TR-I-0067

対話翻訳のための階層型プラン認識モデル
Tri-Layered Plan Recognition Model for Dialogue Machine Translation

有田 英一、飯田 仁
Hidekazu ARITA and Hitoshi IIDA

1989.2

内容梗概

対等な話者間で交わされる目標指向型対話の理解のためのプラン認識モデルについて述べる。3層の階層的プラン認識モデル(T-Plan Model : Tri-Layered Plan Recognition Model)を提案する。3層とは対話のトピックの構造を表わすドメイン･プラン、対話のやり取りのまとまりを表わすインターラクション･プラン、そしてドメイン･プランとインターラクション･プランを結び付けるディスコース･プランである。ドメイン･プランはドメインに依存する知識であるが、ディスコース･プラン、インターラクション･プランはドメインに依存しない知識である。このモデルは対話翻訳において省略の解釈、訳語選択、次発話の予測などに応用できる。

ATR 自動翻訳電話研究所
ATR Interpreting Telephony Research Laboratories

目次

0. ショート･サマリ 1
1. はじめに 2
2. 自動翻訳電話のための対話のモデル 5
3. 階層型プラン認識モデル 8
(T-Plan Model : Tri-Layered Plan Recognition Model)
3.0 概要 8
3.1 発話の表現 12
3.2 インターラクション･プラン 13
3.2.1 発話タイプの分類 13
3.2.2 発話タイプをもとにした対話対 18
3.2.3 インターラクション･プラン 23
3.3 ディスコース･プラン 24
3.4 ドメイン･プラン 26
3.5 推論規則 28
3.6 プランナ 31
3.6.1 構成 31
3.6.2 基本動作 31
3.6.3 インターラクション･プランから 35
ドメイン･プランへの関連づけ
4. 例 45
5. 自動翻訳電話への応用 55
5.1 省略の解釈 55
5.2 訳語選択 55
5.3 次発話の予測 55
6. 関連研究 57
7. おわりに (問題点･今後の課題) 61
謝辞
参考文献
付録 65
A1. 「国際会議の問合せ」における A-1
ドメイン･プランとオブジェクトの例
A2. 発話タイプ、スピーチ･アクト、 A-7
インターラクション･プランの関係
A2.1 パーサの解析結果と発話タイプの対応 A-7
A2.2 スピーチ･アクトとインターラクション･プラン A-9
の関係
A3. [Allen and Perrault 80] の plan inference rules と A-13
T-Plan Model の推論規則の対応
A4. [Litman and Allen 87] のプランと A-14
T-Plan Model のプランとの関係
A5. プロトタイプ･システムの動作例 A-17
A5.1 動作例 A-19
A5.2 対話構造の例 A-35

A5.3　ドメイン･プラン、ディスコース･プラン、インターラクション･プランの例　..　A-49

0. ショート・サマリ

☆目的：

電話対話の自動翻訳を実現するためには、対話の理解が不可欠である。本研究では対話理解を行なうために基礎となる対話の構造を明らかにし、それを作り出すプラン認識モデルを研究する。そして、このモデルに基づいて対話文の理解に有効な対話構造構成手法を明らかにする。

☆研究の背景：

①発話と意図の関係を明らかにした論文。
[Allen and Perrault 80] J.F.Allen and C.R.Perrault:"Analyzing Intention in Utterances", ARTIFICIAL INTELLIGENCE, Vol.15, p143-178(1980)

②発話とドメイン・プランの関係を明らかにした論文。
[Litman and Allen 87] D.J.Litman and J.F.Allen:"A Plan Recognition Model for Subdialogues in Conversations", Cognitive Science, Vol.11,p163-200(1987)

☆手法：

プランに基づく問題解決的アプローチ (Plan-based approach)

☆オリジナリティ：

3層のプランからなるモデルの提案
対話の Turn-Taking の構造を表わすインターラクション・プランの導入
Task Oriented Dialogueにおける Communicative Act の設定

☆応用 (自動翻訳電話への応用)：

① 省略の解釈 (格要素の省略、述語の省略および代用的な表現)
② 訳語選択 (ドメインを小さくすることでは解決できない訳語選択)
③ 次発話の予測 (字面ではなく、発話内容に関する予測)

☆現時点での状況：

① インターラクション・プラン、ディスコース・プランについては(ほぼ)完了
② モデルの動作を確認するためのプロト・タイプを Symbolics 3620 上に実現

☆問題点・今後の課題：

① 質問の前提が間違っている場合の処理
② 意図的な言語表現が不明確な発話の処理　ー対話対把握の言語情報欠如ー
③ ゴールリストの管理
④ ドメイン・プランの記述基準
⑤ ドメイン・プランの探索
⑥ ドメイン・プラン外の発話の扱い
⑦ 異表現の同義解釈の問題
⑧ 発話解析モジュールとの結合

1. はじめに

本稿では自然な対話を理解する試みとして、対等な話者間で交わされる目標指向型対話の理解のための一つの方法について述べる。ここで理解とは自動翻訳電話を実現するために必要な理解であり、適切な翻訳を生成するのに必要な情報を文脈やドメインの知識から得ることを指す。

(1) [対話コーパスの性質]

ATRでは自動翻訳電話の対話コーパスとして「国際会議の問合せ」の対話を設定して、電話や計算機端末間の対話を収集・分析している[Iida 86, Arita 87]。これらのコーパスは、Allen等[Litman and Allen 87, Allen and Perrault 80]が扱ってきた駅構内でのPassengerとClerkとの一過性の強い対話とは異なる。またCohen [Cohen 84]のエア・ポンプの組立て指導やGrosz等[Grosz 86]の機械部品修繕の指導における師弟間の対話では、実際の行動とそれに伴う状態の変化とが発話内容に影響してくると考えられるので、我々の対象としているコーパスとは異なる。収集・分析した端末間対話において多用される言語的特徴を日本語についてまとめると次のようになる[Iida 87]。

1. 接続詞を使った複文の用法と、主節を省略した条件節だけの婉曲表現
   例：「会議に申し込みたいのですが。」
2. 旧情報等の省略
   例：「それでは、21部送ってください。」
3. 文末に現れる意図の表現
   例：「お名前をお願いします。」
4. 疑問文型による意図の表現
   例：「ご住所をお聞かせねがえますか。」
5. 対話固有の応答形式
   例：「それでは、21部送ってください。」
   「はい、承知いたしました。」
6. 社会・慣習的な固定表現
   例：「宜しくお願いします。」
7. 主題を中心に据えた表現 (いわゆる「ダ文」)
   例：「いいえ、まだです。」

(2) [対話の理解がないと機械翻訳が困難な例]

対話の理解がないと適切な機械翻訳が困難な例として、省略、訳語選択、照応に関する例を示す。

① 省略 (名詞句および動詞句の省略)

(例1) (主題となっている名詞句が省略される場合、および動詞句部分が省略され、それに対応するものがそれまでの発話の履歴に現れていない例)

(d1-1) 質問者： 原稿の締切りはいつですか?
(d1-2) 事務局： 12月25日ですので
(d1-3) お急ぎ下さい

Dialogue 1

(d1-2)の発話は、「ＡはＢだ」という文法的な型をとる文において、Ａの部分が省略された文になっていて、前発話で提示された主題「締切り」が省略されている。この発話をそのまま英訳する場合、述部をもたない名詞句単体にして"Ｂ"とする†か、暗黙的に"it is"を付加して"it is Ｂ"とする方法がとられることが多い。しかし、その場合、聞き手側に解釈の曖昧さを残すか、もしくは代名詞の使い方が誤り、その照応を適切に捉えられないことになる。英文生成時に的確な訳文を生成できるようにするため、省略されたＡの部分が何であるか解釈することが必要となる。

(d1-3)の発話では何を急ぐのかそれまでの発話の履歴には現れていない。原稿は締切りまでに申し込み者が事務局に送るというドメイン知識がないと(d1-3)は解釈できないので"Please return it as soon as possible."のような訳を生成できない。この例は(d1-2)のように格要素の省略([原稿の締切りが]12月25日です)ではなく述部の省略であることに注意する必要がある。

