

## 反射と熟考の相互作用に基づく協調的対話モデル

長谷川 隆明 中野 有紀子 加藤 恒昭

NTT 情報通信研究所

〒 238-03 神奈川県横須賀市武 1-2356

Tel: 0468-59-8214

E-mail: {hasegawa|yukiko|kato}@isl.ntt.jp

あらまし 人間同士のタスク指向対話では、対話参加者は自分の目標を達成するために相手に一方的に説明しようとするのではなく、相手からの質問や確認といった割り込みに素早く反応し、その割り込みの内容から相手の理解度を推測することによって適切に応答しながら協調的に対話を進めている。つまり、対話参加者は相手の理解度に応じて適切な発話を行っていると言える。

人間が行うような自然な対話を計算機に行わせるためには、人間が持つ特徴を考慮すれば、計算機は相手の理解度に応じて説明する方法や内容などの対話の戦略を自律的に変更できる機能を持たなければならない。本稿では、この機能は対話を維持しようとする反射的な側面とタスクの遂行を行う熟考的な側面との相互作用から創発したものであると捉え、この捉え方に基づく対話エージェントのためのモデルを提案する。

キーワード 対話モデル、人間とエージェントとの協調、ハイブリッドアーキテクチャ、ユーザモデル、マルチエージェント、自然言語処理

## A Collaborative Dialogue Model Based on Interaction between Reactivity and Deliberation

Takaaki Hasegawa, Yukiko I. Nakano and Tsuneaki Kato  
NTT Information and Communication Systems Laboratories  
1-2356 Take, Yokosuka 238-03 Japan  
Tel: +81-468-59-8214  
E-mail: {hasegawa|yukiko|kato}@isl.ntt.jp

Abstract In human dialogue, the participants not only try to accomplish their own goal but also collaborate with each other. To put it concretely, the participants generate utterances that are appropriate to the degree of the partners' understanding.

To allow a computer to communicate with us as naturally as we usually communicate with other people, we focus on this feature. In short, we think that a computer should be able to change its dialogue dynamically and autonomously according to the human's understanding.

In this paper, we propose a dialogue model with this feature. We think that a dialogue agent should have two characteristic aspects; namely, the reactive aspect that tries to maintain the dialogue and the deliberative aspect that tries to accomplish a task. Our model yields natural interaction through the interplay between these two aspects.

key words Dialogue models for autonomous agents, collaboration people between agents, hybrid agent architectures, user modeling, multi agent systems, natural language processing.

## 1 はじめに

人間同士のタスク指向対話には、相手からの問い返しや確認などの割り込みに対し対話を維持しようとする反射的な側面と、より時間を要するが対話の状況を考慮し相手にとって必要十分な情報を伝達しようとする熟考的な側面とが見られる。これら二つの側面の組み合わせにより、対話参加者は効率の良い協調的な対話を行っている。

しかし、従来の古典的プランニングを用いた対話システム [2] では、最初に発話する内容を決めそれから発話を行うために、相手の割り込みに対処できなかつたり、発話を行うときには時間の経過や世界の変化のために既にその発話内容が適切でなかつたりということが起こり、人間同士の対話のような自然な対話を実現することができなかつた。

我々は、これらの問題を解決するために、人間と計算機との自然な対話を実現するための試みの一つとして、1) 反射的な側面と熟考的な側面との相互作用のモデル化、2) 対話に反映させるべき対話相手のモデルの扱い、について取り組み、人間と対話を行うエージェントのためのモデルを提案する。本モデルは、動的に変化する環境に対応するリアクティブプランニングの枠組みを拡張し、反射的なモジュールと熟考的なモジュールとを設け、自律したこれらのモジュール同士の相互作用によって、対話における二つの側面を同時に併せ持つ。さらに、対話の中で変化する対話相手のモデルを考慮し、人間と効率的かつ協調的な対話を行うことが可能なモデルであり、対話状況の変化に応じて、対話の途中で動的に説明の戦略が出現する。

本稿では、第2節で人間対人間の対話の観察について、第3節で我々が拡張する基になつた Agent Network Architecture(ANA)[3]の概要とそれを対話に適用する際の問題点を、第4節で我々が提案する対話エージェントのモデルについて、第5節で本モデルを用いた対話実験の結果についてそれぞれ述べる。

