書エージェントの情報フィルタリングの事例を通じて、4 章で提案した拡張 BDI アーキテクチャが、人間または他のエージェントと協調して問題解決を行うために対話プランを構成する言語行為を意図的に遂行する合理的エージェントの基本設計に有効であることを示す。

6.5.1 研究者秘書エージェントの基本設計

秘書エージェントの基本設計は、拡張 BDI アーキテクチャを用いれば、利用者の個人情報に依存して情報フィルタリングを行うために研究者が持っている信念、目標（願望）、および、対話プランを分析し、研究者秘書エージェントの心的状態として登録することである。たとえば、権利委譲した研究者が持っている信念、目標、対話プランを、図 7、図 8 のように登録する。

(G1),(B1),(P3) は、それぞれ「研究者と C-MAP に関する議論をしたい」という願望、「議論より売り込みの方が重要である」という信念、「見学者が研究者の場合は、C-MAP 見学対応を申し込む」という対話プランである。特に、対話プランは、研究者の目標に対応したトップレベルのプラン、その予約行為の仕方の一つとして、このサブゴールを達成する「見学対応調整」サブプラン、および、そのサブゴールである見学テーマ提案エージェントとの交渉を行うサブプランの 3 階層とした。これは、それぞれのレベルで、行為の選択肢があるためである。また、提示文は、図 9 のようにデータベース化して使用する。