② 訳語選択

(例2)　(発話の解釈により、適切な訳語を選択する必要がある例)

(d2-1) 質問者：　支払はどうすれば良いですか。
(d2-2) 事務局：　銀行振り込みです。
(d2-3)　　期限は9月12日です。

Dialogue 2

一発話の解析に基づいた直訳は、日本語および英語における文脈が等価に保存されていることを前提にしないと両言語における解釈が異なることになる。(d2-3)の「期限」は「支払の期限」の意味である。このことが分からないと、"Payment should be made by September 12.", "The deadline for payment is September 12."のような訳を生成する場合の"payment"という語の選択はできない。また、「期限」の訳語は一般的にはdeadline、あるいはtime limitであるが、「締切時間」という意味でdeadlineが選択されるべきである。

(例3)　(一発話ユニット††全体を解析しなければ訳語が決まらない例)

次の例[Kume 88]では、一発話の解析だけでは正しい翻訳が得られないことを指摘している。

(d3-Q)：登録用紙をお送りしましょうか。
(d3-A1)：はい、有り難うございます。
(d3-A2)：はい、有り難うございます。よろしくお願いします。
(d3-A3)：はい、有り難うございます。今は結構です。

Dialogue 3

質問(d3-Q)に対して、応答(d3-A1)、(d3-A2)の「有り難うございます」は「送って下さい / Thank you.」の意味であるのに対して、(d3-A3)は「送らなくてもいい / No, thank you.」の意味である。

(注) †noun phrase fragmental sentenceと呼ぶ文の体裁をなさない名詞句単体の発話を指す。
††発話ユニットとは一人の話者が1回のターンで行なった発話の集まりである。

(例4)　(対話のターン・テイキングを考慮しなければ訳語が決まらない例)

(d4-1) 質問者：　学生を20人程連れて行きたいのですが、料金の割引は、ないでしょうか。
(d4-2) 事務局：　学生の方に関しましては、登録費は1000円となっていますが、団体割引というようなことは、あいにくですがございません。
(d4-3) 質問者：　分かりました。
(d4-4)　申込み用紙を21部送って下さい。
(d4-5) 事務局：　分かりました。

Dialogue 4

(d4-3)、(d4-5) は共に、確認の発話であるが、承知と了承の違いがあり、情報提供の発話に対し承知する対話の対、および行為依頼に対しその行為遂行を了承する対話の対において、その違いが明らかになる。(d4-3)の「分かりました」は"I see." または "I understand." という承知の応答であり、了承の応答である"All right." は不適切である。しかし (d4-5) では"I see." は不適切である。

③ 照応

(例5)　(単純なスタックを利用した照応解析では照応に失敗する例)

(d5-1) 質問者：　ところで参加料はいくらですか。
(d5-2) 事務局：　10000円です。
(d5-3) 質問者：　ホテルの宿泊費は別ですか。
(d5-4)　その費用には何が含まれていますか。

Dialogue 5

(d5-4) の「その費用」が直前の文の名詞句「宿泊費」ではなく、(d5-1) の「参加料」と照応していることが分からなければ"the fee"とは訳せない。「費用」の訳語は一般的には expense, cost である。

本稿では、まず第2章で自動翻訳電話のための対話のモデルについて述べる。そこで、通訳電話の形態について述べる。またQuestion-Answerシステムなどの対話のモデルとの違いについて述べる。次に第3章で階層型のプラン認識モデルについて述べる。対話のターン・テイキングを表わすインターラクション・プラン、インターラクション・プランとドメイン・プランを結び付けるディスコース・プラン、対話のトピックの構造を表わすドメイン・プランについて述べる。また推論規則とプランナの動作について説明する。第4章で具体的にシステムがどのように動作するか例をもとに具体的に説明する。第5章で本システムの自動翻訳電話への応用について述べる。自動翻訳電話のための省略の解釈と訳語選択、音声認識部のための次発話の予測について述べる。第6章で関連研究について述べる。第7章で本研究の問題点・今後の課題について述べる。

2. 自動翻訳電話のための対話のモデル

自動翻訳電話のための対話のモデルを図1に示す。あるドメインの知識を有する話者Aとそのドメイン知識を得ようとする話者Bが行なう対話をシステムが解釈しようとするモデルである。対話の知識、推論規則、対話のある時点でのゴール、及び対話によって得られた情報は話者A、話者B、システムに共通であるものとする。ドメインの知識は話者Aとシステムが共有しているものとする。対象とする対話は、固定領域の話題に関する自由な対話であり、話し手がとる発話の方策などを予め限定しておくものでない。つまり、問合せの話し手が何をどのように質問するかというような知識を使ってシステムが発話の理解をするのではなく、一般的な領域の知識や対話実施に関する約束事などに基づいて理解を進めるモデルを考える。

また、このモデルは一方の話者の発話が正しく機械翻訳され他方の話者に伝わることを前提にしている。即ち、誤訳によって対話が複雑になる現象を扱うことは今後の課題とする。

このようなモデルを仮定した理由は次のとおりである。

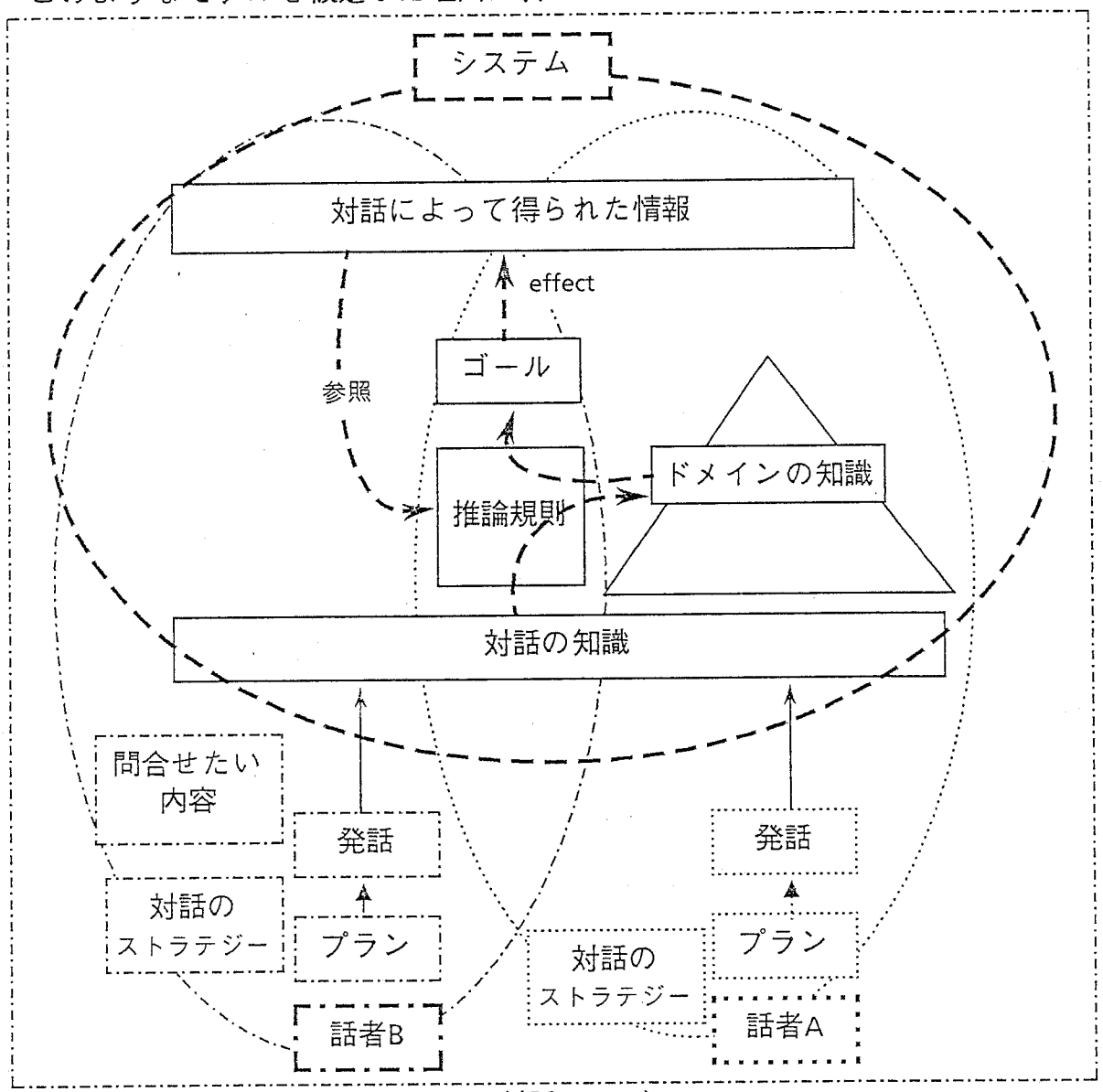


図1　対話のモデル

① ドメインを限った対話システムを考える場合、予めドメインの知識をシステムに用意しておくことが可能である。
② 不特定の話者である質問者(話者B)がどこまでドメインの知識を持っているか予め知ることはできない。
③ 不特定の話者である質問者(話者B)がどのようなストラテジーで発話を組み立てるか予め知ることはできない。またそれに対応して事務局(話者A)がどのような対話のストラテジーをとるか予め知ることはできない。