## 2 対話における collaboration

人間同士の対話例を観察すると、説明者が一方的に説明するのではなく、利用者からの質問や確認などの割り込み素早く反応し、その割り込み内容に適切に対処することで、対話が進行している。言い換えれば、人間は次の特徴を持っているように考えられる。1つは、相手の割り込みに対し即座に応答したり、相手からの承認を求めながら説明したりする、コミュニケーションを維持するという特徴である。もう1つは、相手との collaboration[1]を取りながら、相手の理解度に応じて適切な発話を行うという特徴である。

その様子を図1に示す。この対話例は、説明者が利用者に留守番電話機の設定の説明を行っている一場面を抜き出したものである。利用者の「左側に行くと

### 1-1: 利用者が先回りの確認をする場合

- (284) E: はい。  
それです、  
このスピーカーの音量も変えることができます。
- (285) U: はい。
- (286) E: それはここにレバーがついているんですけども、このレバーを動かすと音が大きくなったり小さくなったりします。
- (287) U: 「あつ」はい。
- (288) E: なっていますね。
- (289) U: はい。  
左側に行くくと小さくなるんですね。
- (290) E: そうです。
- (291) U: はい。

### 1-2: 利用者が間違つた理解や不理解を示す場合

- (80) E: 「えーと」本体の上の方にあります
- (81) U: ええ。
- (82) E: 「えーと」カセットカバーを開けてください。
- (83) U: ええ、これはどうやって開けるんでしょうかね。
- (84) E: 「えーと」手を、横に手を当てまして、
- (85) U: 「あつ」横に、「あつ」そうですね、はい。
- (86) E: 持ち上げますと簡単に
- (87) U: 「あつ」開きました、はい。
- (88) E: 簡単に開くと思います。

図 1: 人間対人間の対話 (E: 説明者 U: 利用者)

小さくなるんですね」という発話に見られるように、利用者が説明者の先回りをするように説明者に確認する場合は、利用者の理解が十分であるため、説明者は「そうです」とだけ発話する。あるいは、利用者の「ええ、これはどうやって開けるんでしょうかね」というような、利用者の理解が不十分だつたり間違つていたりしたときの発話に対しては、説明者は以前よりも詳細に説明を繰り返す。また、相手の注意を引くために相手に確認を行うなど、相手とのコミュニケーションを維持しようとする発話が見られる。

これらの特徴を対話エージェントに実現するには、

- 相手の割り込みに対して素早く反応したり、自ら相手に確認したりするなど相手とのコミュニケーションを維持すること
- 相手の理解度に合わせて、対話途中でも説明の戦略を適切に変更できること

という2つの機能が必要である。

collaboration とは、説明者のみが対話を進行させるのではなく、説明者と利用者が一緒になつて対話を進行させることであるが、我々はこれらの機能によって相手との collaboration が実現されていると考える。

### 3 Agent Network Architecture

動的に変化する環境でのプランニングとして、リアクティブプランニングがいくつか提案されている。その中でも Agent Network Architecture(ANA)[3]は、伝統的なプランナーとリアクティブシステムの両方の特徴を併せ持っている。ANAでは、エージェントはある振る舞いに相当するモジュールを単位とするネットワークと見なされ、エージェントの行動選択はモジュール間の活性化/抑制のダイナミクスから創発する。このメカニズムでは常に現在の状態を考慮しているので、現在の状態が突然変化しても柔軟な再プランニングが可能である。

#### 3.1 ANA のメカニズム

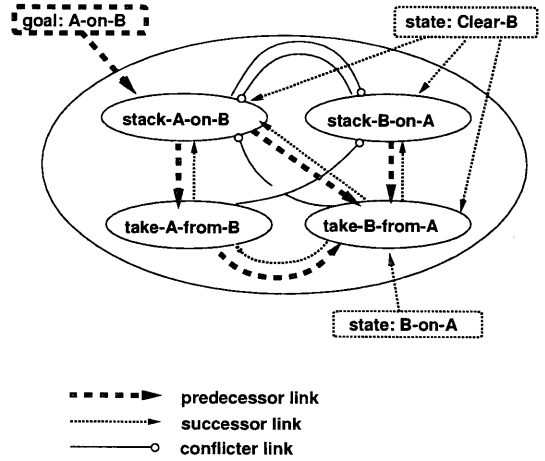
ANAでは、ひとつのモジュールが一つの振る舞いに相当する。このモジュールが複数個、粗に繋がったネットワークがANAである。ネットワークの中で、モジュール同士が活性エネルギーの活性化/抑制を通じて、命題の集合で表される現在の状態(環境)や目標状態を反映するように、ネットワークの活性状態が変化し、各モジュールに活性エネルギーが溜る。振る舞いを行うための前提条件をすべて満たし(実行可能で)活性エネルギーがある閾値を越え最も大きいモジュールが選択される。これにより状況に最も適した振る舞いが選択される。図2に積み木の世界を例に取ったANAを示す。この場合、モジュールは積み木を動かすという振る舞い、環境は積み木の置かれている状態に相当する。モジュール  $x$  は次のように STRIPS のプラン流に記述される。