6.5.2 秘書エージェントの処理概要

以下で、6.2 節で提示したシナリオに基づいて、秘書エージェントを拡張 BDI アーキテクチャを用いた処理概要を記述することで、拡張 BDI アーキテクチャ

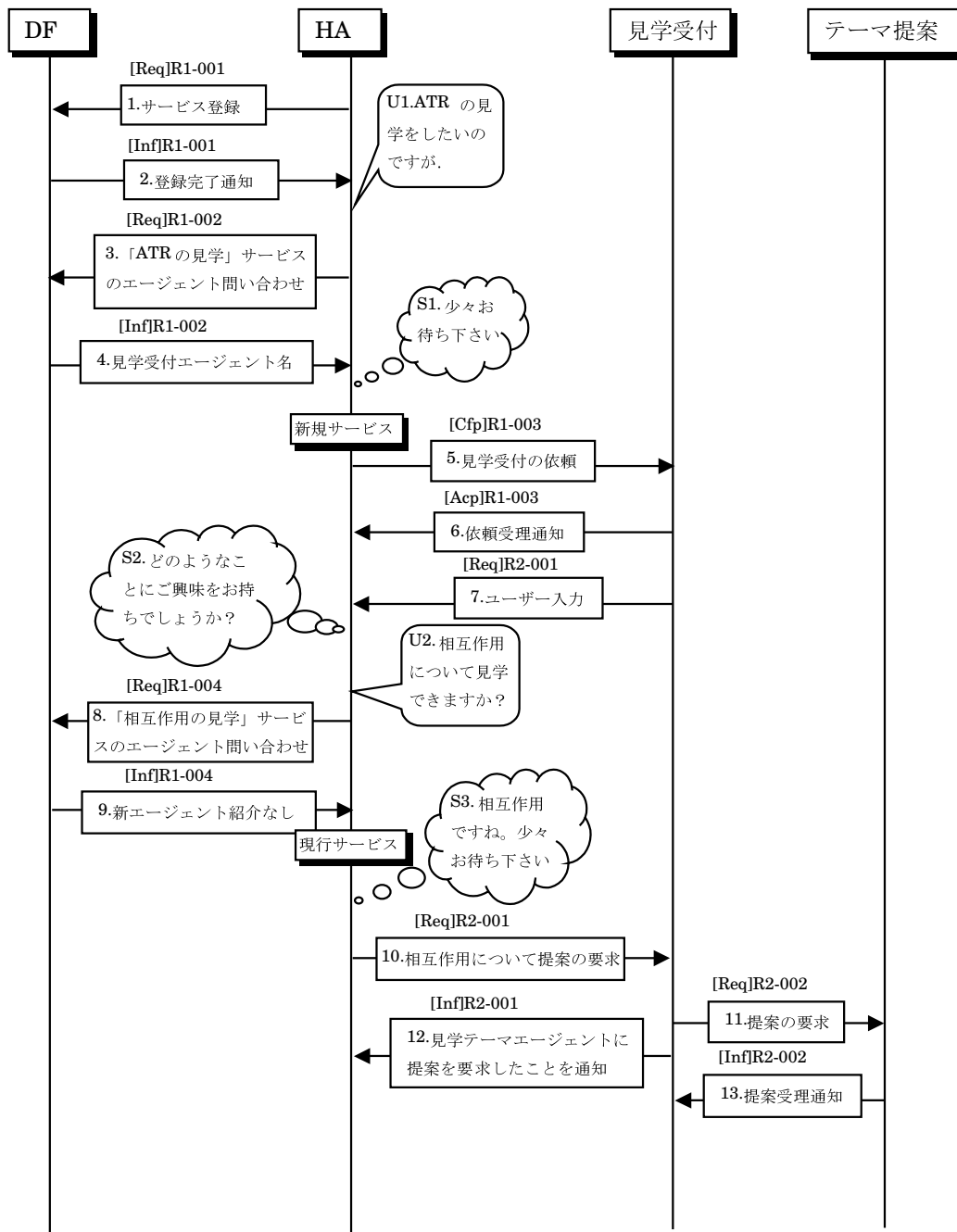


図 6: エージェント間のコミュニケーション

の基本設計の有効性を検証する。

研究者秘書エージェントは、目標 (G1) が与えられているので、この目標を達成するために、熟考して選択した対話プラン (P1) 「研究者への見学対応」を意図として形成する。そして、プラン本体の先頭のサブゴール appointment を達成するために、g-add 内部イベントを発行する。その後、そのサブゴールを達成する (P2) 「研究者への見学対応調整」が意図として形成され、同様に、権利委譲した研究者は病気ではないので、実行され、先頭のサブゴール propose を g-add 内部イベントとして発行し、その後、(P3) 「研究者への見学対応交渉」が意図として形成される。

同様に、目標 (G2) も与えられているので、対話プラン (P4) 「学芸員への見学対応」、サブプラン (P5) 「学芸員への見学対応調整」、サブプラン (P6) 「学芸員への見学対応交渉」が順次、意図として形成される。

その後、契約ネットプロトコルを用いた見学テーマ提案エージェントからの見学対応告知により、連想条件にマッチするプラン (プラン記述は省略) によって「received-cfp(見学テーマ提案AG, 研究者への対応)」と「received-cfp(見学テーマ提案AG, 学芸員への対応)」が、信念に追加される。

その後、g-add 内部イベント “研究者への C-MAP 見学対応の提案” は、プラン (P3) の連想条件とマッチし、かつ、前提条件が B から真と推論できるので、このプランを意図として形成し、さらに、options に追加する。この意図が次に実行され、その告知を入札して、対話プラン (P3) の待機文は、update-waiting-list によって待ちリストに書き込まれる。

その後、g-add 内部イベント “学芸員への C-MAP 見学対応の提案” もプラン (P6) の連想条件とマッチし、かつ、前提条件が B から真と推論できるので、このプランも意図として形成され、さらに、options に追加される。

次に、見学テーマ提案エージェントからの “研究者への C-MAP 見学対応入札を受理する” という外部イベントは待ちリスト中の対話プラン (P3) の待機文と単一化したので、options として選択される。そして、deliberate において、「議論より売り込みの方が重要」という信念 (B1) を用いて option-generator で選択した二つの実行プラン候補の中から (P6) を次に実行することを選択し、 \neg (EF propose(研究者, C-MAP)) を信念に追加するようなメタレベルプラン [23] を適用する。このメタレベルプランの記述は省略する。そして、プラン (P6) の「学芸員への見学対応交渉」を実行することで学芸員への見学対応を入札する。

