

対話参加者のもつプランゴール構造の差異を考慮した 対話制御方式について・

今村 誠 小谷 亮 近藤 省造
三菱電機株式会社 情報電子研究所

本稿では、ユーザの支援を目的とする対話の制御方式について検討する。このような対話は、ユーザとシステムがお互いに必要とする情報をやりとりすることによってなされる共同の問題解決過程とみなすことができる。このような対話を行なう際には、ユーザとシステムの意図構造に応じた応答を生成することが重要になる。ここでは、この意図構造をプランゴール構造(ゴール、ゴールを達成するためのプラン、プランニングの際になされた仮定からなる構造)を用いて表現した。また、推定されたユーザのプランゴール構造をシステムのプランゴール構造と比較し、その結果をもとに応答を生成する方式を提案する。

A METHOD FOR CONTROLLING DIALOGUE BASED ON DIFFERENCE BETWEEN PLAN-GOAL STRUCTURES OF DIALOGUE PARTICIPANTS **

Makoto IMAMURA Akira KOTANI Shozo KONDO
Mitsubishi Electric Corporation
Information Systems & Electronics Development Laboratory
5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan
e-mail address: imamura@isl.melco.co.jp

In this paper, a method for controlling dialogue which aims to support user's goal achievement is discussed. This kind of dialogue is regarded as cooperative problem solving by exchanging information with each other. In order for the system to generate cooperative responses to a user, the system should take the difference between user's intentional structure and system's one into account. This intentional structure is represented by plan-goal structure. The plan-goal structure consists of goals, plans to achieve the goals, and hypothesis to be used in making the plans. In this paper, we focus on the problem solving to continue the dialogue. We propose a method to generate responses by comparing the recognized user's plan-goal structure and the system's one.

*本研究は、ICOTからの委託により第五世代コンピュータプロジェクトの一環として行なっている。

**This work was supported by ICOT(Institute for New Generation Computer Technology).

1 はじめに

ユーザのゴールの達成を支援することを目的とする対話の処理は大きく分けると次の二つからなっている。

- ユーザのもつゴールや知識を推定すること。
- 推定された結果をもとに、与えられた場面に適合した応答を生成すること。

このような対話を行なうシステムを構築するためには、構文解析、意味解析、プラン認識、知識表現、ユーザモデル、問題解決、文生成などの多くの技術が必要とされる。それらのうちの多くは、文章理解や対話理解を扱う際に必要な技術と似ているが、対話を行なうシステムに特有の点としてつぎのようなものがある。

- 解析が完全に成功しない時に相手に尋ねることができる。
- 生成に不備な点があっても、相手側がその点を指摘してくれることを期待して対話を進めることができる。
- 自分の発話を工夫することで、つぎの相手の発話に制限を加えることができる。

ここでは、以下のような観点から対話特有のインタラクションの問題を扱うことにする。

ユーザのゴールの達成を支援することを目的としてなされる対話は、ゴールを達成するためお互いが必要とする情報をやりとりすることによって実現されるユーザとシステムの共同による問題解決過程とみなすことができる。このような対話では、ユーザとシステムが解くべき問題に課される条件や解法について合意に達する必要がある、そのような合意を形成するために行なわれる対話が重要になる。合意を形成するためになされる発話には、“相手の知識を確認する発話”や“相手と自分で矛盾する知識を指摘する発話”や“相手になくて自分にある知識を伝達する発話”や“自分になくて相手にある知識の伝達を要求する発話”などがある。そして、ユーザの目標を達成する際に関係のある知識が対話参加者同士で食い違いがあると判断した場合に、このような合意を形成する発話を生成することが重要になる。そこで、このような発話を生成するための手続きについて調べることにした。

以下のような順序で考察を進める。2章では、対話例文を解析するための上台となる対話処理のモデルを提案する。3章では、2章で提案した対話処理のモデルの観点から対話例を分析する。4章では、その分析結果をもとにして対話処理の手続きを提案し、具体的な例を示す。5章では、今後の課題について述べる。