$$x = (c_x, a_x, d_x, \alpha_x)$$

$c_x$  は振る舞い  $x$  が実行できるための前提条件を、 $a_x$ 、 $d_x$  は振る舞い  $x$  が実行された後の効果を、 $\alpha_x$  は振る舞い  $x$  の活性値をそれぞれ表す。活性値  $\alpha_x$  は他のモジュールからの相互作用によってモジュール間のリンクを通して活性化/抑制されたり、命題の集合からなる環境と目標からの活性エネルギーが流入したりすることで変化する。モジュール  $x$  のリンクは、 $c_x, a_x, d_x$  と他のモジュールのそれとに関係して、後継のモジュールに繋がる successor link、前任のモジュールに繋がる predecessor link、競合するモジュールを抑制する conflicter link が決定される。ANAの行動選択全体の振る舞い調整するためのグローバルパラメータを変化させるだけで、目標指向になったり状況指向になったり、環境の変化に鋭くなったり鈍くなったりさせることができる。

#### 3.2 対話における ANA の問題点

積み木の世界の例では、ANAは適切な振る舞いの系列を選択することができるが、この枠組みを直接的



初期状態でのモジュール間のリンク

図2: 積み木の世界に ANA を適用した例

に対話へ適用すると、以下の2つの問題が生じる。第一に、ANAの中にすべての振る舞い(発話)が含まれているために、質的に異なる対話における二つの側面を、単一のパラメータセットで動作するANAで扱うことは困難である。第二に、ANAでは状態を2値的にしか扱うことができないため、ユーザの理解度など2値で表すことができない状態を扱うことができない。本稿では、これらを解決するためにANAの拡張を行い、自然な対話を実現する協調的対話モデルを提案する。

### 4 対話エージェントのためのモデル

対話エージェントのためのモデルとして、基本的にANAのメカニズムを用いる。具体的には、ANAのモジュール一つを一発話とし、環境をユーザモデルを含んだ対話の状態とし、目標状態を相手に伝えたい意味内容とする。そして、主に2つの点を拡張する。第一に、対話における二つの側面である、コミュニケーションを維持する反射性と目標を達成する熟考性が並列に動作し、これらが相互作用することによって、対話の二面性を同時に表現する。第二に、ユーザモデルの表現方法を2値的ではなく連続的にすることによって、相手の理解度の変化に柔軟に対処する。

#### 4.1 対話の二面性

対話にはコミュニケーションを維持する能力と、与えられた目標を達成する能力の二つが必要である。

#### 4.1.1 発話の単位と発話出現のタイミング

対話の二面性について論じる前に、対話に ANA を導入するために、まず次の事に触れておく必要がある。

- 発話の単位
- 発話出現のタイミング

ANA のモジュールを発話または発話する振る舞いであるとみなすと、何が発話の単位になり得るのか、つまり一文を一発話とするのか、あるいはポーズを置くべきところまでを一発話とするのかなど、一発話ほどの程度の粒度で表現するのかという問題が生じる。実際の人間対人間の対話を観察しても、一つ一つの文節を分けて発話したり、複数の文節を一気に発話したりするなど様々な場合があり、明確な規則性は見い出すことができない。本研究では、発話単位は文節やポーズに捕われずに意味のレベルでのまとまりであると考え、ANA のモジュールをあるまとまった意味内容を発話する振る舞いと捉える。発話単位をこのように捉えれば、通常のプランニングの考え方をを用いることができる。例えば、目標が意味内容  $A$  と  $B$  からなるとすると、説明者が  $A$  を発話し、利用者がそれを聞いて  $B$  を発話したとすると、説明者は  $B$  を発話することなく、目標を達成したことになる。