[通訳電話の形態]

通訳電話の形態としては、次のものが考えられる。発話の曖昧さ†の解消のために通訳が発話者に尋ねるかどうかという観点から、次の分類ができる。

Ⓐ 通訳と発話者の間でローカルな対話をするもの、
Ⓑ 通訳と発話者の間でローカルな対話をしないもの。

また、翻訳の単位の観点から、次の分類ができる。

㋐ 通訳が1発話を聞く毎に、それを翻訳する、
㋑ 通訳が数発話を聞いて、それらを翻訳する、
㋒ 通訳が数発話を聞いて、それらを要約して翻訳する。

ⒶⒷと㋐㋑㋒の組合せによって6種類の通訳電話の形態が考えられる。人間の通訳者はⒶと㋑または㋒の組合せの形態で通訳を行なっているようである。本稿では次の理由からⒷと㋑の組合せの形態の自動通訳電話を想定している。

① 自動翻訳電話における発話の曖昧さには音声認識レベルのものと、発話の内容のレベルのものがある。現在の音声認識技術では100%の認識はできないので、曖昧さの解消のためのローカルな対話を行なうと、そのローカルな対話にも音声認識の問題が生じる可能性がある。

(但し、音声認識レベルの曖昧さの解消のためには、コンピュータがディスプレイに音声認識した結果を表示して、人間がキーボード等の音声以外の手段や、「はい」「いいえ」等の100%音声認識できる言葉で応答するという形態も考えられる。)

また発話内容のレベルの曖昧さを発話者との対話で解消することは、コンピュータによる発話の解釈をあるレベルで止めることになる。本稿では、コンピュータによる発話の解釈が研究の中心課題であるのでローカルな対話は行なわないことにする。

② 第1章の例3で述べたように、対話では1発話の解析だけでは訳語が決まらないことがある。また対話内容の要約は技術的課題が多いことに加えて、対話においては、従来の文書の要約技術の判断基準ではあまり意味がないような情報が重要である場合もある。

以上のほかにも、聞き手に通訳の発話だけを聞かせるのか、あるいは話し手と通訳の両方の発話を聞かせるのかという観点がある。これは、翻訳の処理や発話者と通訳の間の

(注)†ここでいう曖昧さとはコンピュータが一意に可能性を決定できないことをいう。

ローカルな対話に時間がかかり、聞き手にとって無音の時間が長くなると聞き手が不安になるので望ましくないといった認知科学的な観点から検討しなければならない。

[データ･ベース検索などのための対話のモデルとの違い]

データ･ベース検索などのための対話のモデル[Katou 88, Kaplan 83]と本稿で想定している自動翻訳電話のための対話のモデルの違いは、前者は1人の話者(人間)と計算機の対話であり計算機が対話の主導権を持つことができるのに対して、後者は2人の話者(人間)の対話を計算機が理解し、翻訳するもので、対話の流れや発話の仕方を制御することは、対話者間に介在するシステムの透明感を失うことになる。計算機主導型の通訳システムも考慮すべきモデルだが、以上の理由から、ここではシステムが主導権を持たないモデルを考える。計算機が対話の主導権を持つことができないことである(図2参照)。この違いによって、計算機に予め用

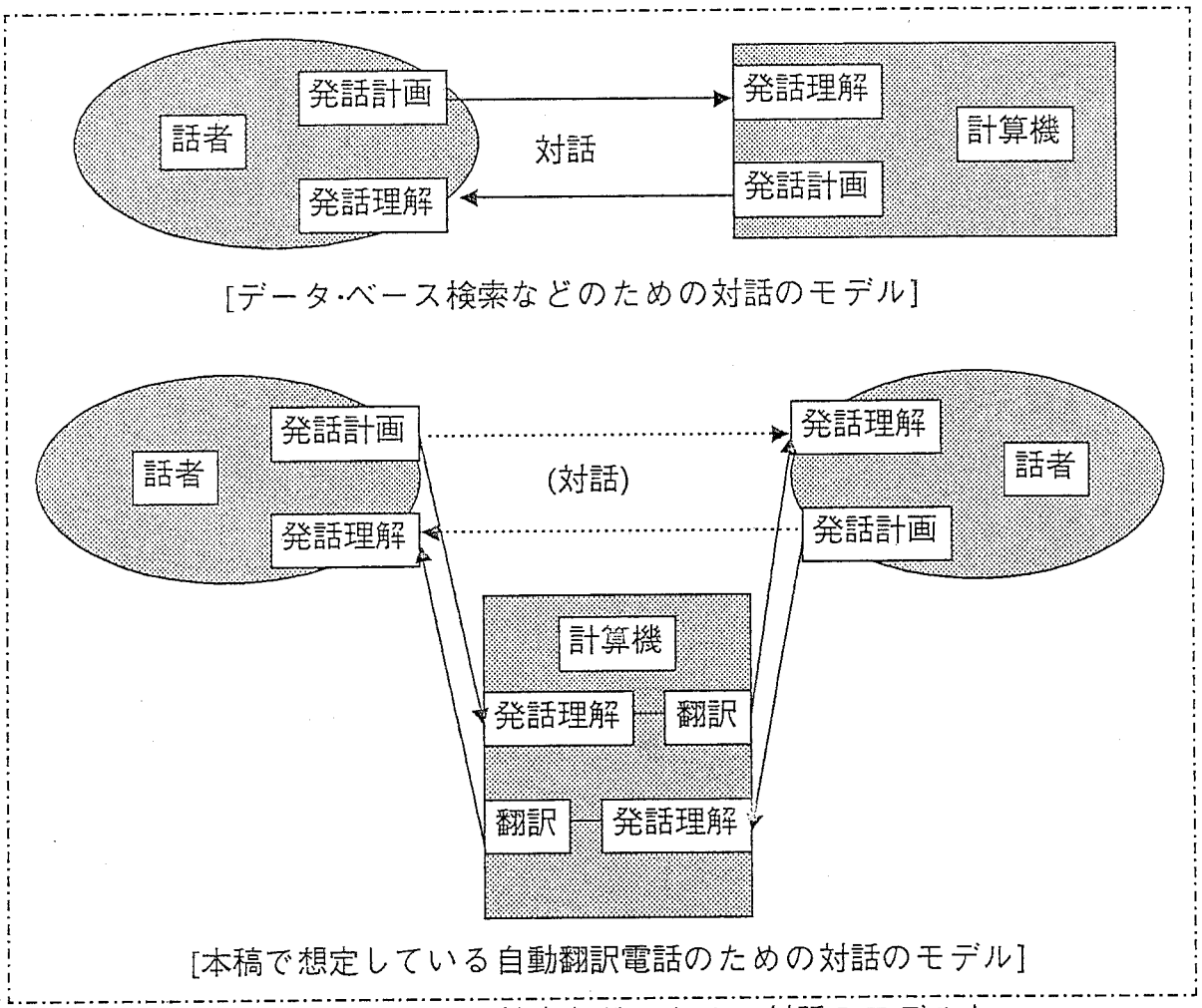


図2　データ･ベース検索などのための対話のモデルと
本稿で想定している自動翻訳電話のための対話のモデル

意されていない知識について発話がなされた場合、計算機のとりうる戦略が変わる。前者のモデルでは、発話計画によって計算機がその持っている知識の範囲を越えないように対話を制御する。後者のモデルでは、計算機が対話を制御しないので発話の表層情報だけから翻訳を試み、その解釈は聞き手(人間)に委ねる。従って、この場合システム上に蓄積される文脈形成に係わる情報が正しく把握さ