2 対話処理のモデル

この章では、ユーザの目的の達成を支援することを目標としてなされる対話処理のモデルについて述べる。ユーザとシステムの共同による問題解決を行なうためには、どのような問題を解きたいのか、またどのような条件のもとで解くのか、またどのような手順で解こうとしているかを表現する必要がある。ここでは、そのような知識を以下で説明するようなプランゴール構造で表現し、プランゴール構造をインクリメンタルに書き換える作業として対話処理をとらえることにする。

2.1 行為や状態に関する知識の表現

対話をモデル化する際に、発話を相手のゴールや知識に影響を与えるためになされる行為として扱う方法[4]がある。このようなものの見方をすることにより、発話を AI でよく研究されているプランニングの表現を用いて記述することができる。このことによって、対話システムにおける問題解決や発話の解析や生成の処理を統一的に扱うことが可能になった[7][1]。ここでは、行為や状態に関する知識を以下のように表現する。

① 行為の表現

行為は次のようなスロットと値の対の集合で表現される。

header : 行為

precondition : header を実行する際の前提条件

decomposition : header を分解した行為列

effect : header 実行後の状態を特徴づける命題

これは、[header の行為を実行するためには、precondition の条件が成立していることが必要である。また、その行為は decomposition の行為列を実行することによって達成される。そして、その行為列の実行後は effect の命題が成立する状態になる。]という知識を表現している。例えば、情報を伝達する行為はつぎのように表現される。

(例) 情報を伝達する行為の表現

header : inform(A, B, P)

precondition : know(A, P)

decomposition : utter(A, B, P)

effect : know(B, P)

② 状態を引き起こす状態の記述

同様に、状態を引き起こす行為は次のように表現される。

header : 状態

precondition : effect の状態を引き起こす前提条件

effect : header の状態から引き起こされる状態

これは、[header の条件を満たす状態であれば、precondition の条件が成立する場合には、effect の条件

を満たす状態になる」という知識を表現している。
 例えば、「ある状態 G が成立して欲しいと B が思っていることを A が知っている時に、A が B に対して協調的な関係にある場合は、A も状態 G が成立して欲しいと思う」という知識は以下のように記述される。

(例)

header : believe(A, want(B, G))
 precondition : believe(A, cooperative(A, B))
 effect : want(A, G)

③ 行為を引き起こす状態の記述

同様に、行為を引き起こす状態は以下のように表現される。

header : 状態
 precondition : effect の行為を引き起こす前提条件
 effect : header の状態から引き起こされる行為
 これは、「header の条件を満たす状態であれば、precondition の条件が成立する場合には、effect にある行為が実行される。」という知識を表現している。
 例えば、「A が行為 Ac を実行したいと思っている時に、Ac が実行できると思う場合には実行する。」という知識は以下のように記述される。

(例)

header : want(A, Ac)
 precondition : believe(A, can-do(A, Ac))
 effect : do(A, Ac)

④ 状態の詳細な情報に関する知識の記述

同様に、状態の詳細な情報に関する知識は以下のように表現される。

header: 状態
 decomposition: header を分解した状態
 これは、「header の条件を満たす状態であるときには、decomposition の状態に関する条件が成立する。」という知識を表現している。このような知識は、扱われる状態に関する制約を表現する知識の細かさを調節する際に用いられる。

2.2 ブランゴール構造

ブランゴール構造は、対話を通じてなされる問題解決過程を表現する際に用いられるデータ構造であり、次のものから成り立っている。

- ゴール
- ゴールを達成するために作られたプラン
- プランを作る際になされた仮定

ブランゴール構造は、行為や状態に関する知識を用いて作られ、状態を特徴づける命題と行為をノードとするラベル付き有向グラフによって表現される。具体的には次のものから成り立っている。

- 行為 \xrightarrow{effect} 状態に関する制約
- 状態に関する制約 \xrightarrow{effect} 行為
- 行為 $\xrightarrow{precondition}$ 状態に関する制約
- 行為 $\xrightarrow{decomposition}$ 順序制約付き行為列
- 状態に関する制約 $\xrightarrow{decomposition}$ 状態に関する制約

但し、対話処理のモデルにおいてブランゴール構造が用いられる際には、グラフのノードにいくつかの情報を持たせる必要が生じる²。

2.3 問題解決的な対話処理のモデル

本稿では、対話に課せられた拘束条件をなるべく満たすように、「聞き手としての問題解決」と「話し手としての問題解決」を行なうこととして、対話処理をモデル化する。ここでは、この処理を行為に関する知識を用いて、ブランゴール構造を表現するグラフを書き換えることによって行なう。