ある意味レベルのまとまりを発話する振る舞いを ANA のモジュールとみなせば、発話出現のタイミングとは、モジュールの選択されるタイミングに相当する。ANA を対話に適用したとき、上述したモジュールの表現を使って、発話を表すモジュール  $u$  は、次のように書き換えることができる。 $c_u$  は発話を行うための前提条件で、ある意味レベルのまとまりが成り立っていないと、発話を行うことができない。 $a_u, d_u$  は発話を行った後に状態を変更するであろう予測効果を表し、 $a_u$  の追加  $d_u$  の削除により現在の状態（環境）を変更する。 $\alpha_u$  は発話を表すモジュールが持つ活性値である。 $\alpha_u$  は環境の変化や他の発話を表すモジュールの相互作用によって常に変化している。 $c_u$  をすべて満たし  $\alpha_u$  がある閾値を越えて最も高い値を持つときに、モジュールが選択され、すなわち発話が出現する。発話出現のタイミングは、発話を表すモジュールの前提条件と活性値の大きさによって決定される。

#### 4.1.2 反射と熟考の相互作用

コミュニケーションを維持する性質と目標を達成する性質を持つという対話における二面性を、パラメータセットの異なる二つの ANA を並列に動作させることによって実現した。コミュニケーションを維持するために働くモジュールのネットワークを反射エージェントとし、目標達成のために働くモジュールのネットワークを熟考エージェントとした。これによって、二

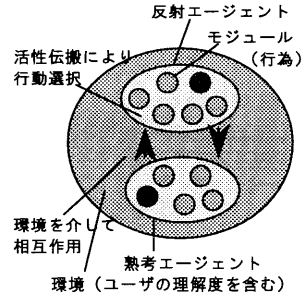


図 3: 対話モデル

つの性質が同時に現れることが可能となる。図 3 に反射エージェントと熟考エージェントの相互作用の様子を示す。

これらのエージェントは基本的に ANA で動作しているので、常に環境からエネルギーの流入があり、モジュール間の活性伝搬によってエージェントが取る振る舞い、つまり発話が決まる。さらに、エージェントの発話によって環境が変更され、変更された環境は次の発話選択に影響を与える。並列に動作している個々のエージェントは常にこれらの過程を繰り返している。このモデルでは、自エージェントの発話選択の際に、自エージェントが環境に及ぼした影響だけでなく、他エージェントの及ぼした環境の変化の影響も受けることになる。つまり、反射エージェントが変更した状態を熟考エージェントが考慮して行動を選択する、さらに熟考エージェントが変更した状態を今度は反射エージェントが考慮して行動を選択するという環境を介した相互作用が可能となる。ここには、相互作用のための直接的なメカニズムは存在せず、相互作用は環境を介してのみ行われる。

## 4.2 ユーザモデルの扱い

対話において、相手についてのユーザモデルを考慮することは重要である。本研究では、ユーザモデルを次のように二つに分けて考慮する。

- ユーザの状態
- ユーザのタイプ

### 4.2.1 ユーザの状態

ANA のメカニズムでは、環境の状態を表現する各々の命題は真か偽の 2 値で表される。しかし、対話を考えると、対話にはある意味内容についてユーザがどれくらい理解しているのかや、発話したことが完全に相手に伝わるとは限らない場合を表現しなくてはならなくなってくる。それゆえ、対話には状態を表現する方

法として、2 値的な表現よりも、連続的な表現を扱えることが望ましい。そこで、ANA の拡張の第 2 点目として、本研究ではユーザの状態をユーザに対する命題の確信度と考え、ANA の環境を構成する命題に、2 値ではなく 0~1 までの連続的な値を持たせることを考案した。これにより、例えばユーザがある対象物を同定しているという命題に対する確信度を表現できる。さらに、命題についての真偽の判断基準を設けるため、確信度に閾値を設定した。その様子を図 4 に示す。この閾値を越えている命題は真とみなし（命題 B）、越えていない命題は偽とみなす（命題 A）。