れないで、文脈の記述が不十分になる。しかし、知識の不完全さが招く事態であるので、現状これ以上の方策の検討は控える。

3. 階層型プラン認識モデル

3.0 概要

階層型プラン認識モデル(T-Plan Model : Tri Layered Plan Recognition Model)の概略を述べる。プラン認識のモデルはプランとオブジェクトと推論規則から構成される。各発話がもつ意味は、推論規則によりプランに対応づけられて、全体として対話の構造を構成する。

① プラン

プランは因果関係や時間順序関係で結び付けられた行為(action)と状態(state)とのモジュールである。プランは次のスロット†で記述される。

HEADER: プランの名前。

PREREQUISITE: プランが実行されるときに成立していなければならない状態。プランナがHEADERのプランを実行しようとした時にPREREQUISITEの条件が成立していなければ、HEADERのプランを一時保留し、PREREQUISITEの条件を満たすことを確かめる。

DECOMPOSITION: プランを構成するサブ・プランを要素とするプランの集合。プランはHEADERの記述レベルでPREDICATE&CASESで記述できるひとつのアクションと見なせるが、さらに幾つかのアクションのステップに分解できる場合がある。

PREDICATE&CASES: プランを表わす述語とその格要素。

EFFECTS‡: プランが実行された場合の効果。

CONSTRAINTS: プランを実行するときの前提条件となる制約条件。プランナがHEADERのプランを実行しようとした時にCONSTRAINTSの条件が成立していなければ、HEADERのプランを実行しない。

HEADER、DECOMPOSITIONにはアクションが記述され、PREREQUISITE、EFFECTSにはや手続きが記述される。

プランを次の3つに分類する。

(注) †: 他にプランの属するクラスを記述するBELONG-TOスロット、プランの推論のパスを指定するINFERENCE-PATHスロットがある。

‡: ここでは説明のために総称的にEFFECTSとしているが、プログラムでは、
GOAL-EFFECT : プランのDECOMPOSITIONが成功したときに起動される手続き、
ADD-EFFECT : プランが参照されたときに理解状態のリストに加えるステイト、
DEL-EFFECT : プランが参照されたときに理解状態のリストから削除するステイト、
などがある。

①ドメイン･プラン：
対象領域における目標達成のための行為を、階層的な支配関係、および時間順序関係とに基づいて記述し、行為の生起状況を表す領域固有の知識とする。

②インターラクション･プラン：
対話のターン･テイキングのつながりの様子を表わし、対話実行のための局所的な約束事を反映したプランである。このプランはドメインに依存しない。

③ディスコース･プラン：
ドメイン･プランとインターラクション･プランとを関連づけ、対話実行の大局的な約束事を反映したプランである。このプランもドメインに依存しない。

プランを3層に分ける理由は次のとおりである。ドメイン･プランを遂行するための発話の流れは1つとは限らない。例えば、事務局が質問者に登録用紙を送るというドメイン･プランGET-THE-FORMは

事務局：	「登録用紙をお送りします。」	(Will-Do-Action2)†
質問者：	「お願いします。」	(Accept-Offer)

という発話の流れでも実現できるし、また

質問者：	「登録用紙を送って下さい。」	(Request-Action)
事務局：	「はい、分かりました」	(Accept)

でも実現できる。
ドメイン･プランのレベルでは対話の知識ではなく、ドメインの知識に応じたアクションを表現すればよく、またインターラクション･プランのレベルではドメインの知識とは関係なく対話のやりとりに関する知識だけを表現すればよい。ディスコース･プランのレベルで、あるアクションを実行するには、(1)そのアクションのAgentがそのアクションを実行する意志の発話(Will-Do-Action2)をして、Recipientがそれに対して承諾の意の発話(Accept-Offer)をしてもよいし、(2)そのアクションのRecipientがそのアクションを実行してもらいたいという発話(Request-Action)をして、Agentがそれに対して承諾の意の発話(Accept)をしてもよいということを表現すればよい。このことによって、各発話とドメイン･プランの関係が明らかになる。
以上のことを実現するプランの記述の例を下に示す。(?○○○は変数を表わす。)

ドメイン･プラン

```
HEADER:            (GET-THE-FORM ?sp1 ?sp2 ?form)
PREREQUISITE:      (pred&cases (predicate KNOW)(agent ?sp1)
                                   (object address[?sp2])(identfier ?address))
DECOMPOSITION:     (Execute-Domain-Action ?sp1 ?sp2 ?action)
PREDICATE&CASES:   ?action
CONSTRAINTS:       ?action =  (pred&cases (predicate OKURU)
                                   (agent ?sp1)(recipient ?sp2)
                                   (object ?form)(to-loc ?address))
```

(注)† これは次節で述べる発話タイプである。発話タイプは発話の持つ種々の機能のなかでターン･テイキングの機能に注目して設定されている。

(説明) 話者?が話者?sp2に登録用紙?formを送るには、前提条件として話者?は話者?sp2の住所address[?sp2]を知っていなければならない。またこのアクションを記述する述語とその格要素は(pred&cases (predicate OKURU)(agent ?sp1)(recipient ?sp2)(object ?form)(to-loc ?address))であるということを表わしている。

ディスコース・プラン

HEADER: (Execute-Domain-Action ?sp1 ?sp2 ?action)
DECOMPOSITION1: (REQUEST-ACTION-UNIT ?sp2 ?sp1 ?action)
DECOMPOSITION2: (Will-Do-Action2-Unit ?sp1 ?sp2 ?action)

(説明) あるドメインのアクションの実行に対応するする対話のやりとりには、話者?sp2が話者?sp1にアクション?actionをするよう依頼(要求)するものと、話者?sp1がアクション?actionをする意志を表明するものがあるとういことを表している。

インターラクション・プラン

HEADER: (Will-Do-Action2-Unit ?sp1, ?sp2, ?action1)
DECOMPOSITION: (Will-Do-Action2 ?sp1 ?sp2 ?clue ?topic ?action1)
(Accept-Offer ?sp2 ?sp1 ?clue2 ?topic2 ?action2)

(説明) 話者?sp1がアクション?action1を行なう意志を表明し、話者?sp2がそれを受け入れる対話のやりとりを表わす。

HEADER: (REQUEST-ACTION-UNIT ?sp1 ?sp2 ?action1)
DECOMPOSITION: (Request-Action ?sp1 ?sp2 ?action1)
(Accept ?sp2 ?sp1 ?action2)

(説明) 話者?sp1が話者?sp2にアクション?action1を実行するように依頼して、話者?sp2が受諾する対話のやりとりを表わす。

(各プランについては3.2節、3.3節、3.4節で説明する。)

② オブジェクト

オブジェクトは対話で言及される対象物の一般的記述であり、言及されることにより対象物の実態(discourse entity) が構成される。オブジェクトは各種のプロパティとその値で記述される(付録A1参照)。オブジェクトはドメイン・プランのPREDICATE&CASES スロットの格要素となる。オブジェクトとドメイン・プランはディスーコース・プランのEFFECTSの1つである

GOAL-EFFECT : (focus-plan ?object ?domain-plan)

を介して結び付けられている。focus-plan はオブジェクト ?object が格要素となっているドメイン・プラン ?domain-plan をみつけだして、それをゴール・リストの先頭に置く関数である。(ゴール・リストは発話を解釈する時点での対話のゴールを保持するリストである。入力された発話はゴール・リストの先頭のゴールから順にそれをゴールとする発話と解釈できるかどうか調べられる。)