① 聞き手の問題解決

与えられた場面と相手の発話から、対話の相手がどのようなブランゴール構造をもっているかを推定し、システム自身のブランゴール構造の中に位置付けようとする³。これは、具体的には、次のような処理からなる。

- 場面のタイプと発話のタイプから、ユーザのブランゴール構造を表現するグラフを作る。うまく構成できない場合には、不明点を尋ねたり、確認したりするためのゴールを生成する。
- 推定されたユーザのブランゴール構造を検証する⁴。妥当でないと判断され、ユーザとシステムの知識に食い違いが発見された場合には、対話を通じて食い違いをなくすためのゴールを生成する。
- 検証の結果、ユーザのブランゴール構造が妥当であると判断される場合には、推定されたユーザのブランゴール構造を表現するグラフをシステム自身のもつブランゴール構造を表現するグラフに連結することを試みる。

② 話し手の問題解決

「聞き手の問題解決」を行なう際に生成された複数のゴールを考慮に入れ、プランニングを行なうことにより

²具体的には、実行された行為かどうか、実現された状態かどうか、ゴールであるかどうか、どのくらい重要なゴールであるか、現在の話題となっている行為や状態かどうか、などを表す属性がノードに付与される

³発話と発話がどのような関係があるかを捉えることは大変重要である。このような位置付けがうまくできれば解析のあいまい性の解消などに役立つ[3]。

⁴合意を達成するために必要である。

システムのプランゴール構造を更新する。そして、更新されたプランゴール構造中で実行すべきだと思われる行為を実行する。これは、具体的には、次のような処理からなる。

- 行為と状態に関する知識を用いて、システムのプランゴール構造を表現するグラフを書き換える。
- システムのプランゴール構造を表現するグラフ中のシステムの行為のノードのうちで実行すべきであると判断したものを実行する。そして、行為の実行後に、実行された行為に関係するノードの属性を書き換える。

③ 対話に課せられた拘束条件⁵

話し手や聞き手の問題解決の際に行なわれる処理についてのべたが、その処理をどのように制御するかは必ずしも明らかでない。このような作業を実際に行なう際の制御情報として次のようなものがあると考えた。

- 発話のプランニングにはあまり時間がかかってはいけない。
- 発話を実行する時間はあまり長くはいけない。
- 発話は自分の目標の達成に役立つねばならない。
- 発話は相手の目標の達成に役立つ方がよい。
- 発話はそれを聞く相手が理解しやすいものでなければならない。
- 発話に含まれる長さ当たりの情報量は多い方がよい。
- 発話は与えられた場面との整合性をもっていなければならない。
- 発話は前の相手の発話や前の自分の発話と適切な関係にある方がよい。

3 対話例の分析

前章であげたようなモデルでは、具体的にどのようなゴールが生成されるのか、どのような行為に関する知識が必要なのか、また書き換えはどのような順序でどのくらい行なわれるについては述べなかった。これらの問題を、個別の例文の場合に調べてみることによって、上記のような問題解決を行なう際に必要な知識や推論について検討する。

3.1 ユーザの誤解を訂正する発話の生成

推定されたユーザのプランゴール構造を検証した時に、システムの知識との食い違いが発見され、かつシ

⁵[2]を参考にした。

ステム自身の知識の方が正しいと判断した場合には、誤解を訂正する発話を生成することが望ましい。誤解を訂正する発話とは、異なっているユーザの知識をシステムの知識で置き換えることを目的とする発話のことをさす⁶。例えば、次のような電話の応対における対話例の場合の問題解決について調べてみる。