次サイクルの `deliberate` で、研究者に対するプラン (P3) が選択され `update-intentions` で待機文の評価を行い、`if` 文の条件 $\neg (\text{EF propose}(\text{研究者}, \text{C-MAP}))$ は B が真であるので `:cancel` 文が選ばれ、次の `execute` で実行される。

以上、6.2 で定義したシナリオは実行され、秘書エージェントは、研究者の信念にしたがって、先に入札した研究者との見学対応をキャンセルし、学芸員への見学対応を落札することになる。

7 考察と課題

7.1 拡張 BDI アーキテクチャの実装

Rao らも、彼らが提案した BDI アーキテクチャにおいてマルチエージェントシステムの相互作用の様相を直接扱う方法を課題として取り上げている [23]。我々の提案は、FIPA の基本通信行為やプロトコルを言語行為として、マルチエージェントシステムの相互作用をプラン本体に表現できるように BDI アーキテクチャを拡張したことで、その課題に答えた。現在、5 章の実装方法に基づき、CGAEA [33] を用いて、BEL 以外の様相オペレータが記述された BDI logic の証明と動的コミットメント戦略を除いて、拡張 BDI アーキテクチャの実装を終了している。また、6 章の秘書エージェントを構築して、拡張 BDI アーキテクチャの検証もほぼ終了している。FIPA の基本通信行為は、信念、願望、意図に基づいた言語行為として、KD45 の構造を持つ SL と呼ぶ様相論理を用いて形式的に定義されている。その定義中には、その行動を計画するために満たさなければならない「可能事前条件」とその行動が選択された理由となる「合理的な効果期待」の記述があり、それぞれ、拡張 BDI アーキテクチャの「連想条件」「前提条件」と「add/delete list」に相当する。そこで、他のエージェントと通信することによって協調動作を行うように分類された FIPA の基本通信行為や相互作用プロトコルを拡張 BDI アーキテクチャのプランとして記述したことによって、信念、目標（願望）、意図に基づいて、他のエージェントと協調動作を行う合理的エージェントが構築できたと言える。

7.2 心的状態の整合性

拡張 BDI アーキテクチャにおける心的状態の更新は、2.1 章の BDI logic の心的状態の整合性公理を満足するように更新する必要がある。信念管理については、TMS [35] または ATMS [36] などの先行研究があるが、複数の様相オペレータを対象とした BDI logic には、そのままでは適用できない。ただし、2.1 章の (A4)(A5) の意図や目標の内省公理を用いれば、信念に関する整合性のみを対象とすればよいと考えられるが、今後の検討を要する。

7.3 意図形成条件を用いた対話プランの推論

また、我々の目標はコミュニティウェアの構築技術であり、拡張 BDI アーキテクチャの自然言語対話システムへの応用を予定している。その為、他のエージェントからの言語行為のプラン推論を行う必要がある。本論文での事例で提示したように、まず、典型的なプロトコルを種々のアプリケーションから集積し、プランとして登録することから始める。次に、これらの多くのプランの状況分析を信念として構築しプラン推論の足掛かりとする。我々が 4.2 節で提案した未来指向的意図 [22] の形成条件はまさに、対話プラン解釈条件に該当し、従ってプラン推論の一つの方法になると考えている。特に自然言語処理対話には、6 章の事例のように、明示的に目標が与えられることは少なく、文脈を突き通すことで利用者の要望が分かる。たとえば、秘書エージェントが旅行サービスを提供するとき、利用者とエージェントは「おはようございます」「おはようございます」「中津さんが出張します」「そうですか」「東京でホテルを探しているんだけど」「わかりました」...「新幹線の切符も取って欲しいのだけれど」「わかりました」などの対話が続く。「ホテルを探している」、「切符を取って」などの利用者の要求が分かるまでは、ただ、文脈を蓄え、プランが決まってから、それまでの文脈を利用することになる。このシステムでは意図形成条件で文脈を突き通して、利用者の発話を毎回チェックし、要望が理解できるかどうかのプログラムを実行する。利用者要求が理解できればプラン形成することになる。現在、このシステムも構築中である。