③ 推論規則

推論規則には①応答に関するものと②発話タイプに関するものと③プランに関するものがある。(推論規則については3.5節で説明する。)

④ 対話構造

対話構造とは、発話と各プランとを明確な意味を持つリンクで結び付けた時にできる構造である。この構造はtree構造となる。対話構造により、各発話が対話全体の中で果たす役割が明確になる†。

対話構造の概念図を図3に示す。発話からインターラクション·プランへのリンクはDECOMPOSITIONのリンクである。インターラクション·プランからディスコース·プランへのリンクはDECOMPOSITIONのリンクである。ディスコース·プランからドメイン·プランへのリンクは、(1)直接ドメイン·プランへ張られるリンクはDECOMPOSITIONのリンクであり、(2)オブジェクトを介して張られるリンクはGOAL-EFFECTのリンクである。ドメイン·プランからドメイン·プランへのリンクはDECOMPOSITIONのリンクである。

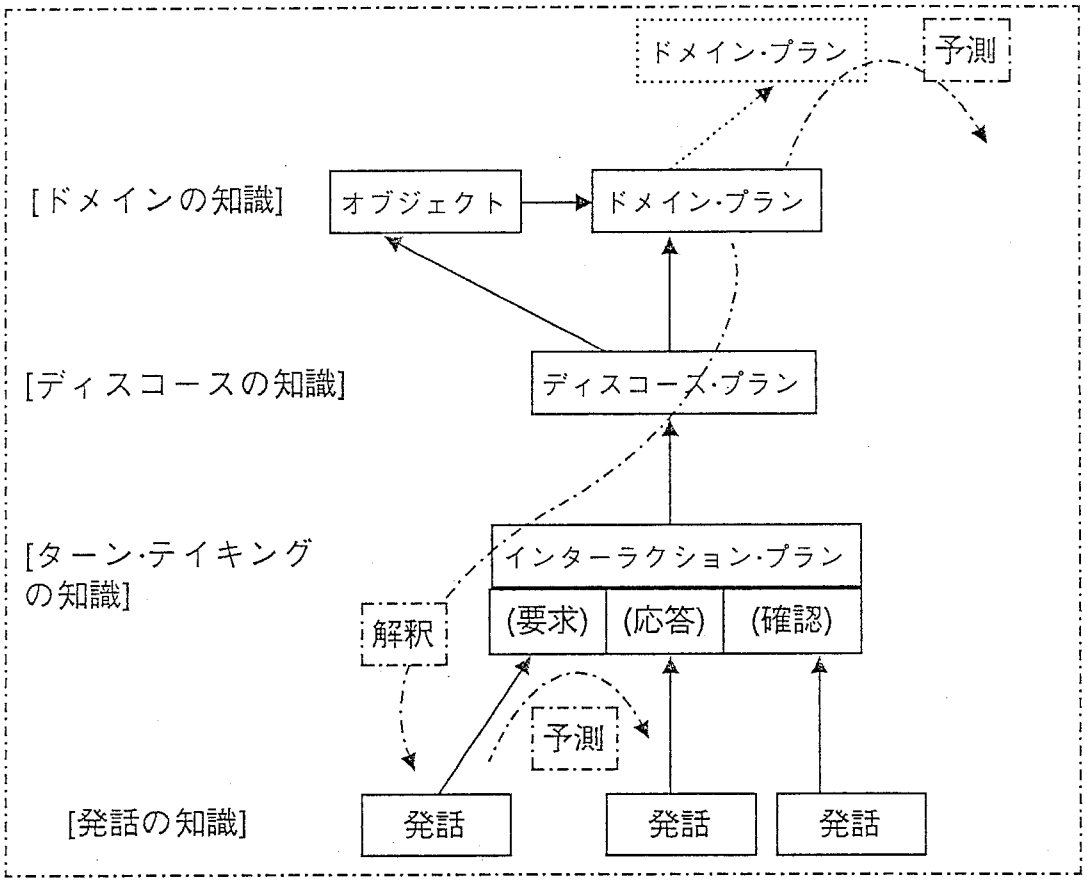


図3　対話構造の概念図

対話構造をもとにして発話の解釈や次発話の予測を行なうことができる。

(注) †　[Grosz 85] は対話の構造を次の3つの要素の複合体としてとらえている。
(1) the structure of the actual sequence of utterances inthe discourse.
(2) a structure of intentions
(3) an attentional state

3.1 発話の表現

ATRでは新しい機械翻訳の方式として発話行為のタイプを伝える"Intention Translation Method(ITM)"という方式を提案している[Iida87 Iida88]。この方式は入力発話を発話行為タイプと命題内容に分離し、原言語における発話行為を目標言語で的確に表現するための方式の一つである。発話行為は言語に依存した戦略も含むため、抽象化した発話行為タイプによる表現を介した翻訳が必要になる。この方式により滑らかな対話の通訳が実現できる。(このように発話を発話行為タイプと命題内容に分離するという考え方はコンサルテーション・システムの対話コントローラでも使われている[Sagawa 88]。)

本モデルでは入力発話が次の形で解析されていると仮定している。

(発話タイプ、発話者、聞き手、クルー・ワード、トピック、命題内容)†

発話タイプ： 入力発話の発話行為のタイプ
発話者： 入力発話の発話者
聞き手： 入力発話の聞き手
クルー・ワード： 話題の転換を明示的に表わす単語/句。
次の3つのタイプを考えるが、いずれかに確定するとは限らない。
forward : 違う話題に移ることを表わす。
(例)「ところで」など
backward : 以前の話題にもどることを表わす。
(例)「話を戻すと」など
continue : 次の話題に移ることを表わす。
(例)「それから」など
トピック： 入力発話のトピック。
(例) [は-格]や「~については」などの[は-格相当格]など
命題内容： 入力発話の命題内容で述語とその格要素で表わされる。

(例)「それでは、登録用紙を送って下さい」の解析結果は次のようになる。

(Request-Action† Speaker Hearer continue 登録用紙
(predicate&cases (predicate 送る)(agent Hearer)
(recipient Speaker)(object 登録用紙))

(注)† プログラムでは各表現のデリミタとして空白を使用しているが、印刷の関係で空白が見にくい場合は、コンマ(、,)をデリミタとしている。
従って(Request-Action, Speaker, ...) は (Request-Action Speaker ...) と同じことを表現している。また発話タイプに注目していることを表わすために Request-Action(Speaker ...)と表現する場合もある。

3.2 インターラクション･プラン

3.2.1 発話タイプの分類

目標指向型対話のCommunicative Actとして以下の発話タイプを設定する†。
(パーサの解析結果と発話タイプの対応については付録参照。)
発話タイプは次のように分類できる。

① オブジェクトに関するもの
- ㋐ ObjectのPropertyの値に関するアクト
  Ask-Value, Confirm-Value
- ㋑ ObjectのPropertyの存在に関するアクト
  Ask-Exist-Value

② アクションに関するもの
- ㋐ Actionの内容に関するアクト
  Ask-Action, Confirm-Action
- ㋑ Actionの実行に関するアクト
  Request-Action, Will-Do-Action, Will-Do-Action2,
  Ask-Acceptability1, Ask-Acceptability2

③ アクションの条件(PREREQUISITE)、効果(EFFECT)に関するもの
- ㋐ 命題内容に関するアクト
  Ask-Truth, Ask-Ref
- ㋑ 情報提供に関するアクト
  Inform, Inform-Value

④ 対話の制御に関するもの
- ㋐ 話題導入(予告)に関するアクト
  Introduce-Object, Notice-Value
- ㋑ Turn-takingの起動に関するアクト
  Require-Question, Notice-Question

(注)† [Cohen 84]は「聞き手の前にあるポンプの組立ての指示」というタスクでの会話の分析について、次のCommunicative Actを定めている。

Communicative Act	*Example*
Request(Assembly Action)	*put that on the hole*
Request(Orientation Action)	*the other way around, the top is the bottom*
Request(Pick-up)	*take the blue base*
Request(Identify-Referent)	*there is a little yellow piece of rubber*
Request(Informif(Identified-referent))	*got it ?, the litttle red plug ?*
Request(Achieve([relation]))	*and the purpose of that is to cover up that hole..,* [relation] = (Cober V2 Hole(TB))]
Label	*that's a plunger*