● 対話例 1.1

以下にあげる対話例では、Sはシステム、Uはユーザの略である。本稿の中では、システム側に向けた対話の問題解決を扱う。

(1)U: bと申しますが、aさんはいらっしゃいますか。

(2)S: aでございますね。

(3)U: はい。

(4)S: aは出張しておりますが、4時頃には戻る予定です。こちらからお電話さしあげるようにいたしますでしょうか。

この場合、システムはユーザの(1)の発話から、例えば次のようなことを推論できればよい。

- ユーザはaさんと話をするというゴールをもっている。
- ユーザはaさんがシステムの近くにいらっしゃると思っ

(4)の発話を生成するために、システムは次のような「話し手の問題解決」を行なう。

- 「aさんが近くにいるという知識」が、システムとユーザで食い違いをおこしていることを発見する。そして、「aさんは出張している」という知識の方が正しいとシステムは判断し、間違いを正すための発話を計画する。
- ユーザのゴールがすぐには達成できないとわかった場合、ユーザのゴールの達成を手助けするためのゴールを生成する。そのゴールを解くためのプランニングを行なって、提案を行なう発話を実行する。

3.2 ユーザとシステムの知識の食い違いを解消しようとする発話の生成

推定されたユーザのプランゴール構造を検証した時に、システムの知識と食い違いが発見され、かつユーザの知識の方が正しいと判断された場合には、異なっているシステムの知識をユーザの知識で置き換え、そして置き換えたことを伝達することを目的とする発話(合意を形成するためには、同意したことを伝達しなければならない)を生成することが望ましい。たとえば、次のような場合である。

⁶この場合については、[8][9][5]などで扱われている。

● 対話例 2.1

- (1)S: a さんのお宅ですか。
- (2)U: いえ、違います。
- (3)S: すいません。間違いました。

(3) のようなシステムの発話を生成するには次のような問題解決を行なえばよい。

- 「a さんの家に電話をかけている」という命題の真偽値に関してシステムの知識とユーザの知識に食い違いがあることを発見する。
- ユーザの知識の方が正しいと判断する。
- 正しくないと思われる自分の知識を正しいと思うものに置き換える。
- 「a さんの家に電話をかけている」という知識の真偽値に対して合意ができたことをユーザに伝える⁷。

推定されたユーザのプランゴール構造を検証した時に、システムの知識と食い違いが発見され、かつどちらが正しいがよくわからない場合(かつ、システムやユーザのゴールの達成に関係がある場合)には、正しい知識が何かを明らかにすることを目的とする発話を生成することが望ましい。たとえば、次のような例の場合である。

● 対話例 2.2

- (1)S: a さんのお宅ですか。
- (2)U: いえ、違います。
- (3)S: 本当にそうですか。
(電話番号は、01-2345-6789 ですか。)

(3) のようなシステムの発話を生成するには次のような問題解決を行なえばよい。

- 「a さんの家に電話をかけている」という命題の真偽値に関してシステムの知識とユーザの知識に食い違いがあることを発見する。
- どちらが正しいがよくわからないと判断した場合には、どちらが正しいかを明らかにするような発話(直接尋ねる発話、発話を生成した際に作られるプランゴール構造の食い違いの箇所を同定しようとする発話)を生成する。

このように、推定されたユーザのプランゴール構造を検証した時にシステムの知識との食い違いを発見した場合に、どちらが正しいと判断するかによって対話はかなり違ったものになるし、また対話の目標の達成

⁷「すいません。」という発話の目的は必ずしもこの点だけではないが、ここではユーザとシステムの知識の食い違いとその解消という観点から分析した。このような発話の生成には語用論的な知識が特に重要になる。

に与える影響も大きい⁸。従って、上にあげたような対話を実現するには、食い違いが生じた場合にどちらが正しいかを判断することが必要になる⁹。このような知識の食い違いに対する判断は、一般的にはむずかしいが、ユーザやシステムの知識に対するメタな知識を利用することでいくらか対処できる。たとえば、上記の例文の場合には、知識のタイプと知識の正しさを主張する人と確からしさの三つ組で表現される知識を利用することである程度実現できる。

3.3 システムの前提の誤りの訂正を要求する発話の理解

対話処理のモデルを実現する際には、システム自身のプランゴール構造の細かさや大きさをどのくらいにすればよいかという問題が起こる。例えば、次のような例題を考えてみる。

● 対話例 3.1

- (1)S:
- (2)U: (音が小さくて) よく聞こえないのですが。
- (3)S: それではもう一度いいます。.....(大きな声で)