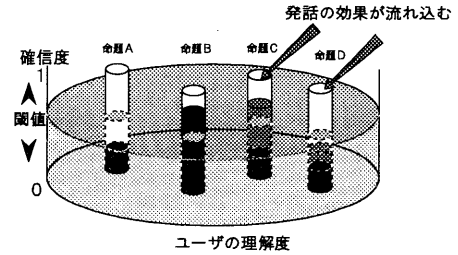


図 4: ユーザの理解度

#### 4.2.2 発話による命題への効果

状態の表現が連続的な値を取り、発話を行った効果として新たな状態に変化することから、発話を行う効果にも命題ごとに連続的な値を持たせることが必要である。ところが、発話を行った効果として、各命題の確信度が変化する度合いはどの程度になるかという問題が生じて来る。ANA ではある振る舞いが行われると、ある命題が偽から真になったり真から偽になったりしたが、対話を考えた場合、発話を行う振る舞いによって命題がすべて同じ度合いだけ変化するの直感的でない。

そこで、発話を行うという振る舞いの効果として、発話とその発話が達成に貢献する命題との関係を考慮して効果の大きさを決める。具体的には、命題を達成するために多くの発話を行わなければならないと予測する場合は、ドメインに依存してその命題が難しいから効果が相対的に小さいと考えることができる。逆に、一つの発話で命題が達成できると予測する場合は、ドメインに依存してその命題が易しいから効果が相対的に大きいと考えることができる。さらに言えば、命題を達成するのに貢献する発話数に注目し、発話による命題への効果はその命題の達成に貢献する発話数の逆数に関係すると考えた。これによって、命題の達成における難易度が表現できることがわかった。

この考え方を ANA に適用すると、ある命題の達成に貢献するモジュールの数で命題の達成における難易度を表すことができ、命題の達成に貢献するモジュールの数に伴って命題の達成が難しくなると言える。数式では次のように、発話  $u$  の命題  $p$  に対する効果  $\epsilon_{u,p}$  は、命題  $p$  ( $p \in add_u$ ) に貢献する発話数の平方根の逆数に比例すると定義する。

$$\epsilon_{u,p} = \frac{1}{\sqrt{1 + A(p)}}$$

ただし、 $A(p)$  は  $p \in add_i$  となるモジュール  $i$  の個数である。

発話を行うことにより命題の確信度に定義された分だけ発話の効果が流れ込み、その結果、閾値を越えて真とみなされることもあれば（命題 C）、逆に発話を

行っても閾値を越えずに偽とみなされたままである場合もある（命題 D）。この様子を図 4 に示す。

例えば、発話  $x$  の命題  $p$  ( $p \in a_x$ ) に対する効果は、命題  $p$  の達成に貢献する、つまり  $p \in a_x$  なるモジュール  $z$  が 6 個ある場合は、 $\epsilon_{x,p} = \frac{1}{\sqrt{1+6}} = 0.378$  だけ命題  $p$  の確信度を上昇させる。このとき、すべての命題に対する閾値が 0.5 で、命題  $p$  の確信度の初期値が 0 ならば、発話を行なったにも関わらず、命題を達成できなかったことになる。逆に、発話  $y$  の命題  $q$  ( $q \in a_y$ ) に対する効果は、命題  $q$  の達成に貢献するモジュールが 2 つしかない時は、 $\epsilon_{y,q} = \frac{1}{\sqrt{1+2}} = 0.577$  だけ命題  $q$  の確信度が上昇し、命題  $q$  の確信度の初期値が 0 ならば、閾値を越えるので、命題を達成できたことになる。

#### 4.2.3 ユーザのタイプの変化

人間同士の collaboration を実現するには、対話の途中でも、ユーザがどの程度熟練しているかを見極め、ユーザのタイプを変更することが有効である。利用者が説明者の伝えようとしていることをよく理解していると思われる時、例えば、利用者が先回りの確認を行なうなどの割り込みがある時は、説明する内容を減らすことが望ましい。つまり、ユーザのタイプを熟練者であると以降はみなす必要がある。また逆に利用者が説明者の伝えようとしていることを理解していなかったり間違って理解している時、例えば、利用者が不理解を示す発話を行ったり間違った問い返しを行ったりした時は、より多くの情報を与えたり繰り返して説明を行ったりしなければならぬ。つまり、ユーザのタイプを未熟練者であると以降はみなす必要がある。