[Cohen 84]の対象とした対話は、物理的な物体に関する対話であり、対話参加者はエキスパートと初心者であり対等でない。即ち初心者の発話は大部分が"Yeah","Okay","Mm-hm"である。一方、T-Plan Modelの対象とした対話は、情報の問合せの対話であり、対話参加者は対等である。この相違により違ったCommunicative Actを設定している。

㈦ 対話のチャンネルに関するアクト

Greeting-Open, Greeting-Close

㈧ その他のアクト

Require-Question2

各発話タイプの記述を以下に示す。

－ ObjectのPropertyの値に関するアクト

HEADER : **Ask-Value**(speaker hearer value)
PREREQUISITE : BELIEVE(speaker KNOW(hearer value))
BELIEVE(speaker NOT(KNOW(speaker value)))
EFFECTS : BELIEVE(hearer WANT(speaker KNOW(speaker value)))
CONSTRAINTS :
(例)「AはWHですか」

HEADER : **Confirm-Value**(speaker hearer value)
PREREQUISITE : BELIEVE(speaker KNOW(hearer value))
BELIEVE(speaker KNOW(speaker value))
EFFECTS : BELIEVE(hearer WANT(speaker KNOWIF(speaker value)))
CONSTRAINTS :
(例)「AはBですね」「AはBですか」

－ ObjectのPropertyの存在に関するアクト

HEADER : **Ask-Exist-Value**(speaker hearer value)
PREREQUISITE : BELIEVE(speaker KNOWIF(hearer EXIST(value)))
EFFECTS : BELIEVE(hearer WANT(speaker
KNOWIF(speaker EXIST(value))))
CONSTRAINTS :
(例)「～はありますか。」「～は要るのですか」

－ Actionの内容に関するアクト

HEADER : **Ask-Action**(speaker hearer action)
PREREQUISITE : NOT(KNOW(speaker HOW-TO(action)))
EFFECTS : BELIEVE(hearer WANT(speaker
KNOW(speaker HOW-TO(action))))
CONSTRAINTS :
(例)「～するにはどうすればよいでしょうか」

HEADER : **Confirm-Action**(speaker hearer action)
PREREQUISITE : (KNOW(speaker HOW-TO(action))
EFFECTS : BELIEVE(hearer WANT(speaker
KNOWIF(speaker HOW-TO(action))))
CONSTRAINTS : speaker = AGENT(action)
(例)「～すればよいのですね」

— Actionの実行に関するアクト

HEADER : **Request-Action**(speaker hearer action)
PREREQUISITE : WANT(speaker action)
EFFECTS : BELIEVE(hearer WANT(speaker action))
CONSTRAINTS : hearer = AGENT(action)
(例)「～して下さい」

HEADER : **Will-Do-Action**(speaker hearer action)
PREREQUISITE : WANT(speaker KNOWIF(speaker WANT(hearer action)))
EFFECTS : BELIEVE(hearer WANT(speaker KNOWIF(speaker WANT(hearer action))))
CONSTRAINTS : speaker = AGENT(action)
(例)「～しましょうか」

HEADER : **Will-Do-Action2**(speaker hearer action)
PREREQUISITE : BELIEVE(speker WANT(hearer action))
EFFECTS : BELIEVE(hearer WILL-DO(speaker action))
CONSTRAINTS : speaker = AGENT(action)
(例)「～します」「～させていただきます」

HEADER : **Ask-Acceptability1**(speaker hearer action)
PREREQUISITE : WANT(speaker KONWIF(speaker (CAN-DO action)))
EFFECTS : BELIEVE(hearer WANT(speaker KONWIF(speaker (CAN-DO action))))
CONSTRAINTS : speaker = AGENT(action)
(例)「～できますか」

HEADER : **Ask-Acceptability2**
PREREQUISITE : WANT(speaker KONWIF(speaker (CAN-DO action)
EFFECTS : BELIEVE(hearer WANT(speaker KONWIF(speaker (CAN-DO action))))
CONSTRAINTS : hearer = AGENT(action)
(例)「～できますか」「～してもらえますか」

— 命題内容に関するアクト

HEADER : **Ask-Truth**(speaker hearer proposition)
PREREQUISITE : WANT(speaker KNOWIF(proposition))
EFFECTS : BELIEVE(hearer (WANT speaker KNOWIF(proposition)))
CONSTRAINTS :
(例)「(proposition)か」

HEADER : **Ask-Ref**(speaker hearer value)

PREREQUISITE : WANT(speaker KNOWIF(speaker KNOWREF(hearer value)))
EFFECTS : BELIEVE(hearer WANT(speaker KNOWIF(speaker KNOWREF(hearer value))))
CONSTRAINTS :
(例)「(wh)かわかりますか」ex. 会議は何時に始まるかわかりますか。

— 情報提供に関するアクト

HEADER : **Inform**(speaker hearer proposition)
PREREQUISITE : KNOW(speaker proposition)
BELIEVE(speaker WANT(hearer KNOW(hearer proposition)))
EFFECTS : KNOW(hearer proposition)
CONSTRAINTS :
(例)「(proposition)」

HEADER : **Inform-Value**(speaker hearer value)
PREREQUISITE : KNOW(speaker value)
BELIEVE(speaker WANT(hearer KNOW(hearer value)))
EFFECTS : KNOW(hearer value)
CONSTRAINTS :
(例)「AはBです」

— 話題導入(予告)に関するアクト

HEADER : **Introduce-Object**(speaker hearer object)
PREREQUISITE :
EFFECTS : WANT(hearer REPLY(Introduce-Object))
CONSTRAINTS : WANT(speaker INFORM(hearer speaker D(x))
(例)「~についてお伺いしたいのですが。」

HEADER : **Notice-Value**(speaker hearer value)
PREREQUISITE : WANT(speaker (INFORM-VALUE speaker hearer value))
EFFECTS : BELIEVE(hearer WANT(speaker (INFORM-VALUE speaker hearer value)))
CONSTRAINTS :
(例)「(先に)~を言います。」

— Turn-takingの起動に関するアクト

HEADER : **Require-Question**(speaker hearer)
PREREQUISITE : WANT(speaker INFORM(hearer speaker WANT(hearer INFORM(speaker herer D(x))))
EFFECTS : WANT(hearer REPLY(Require-Question))
CONSTRAINTS :

(例)「他に何かありますか」「どのようなご用件でしょうか」

HEADER : **Notice-Question**(speaker hearer)
PREREQUISITE :
EFFECTS : WANT(hearer REPLY(Notice-Question))
CONSTRAINTS : WANT(speaker INFORM(hearer speaker D(x)))
(例)「お尋ねしたいことがあるのですが。」

— 対話のチャンネルに関するアクト

HEADER : **Greeting-Close**(speaker hearer)
PREREQUISITE :
EFFECTS : WANT(hearer (OPEN-DIALOGUE))
CONSTRAINTS : WANT(speaker (OPEN-DIALOGUE))
(例)「失礼します」

HEADER : **Greeting-Open**(speaker hearer)
PREREQUISITE :
EFFECTS : WANT(hearer (CLOSE-DIALOGUE))
CONSTRAINTS : WANT(speaker (CLOSE-DIALOGUE))
(例)「もしもし」

— その他のアクト

HEADER : **Require-Question2**(speaker hearer)
PREREQUISITE :
EFFECTS : WANT(hearer REPLY(Require-Question2))
CONSTRAINTS :
(例)「分からない点がありましたら、いつでもお聞き下さい」

これらは主としてDemand Classに属する発話タイプであるが、これらと対話対を構成するResponse Classに属する発話タイプがある。それらには、命題内容を含むもの(「～して下さい(Direction)」、「Bです(Inform-Value)」など)と命題内容を含まないもの(「はい、そうです(Affirmative)」、「はい、お願いします(Accept-Offer)」)がある。これについては次節で述べる。また発話タイプと表層表現の対応についても次節で述べる。

3.2.2 発話タイプをもとにした対話対

[対話の内部構造]