この例では、例えば次のような問題解決を行なう。

- (1) の発話を生成した際に作られたプランゴール構造において、発話を行なう行為の前提条件の記述(ラベル precondition の値)中の「声の大きさに関する条件」が正しくないため、システムのプランがうまく成功しなかったのだと解釈する。
- システムのプランゴール構造を修正することによって、もう一度発話を行なうという行為を計画する。
- ユーザの発話を理解し、代案を生成し実行したことを相手に伝達する。

このようなプランゴール構造の前提条件の誤りを指摘する発話を理解するために必要な知識はきりが無い¹⁰。したがって、対話に課せられた拘束条件を満たすような問題解決行なって文を生成する際には、必要な場合もあるが必要でない場合が多いような知識はなるべく用いずにプランニングを行い、不都合が生じてから再プランニングを行なうのが望ましい。具体的には、2章で説明したような行為に関する知識を利用する際

⁸例えば、システムがデフォルトとして行なった推論をユーザの指摘によって正す場合などがある。

⁹本当にどちらが正しいかを判定することが重要なわけではなく、とりあえずどちらが正しいかを判断を行なって対話を先に進めることが重要である。

¹⁰「どのくらいの音の大きさがよいか」とか「相手はどのような言語が理解できるか」とか「相手は音を聞くことができるか」などという条件。

に、その知識のどの部分が必要かを選択する問題にあたる。

3.4 生成される発話の評価基準

発話の生成の際に、発話の可能性が複数ある場合がある。実際に行なう発話の一つでなければならないから、その場合にはどの発話がよいかを決める必要がある。ここでは、発話の評価基準のひとつである発話の効率について考察する。発話は目的を達成できる限り簡潔な方が効率が良いが、あまりに簡潔すぎると相手はその発話の不完全さを補うための発話が必要となるためかえって効率が悪くなる。このことは、次のような例からわかる。

システムが車のセールスマンであり、ユーザが客であるとし、システムがユーザの対象となっている車の色を決めようとしている場面を考える。この場面において、システムのゴール「ユーザが好きな色を知る」を達成するために計画された二つの発話「何色にしますか(以下、発話(a)とよぶ)」「赤、白、黒がありますが、何色にしますか(以下、発話(b)とよぶ)」を比較する。

- 対話例 4.1
 - (1)S: 何色にしますか。
 - (2)U: 白にします。
- 対話例 4.2
 - (1)S: 何色にしますか。
 - (2)U: 何色がありますか。
 - (3)S: 赤、白、黒があります。
 - (4)U: 白にします。
- 対話例 4.3
 - (1)S: 何色にしますか。
 - (2)U: 緑にします。
 - (3)S: 申し訳ありませんが、赤、白、黒しかありません。この中では何色がよいですか。
 - (4)U: 白にします。
- 対話例 4.4
 - (1)S: 赤、白、黒がありますが、何色にしますか。
 - (2)U: 白にします。

上記の例文から、次のようなことが考えられる。

- ユーザが赤、白、黒から選択しなければならないことを知っていることをシステムが知っている場合には、(a)が効率的である。(対話例 4.1)
- ユーザが赤、白、黒から選択しなければならないことを知らないことをシステムが知っている場合には、(b)が効率的である。(対話例 4.4)
- それ意外の場合はどちらともいえない。「相手がわからないことがあれば聞き返してくるわけだから、(a)でも十分目的を達成できるので、(a)の

方が簡潔でよい]とも考えられるし、[(b)の方が(a)よりも次のユーザの発話をより強く制限しており、対話システムの解析の付加を軽くすることに役立っている。]とも考えられる¹¹。

3.5 まとめ

以上の分析をまとめると以下ようになる。

- ユーザのプランゴール構造を検証する際に生じる様々なゴールを解決することで、ユーザとの合意を形成することを目的とする協調的な発話を生成できる。
- ユーザとシステムの知識の食い違いの発見は、対話の制御に大きな影響を与える。
- 不都合が発見されない限り粗いプランニングを行い、必要に応じてより詳細な再プランニングを行なう推論が必要になる。
- ゴールを効率的に達成するような発話を生成するためには、ユーザの知識に関する情報が必要である。