そこで、ユーザの割り込みの内容によって反射エージェントが命題の真偽を判定する閾値を変化させることを提案する。つまり、ユーザが先回りの割り込みをした場合は命題の真偽を判定する閾値は下がり、それによって命題は真になりやすくなり、反対にユーザが不理解や間違いを表す割り込みをした場合は命題の真偽を判定する閾値は上がり、それによって命題は真

になりにくくなる。これによりユーザのタイプを変化させることが可能になる。このメカニズムも環境に付随する閾値を介するだけで間接的に反射エージェントと熟考エージェントの相互作用が行われる。

例えば、上述した例でのモジュール  $x$  の効果は  $\epsilon = \frac{1}{\sqrt{1+\epsilon}} = 0.378$  だけ  $a_x$  に含まれている命題の確信度を上昇させるが、このとき、閾値が 0.35 まで下がっていれば、閾値を越えることが可能になる。逆にモジュール  $y$  の効果は  $\epsilon = \frac{1}{\sqrt{1+\epsilon}} = 0.577$  だけ  $a_y$  に含まれている命題の確信度を上昇させるが、このとき、閾値が 0.75 まで上がっていれば、閾値を越えることができなくなり、この命題に貢献するさらなるモジュールの振舞いが必要になる。どの程度閾値を変化させるのかは今後の検討課題である。

#### 4.2.4 再プランニング

ユーザモデルが変化することによって、それまで真と見なされていた命題が偽になったり、偽と見なされていた命題が真になったりして、エージェントを取り巻く状態が劇的に変化する。これがきっかけとなり、新しい状態で ANA が動作することになり、結果的に再プランニングが行なわれる。つまり、命題  $q$  の確信度が 0.577 であった時、閾値が 0.55 から 0.75 に上昇すると、命題  $q$  は真から偽に変化したと見なされる。このため、再び命題  $q$  を真にしようとする ANA メカニズムの作用のために、命題  $q$  を効果に持つ別のモジュール  $z$  が活性化し、その結果、命題  $q$  の確信度をさらに上昇する。

### 4.3 モジュールの抽象化

ANA では、モジュールの表現を抽象化することができず、全て具体化して記述しなければならなかった。この問題を解消するため、今着目しているオブジェクトについて記述する [6] ことで、さまざまなオブジェクトを現在のフォーカスとして切り替えて扱えることを可能にした。例えば、オブジェクト  $A$  についての説明が終了したら次にオブジェクト  $B$  について説明するといった場合、オブジェクト  $x$  の説明方法だけを記述しておけば、フォーカスを切り替えることで、 $A, B$  どちらのオブジェクトの説明も生成可能である。

### 5 シミュレーションによるモデルの検証

以上に述べたモデルの妥当性を検証するために、計算機上へ実装し実際に人間との対話シミュレーションを行った。

対話のドメインとして、留守番電話機の操作の説明を行なうタスクを用いた。タスクのゴールは留守番電話機を録音状態にすることである。これを達成するには、録音状態にするための「応答 1 ボタン」を同定し、それを押すとどうなるのかという効果を説明し、

さらにそれを押させるように依頼することが必要である。また、「応答 1 ボタン」を同定するのにも、ボタンの場所や色や形や付近の特徴物などを説明しなければならぬ。

留守番電話機のドメインを用いて、実際に対話シミュレーションを行なった結果を図 5 に示す。

熟考エージェントは、この例でのタスク遂行に関係する意味内容を発話するために必要なモジュールから構成されている。例えば、オブジェクトを同定するモジュール *Recognize-Object* やオブジェクトの場所を同定するモジュール *Identify-Location-of-Object* は図 6 のように記述される。

*Identify-Location-of-Object* の add list にある *committed-object* は *Recognize-Object* の condition list に含まれているので、*Identify-Location-of-Object* は *Recognize-Object* に貢献することになる。他にも同様なモジュールが 5 つ存在するため、*Identify-Location-of-Object* が選択され「パネルの下に」と発話した時に上昇する *committed-object* の確信度は 0.378 である。閾値の初期値が 0.55 だとすると、まだ *committed-object* は真とは見なされず、「ボタンがありますよね」とは発話しない。従って、他のモジュール、例えば場所を説明するモジュールが選択され *committed-object* は 0.756 となり、閾値を越えて真と見なされ、「ボタンがありますよね」と発話することができる。