対話の内部構造として(1)挨拶ー挨拶、(2)質問ー応答、(3)提示ー承認、(4)報告ー確認、(5)命令ー実行に類する隣接対があり、現実の対話では隣接対が相互に上位、下位の内部構造を形成する埋め込み構造を成すことが言語学者によって明らかにされている[Yamanashi 84]。また挨拶、質問、提示、報告、命令を相手に何らかの反応を求める「要求」としてとらえ、挨拶、応答、承認、確認、実行をそれぞれの要求に対する「応答」としてとらえ、さらに「わかりました」「そうですか」「なるほど」などを、その応答に対する「確認」としてとらえると、対話の基本構造(下位レベルの対話の構造)として(1)要求ー応答、(2)要求ー応答ー確認の2つの構造があることが対話の分析から明らかになっている[Arita and Iida 87]。

[対話対の分類]

質問ー応答の対話対は次の4つに分類できる。

① 質問に対してその答えをするもの

(例) 質問:費用はいくらですか。
応答:○○円です。

② 質問にそのまま答えるのではなく推論をして答えるもの

(例) 質問:費用はいるのですか。 (存在)→(Value)
応答:はい、○○円です。

③ 質問の前提が間違っているもの [Pollack 86 Kaplan 83]

(例) 質問:パーティには何人くらい参加しますか。
応答:パーティは開かれません。

④ 答えを知らないもの

(例) 質問:~について教えてください。
応答:そのようなことは、こちらではわかりません。

本報告では①②を対象とし、③④は対象としない。

[発話タイプをもとにした対話対]

このように対話があるレベルで一定のパタンをなす面を自動翻訳電話に生かすため、T-Plan Modelでは上記の[Yamanashi 84]の分類よりも詳細な発話タイプをもとに対話対を設定している。T-Plan Modelでの対話対を表1に示す。これらの対話対はインターラクション・プランとして実現される。

[表層表現と発話タイプの対応]

表層表現と発話の機能は1対1には対応しない。[Yamanashi 84]では次の例をあげている。

(a) A:「テーブルの上にケーキがありますが。」
B:「ええ。」

(b) A:「テーブルの上にケーキがありますが。」
B:「よろしいんですか?」

(c) A:「テーブルの上にケーキがありますが。」

B:「私がとってきましょう。」

Aの発話は(a)の場合には陳述[statement]であり、(b)の場合には提供[offer]であり、(c)の場合には依頼、要求[request]である。

T-Plan Modelでも表層表現と発話のタイプは1対1には対応しない。「～して下さい」は発話タイプRequest-ActionとDirectionに対応するが、前の発話が「～するにはどうすれば良いですか」といったAsk-Actionの発話の場合は、対話対の関係から「～して下さい」はDirectionと解釈する。また先行文脈から発話タイプを決定できない場合は、複数の解釈を可能性として残しておき、後続の発話から決定する。

[Kume 88]は次の例で、一発話の解析だけでは正しい翻訳が得られないことを指摘している。

Q:登録用紙をお送りしましょうか。
A-1:はい、有り難うございます。
A-2:はい、有り難うございます。よろしくお願いします。
A-3:はい、有り難うございます。今は結構です。
(Diaogue 5と同じ)

質問Qに対して、応答A-1、A-2の「有り難うございます」は「送って下さい/Thank you.」の意味であるのに対して、A-3は「送らなくてもいい/No, thank you.」の意味である。

T-Plan Modelでは、次のインターラクション･プランで処理する。(インターラクション･プランについては、次節を参照。) インターラクション･プランは2人の話者に共通に持っていて、両話者からアクセスする。

HEADER : Will-Do-Action-Unit(sp1 sp2)†
DECOMPOSITION1 : Will-Do-Action(sp1 sp2)
Accept-Offer(sp2 sp1)
DECOMPOSITION2 : Will-Do-Action(sp1 sp2)
Reject-Offer(sp2 sp1)
DECOMPOSITION3 : Will-Do-Action(sp1 sp2)
Accept-Offer2(sp2 sp1)
Accept-Offer(sp2 sp1)
DECOMPOSITION4 : Will-Do-Action(sp1 sp2)
Accept-Offer2(sp2 sp1)
Reject-Offer(sp2 sp1)

(注)†: sp1, sp2は2人の対話参加者を表わす。その他のパラメータは省略した。

Qはアクションの実行に関する発話タイプWill-Do-Actionと解析される。これはDECOMPOSITION1～4にユニファイする。システムは4つの可能性を保持する。「はい、有り難うございます。」はAccept-Offer又はAccept-Offer2に解析される。この発話はDECOMPOSITION1,3,4とユニファイし、2とはユニファイしないので、この時点で、システムはDECOMPOSITION1,3,4を可能性として保持する。次に「よろしくお願いします。(Accept-Offer)」の発話が来た場合は、DECOMPOSITION3の解釈だけを保持し、「今は結構です。(Reject-Offer)」の発話が来た場合は、DECOMPOSITION4の解釈だけを保持し、それ以外の発話が来た

場合は、DECOMPOSITION1の解釈だけを保持する。「有り難うございます」の訳語の選択に関してはDECOMPOSITION1と3には"Thank you."を、4には"No, thank you."を対応づけておくと変換･生成の訳し分け情報として利用できる。

[Yamanashi 84],[Kume 88]の例は1つの表層が複数の発話タイプと対応するという例であるが、ある文脈ではその複数の発話タイプの1つに決まる例である。しかし、ある文脈で複数の発話タイプを同時に持つ場合がある。

A：何かほかにご質問はございますか
B：会場とホテルは近いですか

Aの要求の発話に対してBの発話は「質問がある」という応答の発話であると同時に新たな要求の発話になっている。

また複数の発話が1つの発話タイプに相当する場合がある。

(1)B: お支払いはどうされますか
(2)A: クレジット･カードは使えますか
(3)B: どこのカードをお持ちですか
(4)A: VISAです
(5)B: VISAでお支払いいただけます
(6)A: ところで.....

(1)の要求の発話に対して(2)~(5)の発話全体で「クレジット･カード(VISA)で支払います」という発話に相当している。

これらのT-Plan Modelでの扱いについては検討中である。

表1　発話タイプをもとにした対話対

Demand Class	Response Class
Ask-Value (例)「AはWHですか」	Inform-Value (例)「Bです」
Confirm-Value (例)　「AはBですね」 「AはBですか」	Affirmative (例)「はい、そうです」 Negative (例)「いいえ、違います」
Ask-Exist-Value (例)　「～はありますか」 「～は要るのですか」	Affirmative-Exist (例)　「はい、あります」 「はい、要ります」 Negative-Exist (例)　「いいえ、ありません」 「いいえ、要りません」
Ask-Action (例)「～するにはどうすればよいですか」	Direction (例)「～してください」
Request-Action (例)「～して下さい」	Accept (例)「はい、わかりました」 Reject (例)「申し訳ありませんが､～できません」 Will-Do-Action2 (例)「～します」
Confirm-Action (例)「～すればよいのですね」	Affirmative (例)「はい、そうです」 Negative (例)「いいえ、違います」
Will-Do-Action (例)「～しましょうか」	Accept-Offer (例)　「はい、お願いします」 「はい、有り難うございます」 Reject-Offer (例)「いいえ、結構です」
Will-Do-Action2 (例)「～します」	Accept-Offer (例)「はい、お願いします」
Ask-Acceptability1 (例)「～できますか?」 Ask-Acceptability2 (例)「～できますか?」	Accept2 (例)　「はい、結構です」 「はい、できます」 Reject (例)「申し訳ありませんが､～できません」
Require-Question (例)　「他に何かありますか?」 「はい、何でしょうか?」	