7.4 動的コミットメント戦略を用いた協調

動的なコミットメント戦略を使用することで、他の合理的エージェントと状況に依存した協調戦略の動的な選択に使用できるものと考えられる。たとえば、blind 戦略はどのような状況でも共同目標の達成を目指して協力する、single-mind 戦略は共同目標が実現可能だと信じている状況では協力する、open-mind 戦略は共同目標を信念などに基づいて撤回するまでは協力する、などの協調戦略を、共同目標単位に動的に選択することができる。この動的戦略を用いると、他のエージェントとの協調、妥協、拒否などに関する言語行為の選択に利用できるものと考えられる。これも今後の課題である。

8 おわりに

本論文は、合理的エージェントの実現に BDI アーキテクチャ [24, 23] を用いる場合の問題点を挙げ、(1) FIPA などの言語行為を用いてエージェント間のコミュニケーションをプラン本体に自然に表現できるプラン、(2) プランに、将来実行するという未来指向的意図の形成条件の追加、(3) エージェントの置かれた状況や心的状態に応じて、*blind*, *single-mind*, *open-mind* などのコミットメント戦略を動的に選択できる意図の持続機能、などの拡張を提案した。また、この提案に基づいた合理的エージェント構築ツールの実装方法について述べた。そして、拡張 BDI アーキテクチャの有効性を応用事例で示した。今後の課題としては、コミットメント戦略の決定メカニズム、心的状態の信念管理、プロトタイプ of 構築・評価、プラン推論、および、他のエージェントとの協調、妥協、拒否などに関する言語行為の選択などが挙げられる。

9 謝辞

本研究を進めるにあたって、ご指導頂いた、本学理学部情報科学科 新出尚之講師に深く感謝致します。

本学人間文化研究科 情報科学専攻の皆様にご深く感謝致します。在学期間を通し大変お世話になりました。本学理学部情報科学科 城和貴教授に感謝致します。ATR 知能映像通信研究所との交流の機会を作って頂きました。

本研究を進めるにあたり、システム実装も含め、数多くの貴重な議論の機会と示唆を頂いた、株式会社 ATR 知能映像通信研究所アート・アンド・テクノロジープロジェクト 高田司郎 客員研究員に深く感謝致します。

本研究の技術基盤を支えてくださった、株式会社 ATR 知能映像通信研究所 中津良平社長に感謝致します。株式会社 ATR 知能映像通信研究所第二研究室 間瀬健二室長、株式会社 ATR 経営企画部 片桐恭弘次長に感謝致します。本研究の初期段階から、議論の機会と貴重な示唆を頂きました。

システム試作実験において、コンテンツ用の情報を提供して頂いた、角 康之氏、竹内勇剛氏、岡田美智男氏、今井倫太氏、鈴木良太郎氏、川戸慎二郎氏、土佐尚子氏に感謝致します。特に、角 康之氏には C-MAP に関連する資料を頂きました。感謝致します。

本論文の校正に協力して下さった、米沢 朋子氏、高橋 徹氏、James King 氏、Rodney Berry 氏に感謝致します。

参考文献

- [1] Toru Ishida, editor. *Community Computing: Collaboration over Global Information Networks*. John Wiley & Sons, 1998.
- [2] Toru Ishida, Toyoaki Nishida, and Fumio Hattori. Overview of Community Computing. In Ishida [1], chapter 1, pp. 1–11.
- [3] Kazuhiro Kuwabara, Hirofumi Yamaki, Hiroshi Ishiguro, and Toru Ishida. Agent Networks for Connecting People Together. In Ishida [1], chapter 2, pp. 13–53.
- [4] 角康之, 間瀬健二. 実世界コンテキストに埋め込まれたコミュニティウェアの構築. *インタラクシオン* 2000, pp. 123–130. 情報処理学会, 2000.
- [5] Katherine Isbister, Hideyuki Nakanishi, Toru Ishida, and Cliff Nass. Helper Agent: Designing an Assistant for Human-Human Interaction in a Virtual Meeting Space. In *Proc. of International Conference on Human Factors in Computing Systems (CHI-2000)*, pp. 57–64, 2000.
- [6] Michael Wooldridge. *Reasoning about Rational Agents*. The MIT Press, 2000.
- [7] Michael Wooldridge and Anand Rao, editors. *Foundations of Rational Agency*, Vol. 14 of *Applied Logic Series*. Kluwer Academic Publishers, 1999.
- [8] Michael Wooldridge and N. R. Jennings. Agent theories, architectures, and languages: A survey. In *Intelligent Agents: Theories, Architectures, and Languages Lecture Notes in Artificial Intelligence LNAI 890*. Springer Verlag, Amsterdam, 1995.
- [9] 石田亨, 片桐恭弘, 桑原和宏. 分散人工知能. 並列処理シリーズ, vol.11. コロナ社, 1996.
- [10] 大沢英一. マルチエージェント環境における協調スキームに関する研究. PhD thesis, 1995.