一方、反射エージェントは例えば図 7 に示すような、オブジェクトの名前についての相手からの問い返しに答えるモジュールから構成される。ユーザが「応答 1 ボタンですよ？」という先回りの確認の発話をして、エージェントが同定しようとする Object を参照した時、モジュール *Reply-Identified-Name-of-Object* が選択され、「はい、その名前のもので」と発話する。このとき、add list に *committed-object* があり、熟考エージェントの時と同様にこの命題に貢献するモジュールが他に 5 個あるため、0.378 だけ上昇し、0.756 になって閾値を越えてしまう。つまり、これはその後本来言うべき「本体、左側、真中にあるんですけど」や「応答 1 ボタンという」という発話を行なうことなしに、ボタンの同定が終了したことを意味している。さらに閾値を下げることで、今後の発話選択の際に 1 つの発話でも命題が真になりやすくなっている。逆に、ユーザが「ダイヤルボタン 1 ですか？」という間違った発話を行なった場合は、モジュール *Reply-Not-Identified-Location-of-Object* が選択され「いいえ、その名前ものではありません」と発話し、*committed-object* を 0.378 だけ下げ、さらに閾値を上げることによって、今後複数の発話を行なうことによって命題 *committed-object* が真になるようになっていく。これらの様子を図 8 に示す。

#### 5-1: ユーザの先回りのな割り込みがあった場合

エージェント: パネルの下に  
deliberate module: IDENTIFY-SOMETHING-STRIKING  
ユーザ: 応答1ですね?  
reactive module: WATCH  
reactive module: REACT\_USER\_ASK\_BACK\_NAME\_OF\_OBJECT  
エージェント: はい、その名前のものです  
reactive module: REPLY\_IDENTIFY\_NAME\_OF\_OBJECT  
エージェント: ボタンがありますよね  
deliberate module becoming active:  
RECOGNIZE-EXISTENCE-OF-OBJECT  
エージェント: それを押して下さい  
deliberate module: REQUEST-ACTION  
エージェント: いいですね  
reactive module: ASK\_FOR\_ACK  
ユーザ: はい  
エージェント: そうすると、赤く点灯するんですが  
deliberate module: EXPLAIN-EFFECT  
エージェント: これで録音が始まります  
deliberate module: EXPLAIN-ACTION

#### 5-2: ユーザの理解が間違っている場合

エージェント: パネルの下に  
deliberate module becoming active:  
IDENTIFY-SOMETHING-STRIKING  
reactive module becoming active: WATCH  
ユーザ: ダイヤル1ボタンですか  
reactive module becoming active:  
REACT\_USER\_ASK\_BACK\_NAME\_OF\_OBJECT  
エージェント: いいえ、その名前ものではありません  
reactive module becoming active:  
REPLY\_NOT\_IDENTIFY\_NAME\_OF\_OBJECT  
エージェント: 応答ボタン1という  
deliberate module becoming active:  
IDENTIFY-NAME-OF-OBJECT  
reactive module becoming active: WATCH  
エージェント: えーと  
reactive module becoming active: HESITATE  
エージェント: パネルの下に  
deliberate module becoming active:  
IDENTIFY-SOMETHING-STRIKING  
エージェント: 本体、左側、真中にあるんですけど  
deliberate module becoming active:  
IDENTIFY-LOCATION-OF-OBJECT  
エージェント: ボタンがありますよね  
deliberate module becoming active:  
RECOGNIZE-EXISTENCE-OF-OBJECT

図 5: エージェントとユーザの対話例

Module: Recognize-Object  
condition: committed-object  
add: recognized-object  
delete: committed-object  
utter: 「ボタンがありますよね」

Module: Identify-Location-of-Object  
condition: committed-location-of-object  
add: committed-object  
delete: committed-location-of-object  
utter: 「パネルの下に」

図 6: 熟考エージェントを構成するモジュール

Module: Reply-Identified-Name-of-Object  
condition: user-identified-name-of-object  
add: committed-object  
delete: user-identified-name-of-object  
utter: 「はい、その名前のものです」