4 対話処理の手続きと処理例

4.1 対話処理の手続き

2章で提案したモデルでは、発話の内容は、「聞き手の問題解決」の際にどのようなゴールが生成されるか、またそこで生成されたゴールがどのように解決されるかで決まる。以下に、どのようなゴールが生成されるかを中心に対話における応答生成の手続きについて述べる。

① ユーザのプランゴール構造の推定

- 与えられた場面とユーザの発話からユーザのプランゴール構造を推定する。

② ユーザのプランゴール構造の検証と連結

- (推定された) ユーザのプランゴール構造を検証する。
- 検証した結果、(推定された) ユーザのプランゴール構造が妥当であった場合には、その構造をシステムのプランゴール構造に連結することを試みる¹²。

¹¹ この問題の一面を期待値の世界でとらえることができる。

¹² ここでは、システムはユーザに対して協調的な関係にあると仮定しているので、連結が成功した場合には、ユーザのゴールはシステムのゴールとして扱われる。このことを、ゴールであることを表す属性をそのノードに付加することによって表現する。また、連結するもとなったノードがその発話の話題であることを表現するために、「話題であることを表す属性」を付加する。

③ ゴールの生成

①, ②の処理を行なっている際に、ユーザとの合意を形成するためのゴールが生成される。

①の処理で生成されるゴール

- ユーザのプランゴール構造の推定が成功した時、推定された結果を相手に確認するゴールを生成する¹³。
- ユーザのプランゴールの推定が成功しない時、そのことについてを相手に質問するゴールを生成する。

②の処理に関連して生成されるゴール

- 検証の結果、(推定された)ユーザのプランゴール構造とシステムの知識との食い違いが発見された場合には、それを解消するためのゴールを生成する。
- 検証の結果、(推定された)ユーザのプランゴール構造は妥当であると判断されたが、システムのプランゴール構造にうまく連結できない場合は、相手の発話のゴールを確かめるためのゴールを生成する。

④ 発話の生成プランニング

システムのプランゴール構造の中の未解決ゴールを達成するためのプランニングを行なう。この処理は、システムのプランゴール構造を表現するグラフを書き換える処理として実現される。

⑤ 発話の実行とそれに伴う処理

- 発話の生成プランニングによって作られたプランゴール構造中にあるシステムの行為の中で、まだ実行されていないもので、かつ現在実行可能なものの中から実行すべき行為を選択する。
- 選択された行為を実行する。
- 発話の実行によって引き起こされる(と予測される)状態や行為を表すノードに、実行済み行為や実現済み状態であることを表す属性を付加する。

4.2 処理例

この節では、上記の手続きを実行する際に、プランゴール構造を表現するどのようなグラフが作られるのかを例によって示す。以下、対話例4.2の(2)の発話の処理を説明する¹⁴。

① システムは、発話(1)の生成後、以下のようなプランゴール構造をもつ。

```
(g)know(S,X@好き(U,色,X))
↑effect
```

¹³これらのゴールがすべて発話に結び付くわけではない。対話例1.1の(2)にあるように確認する場合もあるし、そうでない場合もある。対話では何もふれないということで、理解したということ伝えている場合が多い。

¹⁴発話行為に関する知識の記述は[1]を参考にした。*は実行された行為や実現された状態を、(g)はシステムのゴール、Sはシステム、Uはユーザを表す。

```
inform-ref(U,S,X@好き(U,色,X))
```

```
↑effect
want(U,inform-ref(...))
```

```
↑effect
*request(S,U,inform-ref(...))
```

```
↓decomposition
```

```
*utter(S,U,"何色がいいですか")
```

この発話の生成後、次の相手の発話として、inform-ref(U, S, X@like(U, 色, X))を達成する行為を予測する。

② ユーザの発話(2)から次のようなユーザの(局所的な)プランゴール構造を推定する。

```
*utter(U,S,"何色がありますか")
```

```
↑decomposition
```

```
*request(U,S,inform-ref(S,U,Y@何色(Y)))
```

```
↓effect
```

```
want(S,inform-ref(S,U,Y@何色(Y)))
```

```
↓effect
```

```
inform-ref(S,U,Y@何色(Y))
```

```
↓effect
```

```
know(U,Y@何色(Y))
```