- [11] Philip R. Cohen and Hector J. Levesque. Communicative Actions for Artificial Agents. In *Software Agents*, pp. 419–436. The MIT Press, 1997.
- [12] D. R. Traum. Speech Acts for Dialogue Agents. In Wooldridge and Rao [7], pp. 169–201.
- [13] 高田司郎, 五十嵐新女, 中津良平. インターネット上における自然言語対話システムの分散処理について. 信学技報 NLC2000-9, pp. 63–70, 2000.
- [14] 高田司郎, 新出尚之, 五十嵐新女, 間瀬健二. 動的なコミットメント戦略を考慮した BDI アーキテクチャ. 情処研報 2000-MPS-31, 2000. (to appear).
- [15] 高田司郎, 五十嵐新女, 新出尚之, 榎本美香, 間瀬健二, 中津良平. マルチエージェント環境において意図的に言語行為を遂行する合理的エージェントの基本設計. SAA2000:ソフトウェアエージェントとその応用ワークショップ, pp. 125–132, 2000.
- [16] 高田司郎, 新村 清彦. 対話を仲介するヘルパーエージェント. 人工知能研資, SIG-SLUD-A002-1, 2000.
- [17] 高田司郎, 新出尚之, 五十嵐新女, 間瀬健二. 拡張BDIアーキテクチャ. MACC: マルチ・エージェントと協調計算ワークショップ, ソフトウェア科学会, 2000.
- [18] 高田司郎, 新出尚之, 五十嵐新女, 間瀬健二. 合理的エージェントのコミュニケーション実現方式. インタラクシオン 2001, 情報処理学会, 2001.
- [19] 五十嵐新女, 高田司郎, 新出尚之, 間瀬健二. 個人化エージェントの意図的な情報フィルタリングについて. 第61回情処全国大会 講演論文集, pp 2-109–2-110, 2000.
- [20] 五十嵐新女, 高田司郎, 新出尚之, 間瀬健二. FIPA+BDI アーキテクチャ=合理的エージェントの実現方法. 情処研報 2000-MPS-32, pp21–24, 2000.
- [21] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, 1987. (角脇俊介, 高橋久一郎 (訳), 意図と行為 — 合理性, 計画, 実践的推論, 産業図書, 1994).

- [22] Michael E. Bratman and D. J. Israel and M. E. Pollack. *Plans and Resource-bounded Practical Reasoning*. In *Technical Report Technical Note 425R*, SRI International, 1988.
- [23] Munindar P. Singh, Anand S. Rao, and Michael P. Georgeff. Formal Methods in DAI: Logic-Based Representation and Reasoning. In *Multiagent Systems*, pp. 331–376. The MIT Press, 1999.
- [24] Anand S. Rao and Michael P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In *Proc. of International Conference on Principles of Knowledge Representation and Reasoning*, pp. 473–484, 1991.
- [25] J. Doyle. *Rationality and its roles in reasoning*. *Computational Intelligence*, 8, 376–409, 1992.
- [26] S. J. Rosenschein and L. P. Kaelbling. *The synthesis of digital machines with provable epistemic properties*. In *Proceedings of the First Conference on Theoretical Aspects of Reasoning about Knowledge*. J. Y. Halpern, ed. Morgan Kaufmann, San Mateo, CA, 1986.
- [27] Y. Shoham. Agent0: A simple agent language and its interpreter. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pp. 704–709, 1991.
- [28] John R. Searle. *Speech acts: An Essay in the Philosophy of Language*. Cambridge University Press, 1969. (坂本百大, 土屋俊 (訳), 言語行為—言語哲学への試論, 勁草書房, 1986).
- [29] Tim Finin, Yannis Labrou, and James Mayfield. KQML as an Agent Communication Language. In *Software Agents*, pp. 291–316. The MIT Press, 1997.
- [30] Reid G. Smith. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Trans. on Computers*, Vol. 29, No. 12, pp. 1104–1113, 1980.
- [31] <http://www.fipa.org>.