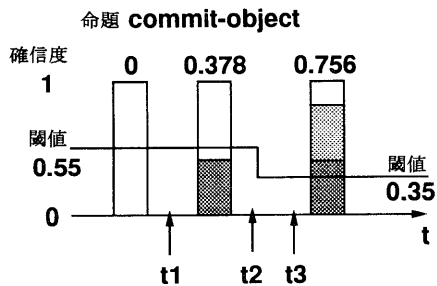
Module: Reply-Not-Identified-Name-of-Object  
condition: user-not-identified-name-of-object  
add: committed-name-of-object  
delete: user-not-identified-name-of-object,  
committed-object  
utter: 「いいえ、その名前ものではありません」

図 7: 反射エージェントを構成するモジュール

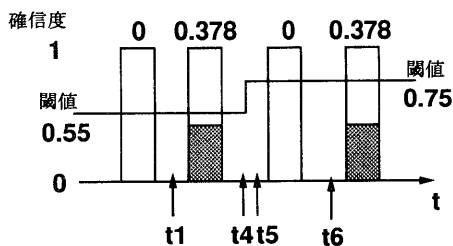
## 6 関連研究

エージェントの即応性と熟考性を統合しようとする試みを述べ、本研究との比較を行う。PRS(Procedural Reasoning System)[4]は、実時間操作のための行動決定メカニズムであり、KA(Knowledge Area)と呼ばれるプランオペレータに相当するものが、外界だけでなく内部状態も変える。状態の知識は一階述語論理で記述され、KAの前提条件とマッチしたものが intention structure に格納され、意図を表現する。一からプランニングするのではなく、知識がないときでも、Interpreterにある程度手続きが埋め込まれているので、実時間での決定が可能になる。KAにはメタレベルKAという intention structure に格納されているKAを選択する機能を持つKAがあり、十分な知識が与えられている時は、メタレベルKAが制御を握り、適切な選択を行うことで、即応と熟考を統合している。

また、Phoenix[5]では、エージェントは reflexive component と cognitive component を持ち、それぞれセンサへのアクセスとエフェクタの制御が可能である。reflexive component はセンサによってトリガーが



- t1 パネルの下に (熟考エージェント)  
 t2 応答1 ボタンですよ (ユーザ)  
 t3 はい、その名前のもので (反射エージェント)



- t4 ダイアル1 ボタンですか? (ユーザ)  
 t5 いいえ、その名前のものでありません (反射エージェント)  
 t6 応答1 ボタンという (熟考エージェント)

図 8: 命題の確信度の変化と閾値

かかり、短い時間の反射的行動を生成する。cognitive component は骨格だけのプランの生成、実行と予測、スケジューリング、モニタリング、エラーリカバリ、再プランニングを担当する。行為は cognitive component 上の TIMELINE に順序付けされずに展開され、スケジューラが次に実行すべき行動、実行に要する時間、実行方法を決定する。

しかし、いずれもドメインが対話であると仮定すると、問題が生じる。PRS はメタレベル KA の設計をあらゆる場合を想定して作り込まなくてはならず、ボトムアップに生成されると考えられる自然な発話が生成できない。Phoenix は、reflexive component がいくら早く反応しても、cognitive component のタイムスパンが長過ぎてプランニングが最後まで終わらない限り、行動に移すことができない。

## 7 まとめ

本研究では、反射的ネットワークと熟考的ネットワークとを設けることとユーザの理解度を連続的な数値にすることで、人間同士の対話に見られるような collaboration を人間対計算機で実現した。

今後はマルチモーダル対話に向けてこのモデルを適用し、モードが多様になるとどのような問題が生じ、どのような拡張が必要なのかなど、モデルのさらなる洗練を図りたい。

## 参考文献

- [1] Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a Collaborative Process. In P. Cohen, J. Morgan, and M. Pollack ed. *Intentions in Communication*, pp.463-493, MIT Press, 1990.
- [2] Philip R. Cohen and C. Raymond Perrault. Elements of a Plan-Based Theory of Speech Acts. *Cognitive Science*, 3(3):177-212, 1979.
- [3] Pattie Maes. How To Do The Right Things. *Connection Science*, Vol.1, No.3, 1989.
- [4] M.Georgeff. Decision-Making in an Embedded Reasoning System. *IJCAI-89*.
- [5] P. Cohen, Michael L. Greenberg, David M. Hart, and Adele E. Howe. Understanding the Design Requirements for Agents in Complex Environment, *AI-Magazine*'89.
- [6] Agre, P. E. and Champman, D. Pengi: An Implementation of a Theory of Activity, *AAAI-87*, pp. 268-272, 1987.