(注意: ループを防ぐ必要がある。たとえば、knowとinformはループする可能性がある)

③ システムは、推定されたユーザのプランゴール構造をシステムのプランゴール構造に連結しようとする。

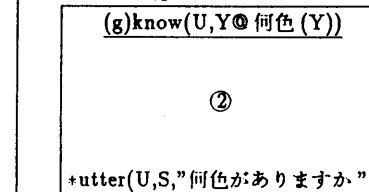
予測通りの行為でないことがわかる。システムは自分のプランゴール構造に不備があると判断し、(1)の生成時のプランゴール構造の前提条件を調べなおす。この場合は、下線の前提条件を追加すると¹⁵、推定されたユーザのプランゴール構造を連結することができることを発見する¹⁶。そして、連結を行なった際に下線部をシステムのゴールとする。

```
(g)know(S,X@好き(U,色,X))
```

```
↑effect
```

```
inform-ref(U,S,X@好き(U,色,X))
```

```
↑precond
```



```
want(U,inform-ref(...))
↑effect
```

```
*request S,U,inform-ref(...)
```

¹⁵実行が遅延されているゴールとしても実現できる。

¹⁶枠で囲まれた部分は、(2)の発話から推定されたユーザのプランゴール構造を示す。

↓decomposition

*utter(S,U,"何色がいいですか")

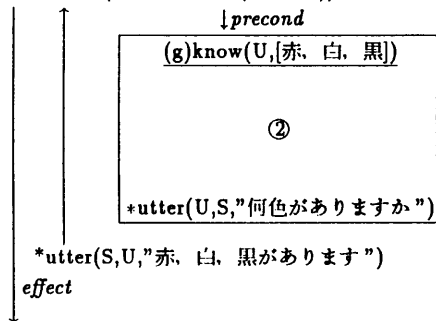
④ システムの発話 (3) の実行
迎結されることで作られたプランゴール構造中の未解決ゴールを解決するために、発話 (3) を実行する。

(g)know(S,X@好き(U,色,X))

↑effect

inform-ref(U,S,X@好き(U,色,X))

↓precond



want(U,inform-ref(...))

↑effect

*requestS,U,inform-ref(...))

↓decomposition

*utter(S,U,"何色がいいですか")

[4] Cohen,P.R and Perrault,C.R.: *Elements of plan-based theory of speech acts*. Cognitive Science 3(3)pp177-212(1979)

[5] 古瀬 蔵, 内藤昭三, 片桐恭弘, 島津明: プラン認識に向けての対話分析, 情報処理学会, 自然言語処理研究会 74-4, pp25-332(1988)

[6] Kaplan,S.J: *Cooperative Responses from a Portable Natural Language Database Query System*, Artificial Intelligence 19(2), PP165-188(1979)

[7] Litman,D.J. and Allen,J.F.: *A Plan Recognition Model for Subdialogues in Conversation*, Cognitive Science 11, pp163-200, (1987)

[8] Pollack, M.E.: *A model of plan inference that distinguishes between the beliefs of actors and observers*, Proc.24th Meeting of the Association for Computational Linguistics, pp207-214(1986)

[9] Quilici,A, Dyer,M and M.Flowers: *Recognizing and Responding to Plan-Oriented Misperceptions*, Computational Linguistics, Vol.14, No.3, pp38-51(1988)

5 今後の課題

現在、このモデルにもとづいて電話の取り次ぎを対象とする対話システムの試作を検討している。その際の課題として次のものがあげられる。

- 発話の機能と表層の発話文の対応関係を表現する行為に関する知識が必要になる。
- ユーザのモデルの構造を明らかにする必要がある。
- 文生成プランニングの際にゴールの重要度を管理する機構が必要になる。

参考文献

[1] Appelt,D.E: *Planning English Sentence* CAMBRIDGE UNIVERSITY PRESS (1985)

[2] Beaugrande.R de and Dressler.W.U: *Introduction to Text Linguistics*, Longman Group Limited, Harlow, Essex.(1981)(テキスト言語学入門, 池上嘉彦他訳, 紀伊屋書店)

[3] Carberry,S.: *Plan Recognition and Its Use in Understanding Dialog*, User Models in Dialog Systems, Springer-Verlag, (1988)