- [32] Anand S. Rao and Michael P. Georgeff. Decision Procedures for BDI Logics. *Journal of Logic and Computation*, Vol. 8, No. 3, pp. 292–343, 1998.
- [33] <http://www.aist.go.jp/ETL/etl/captain/gaea/>
- [34] E. Allen Emerson and J. Srinivasan. Branching Time Temporal Logic. In J.W. de Bakker, W.P. de Roever, and G. Rozenberg, editors, *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, pp. 123–172. Springer-Verlag, 1989.
- [35] Jon Doyle. A Truth Maintenance System. *Artificial Intelligence*, Vol. 12, pp. 231–272, 1979.
- [36] Johan de Kleer. An Assumption-based TMS. *Artificial Intelligence*, Vol. 28, pp. 127–162, 1986.

目標（願望）
(G1) discuss(研究者, C-MAP)
(G2) promote(学芸員, C-MAP)
信念
(B1) more_important(promote, discuss)
(B2) have-interest(学芸員,[CG,VR, パーソナルエージェント])
(B3) have-interet(研究者,[協調, コミュニティウェア])
:
:

図 7: 秘書エージェントの心的状態

```

(P1)
Type: 研究者への見学対応
Invocation: g-add(discuss(研究者, C-MAP))
Precondition: true
Add List: {EF discuss(研究者, C-MAP)}
Body:
  appointment(研究者, 日時)
  :inform(権利委譲した研究者, 研究者, 日時)
(P2)
Type: 研究者への見学対応調整
Invocation: g-add(appointment(研究者, 日時))
Precondition: ¬sick(権利委譲した研究者)
Add List: {appointment(研究者, 日時)}
Body:
  propose(研究者, C-MAP)
  if BEL(propose(研究者, C-MAP))
    then arrange(研究者, 日時)
    else fail
(P3)
Type: 研究者への見学対応交渉
Invocation: g-add(propose(研究者, C-MAP))
Precondition:
  received-cfp(見学テーマ提案AG, 研究者への対応)
Add List: {propose(研究者, C-MAP)}
Body:
  :propose(見学テーマ提案AG, 研究者への対応)
  (:accept-proposal ?;
   if ¬BEL (EF propose(研究者, C-MAP))
     then{:cancel; fail}
     else :inform(見学テーマ提案AG,
                  研究者向けC-MAP 概要)
   | :reject-proposal ?; fail)
(P4)
Type: 学芸員への見学対応
Invocation: g-add(promote(学芸員, C-MAP))
  :
  :
(P6)
Type: 学芸員への見学対応交渉
Invocation: g-add(propose(学芸員, C-MAP))
Precondition:
  received-cfp(見学テーマ提案AG, 学芸員への対応)
Add List: {propose(学芸員, C-MAP)}
Body:
  :propose(見学テーマ提案AG, 学芸員への対応)
  (:accept-proposal ?;
   :inform(見学テーマ提案AG, C-MAP活用方法)
   | :reject-proposal ?; fail)

```

図 8: 秘書エージェントのプランライブラリ

```
概要データベース
<summary-for-promotion>
見学案内に威力を発揮するパーソナルエージェントです
</summary-for-promotion>
<summary-for-researcher>
実世界コンテキストに埋め込まれたコミュニティウェアの研究です
</summary-for-researcher>
:
```

図 9: C-MAP の概要