Require-Question2 (例)「わからない点がありましたら、いつでもお聞き下さい」	Accept-Offer2 (例)「はい、有り難うございます」
Ask-Truth (例)「(proposition)か」	Affirmative-Truth (例)「はい、(proposition)」 「はい、そうです」 「はい、そのとおりです」 Negative-Truth (例)「いいえ、not(proposition)」
Ask-Ref (例)「(wh)か」	Affirmative-ref (例)「はい、(proposition)」 Negative-ref (例)「いいえ、not(proposition)」
Greeting-Close (例)「失礼します」 「色々どうもありがとうございました」	Greeting-Close (例)「失礼します」 Greeting-Response (例)「こちらこそ」 「どういたしまして」
Inform-Value (例)「AはBです」 Inform (例)「(proposition)」	Confirm (例)「そうですか」 Accept (例)「はい、わかりました」
Introduce-Object (例)「~についてお伺いしたいのですが」 Notice-Question (例)「お尋ねしたいことがあります」	Confirm2 (例)「はい、どうぞ」 「はい、何でしょうか」

3.2.3　インターラクション･プラン

インターラクション･プランは対話のターン･テイキングの構造を表わす。インターラクション･プランはドメインに依存しない。インターラクション･プランは対話対と推論規則 (3.5節) から作られる。インターラクション･プランの例を以下に示す。

① アクションの実行に関するプラン(REQUEST-ACTION-UNIT)

REQUEST-ACTION-UNIT		
decomposition1 :	request-action-a	ーしてください
	accept	わかりました
decomposition2 :	request-action-r	ーしてください
	reject	申し訳ありませんが、ーできません

```
((header (request-action-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition    (request-action-a ?sp ?hr ?clue ?topic ?action)
                   (accept ?hr ?sp ?clue ?topic
                                  (pred&cases (predicate wakaru)(agent ?hr)(recipient ?recip)
                                                 (object ?action)(manner ?manner))))
 (belong-to interaction-plan))
```

```
((header (request-action-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition    (request-action-r ?sp ?hr ?clue ?topic ?action)
                   (reject ?hr ?sp ?clue ?topic ?action2))
 (belong-to interaction-plan))
```

【説明】

▸ decomposition :

あるアクション (?action)の実行に関する発話の流れには、(1)話者(?sp)がアクション (?action)を要求して(request-action-a)、話者(?hr)が承諾する(accept)場合(decomposition1)、または(2)話者(?sp)がアクション (?action)を要求して(request-action-n)、話者(?hr)が拒否する(reject)場合(decomposition2)がある。

② オブジェクトのプロパティの値を知るプラン(GET-VALUE-UNIT)

GET-VALUE-UNIT		
decomposition1 :	ask-value	?object は wh ですか
	inform-value	?id です
	confirm	分かりました
decomposition2	inform-value	?id です

```
((header (get-value-unit ?sp ?hr ?clue ?topic
```

```
                    (pred&cases (predicate desu)(object ?property[?object])(identifier ?id))))
(decomposition   (ask-value ?sp ?hr ?clue ?topic
                    (pred&cases (predicate desu)(object ?property[?object])(identifier ?id)))
                  (inform-value ?hr ?sp ?clue ?topic
                    (pred&cases (predicate desu)(object ?property[?object])(identifier ?id)))
                  (confirm ?sp ?hr ?clue ?topic ?action))
(add-effect (know ?sp ?property[?object] ?id))
(belong-to interaction-plan))
```

```
((header (get-value-unit ?sp ?hr ?clue ?topic
                    (pred&cases (predicate desu)(object ?property[?object])(identifier ?id))))
(decomposition (inform-value ?hr ?sp ?clue ?topic
                    (pred&cases (predicate desu)(object ?property[?object])(identifier ?id))))
(add-effect (know ?sp ?property[?object] ?id))
(belong-to interaction-plan))
```

【説明】

▸ decomposition :

質問者(?sp)がオブジェクト(?object)のプロパティ(?property)の値(?id)を知るためには、(1)質問者(?sp)が質問して(ask-value)、事務局(?hr)が答えて(inform-value)、その後 質問者(?sp)が確認する(confirm)という発話の流れ(decomposition1)、または(2)質問者(?sp)が質問しなくても、事務局(?hr)がオブジェクト(?object)のプロパティ(?property)の値(?id)を言う(inform-value)という発話の流れ(decomposition2)で実現できる。

▸ (add-effect (know ?sp ?property[?object] ?id))

このプランが成功するとその効果として、質問者(?sp)がオブジェクト(?object)のプロパティ(?property)の値(?id)を知る。(know ?sp ?property[?object] ?id) が現在の共通理解状態のリストに加えられる。

3.3 ディスコース・プラン

ディスコース・プランはドメイン・プランとインターラクション・プランを結び付けるものである。ディスコース・プランはドメインに依存しない。ディスコース・プランの主なものを以下に示す。

① アクションの実行に関するインターラクション・プランとドメイン・プランを結ぶプラン(INTRODUCE-DOMAIN-PLAN-NODE)

INTRODUCE-DOMAIN-PLAN-NODE
  decomposition1 : REQUEST-ACTION-UNIT
  decomposition2 : WILL-DO-ACTION-UNIT

```
((header (introduce-domain-plan-node ?action))
(decomposition (request-action-unit ?sp ?hr ?clue ?topic ?action))
(belong-to discourse-plan))
```

((header (introduce-domain-plan-node ?action))
(decomposition (will-do-action-unit ?sp ?hr ?clue ?topic ?action))
(belong-to discourse-plan))

【説明】
▸「～して下さい」「わかりました」といった対話対を表わすインターラクション･プランREQUEST-ACTION-UNITや、「～します」「お願いします」といった対話対を表わすインターラクション･プランWILL-DO-ACTION-UNITはドメイン･プランのアクションの実行に直接関連する発話の流れである。

② オブジェクトのプロパティの値に関するインターラクション･プランとドメイン･プランを結ぶプラン（INTRODUCE-OBJECT-PLAN）

INTRODUCE-OBJECT-PLAN
decomposition1 : GET-VALUE-UNIT
decomposition2 : CONFIRM-VALUE-UNIT
decomposition3 : CONFIRM-VALUE2-UNIT
decomposition4 : INTRODUCE-OBJET

((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
(decomposition (get-value-unit ?sp ?hr ?clue ?topic ?action))
(goal-effect (focus-plan ?topic))
(belong-to discourse-plan))

((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
(decomposition (confirm-value-unit ?sp ?hr ?clue ?topic ?action))
(goal-effect (focus-plan ?topic))
(belong-to discourse-plan))

((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
(decomposition (confirm-value2-unit ?sp ?hr ?clue ?topic ?action))
(goal-effect (focus-plan ?topic))
(belong-to discourse-plan))

((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
(decomposition (introduce-object ?sp ?hr ?clue ?topic ?action))
(goal-effect (focus-plan ?topic))
(belong-to discourse-plan))

【説明】
▸(goal-effect (focus-plan ?topic)) :
focus-planは?topicから連想されるドメイン･プランを連想テーブル(topic→domain plan)から求め、それがゴール･リストにある場合は、そのドメイン･プラ

ンをゴール・リストの先頭に移動する。ゴール・リストにない場合は、そのドメイン・プランを新たにゴール・リストの先頭に加える手続である。

3.4 ドメイン・プラン

ドメイン・プランは対話の内容のドメインのアクションの階層関係を表わす。(「国際会議の問合せ」におけるドメイン・プランの例を付録A1に示す。)
ドメイン・プランの例を次に示す。
①MAKE-REGISTRATION

```
((header (MAKE-REGISTRATION ?sp ?hr))
 (decomposition          (get-the-form ?sp ?hr ?form)
                         (fill-the-form ?sp ?hr ?form)
                         (return-the-form ?sp ?hr ?form))
 (pred-structure (pred&cases          (predicate moushikomu)(agent ?sp)(object kaigi)))
(belong-to domain-plan))
```

【説明】
▸ (decomposition(get-the-form ?sp ?hr ?form)(fill-the-form ?sp ?hr ?form) (return-the-form ?sp ?hr ?form)) :
MAKE-REGISTRATIONは質問者(?sp)が会議(kaigi)に登録するためには、まず登録用紙(?form)を手に入れて(get-the-form)、登録用紙(?form)に記入して(fill-the-form)、登録用紙(?form)を返送すればよい(return-the-form)ということを表わすプランである。ここで、送付・記入・返送する登録用紙は同じ変数 ?form で記述されていて同一の登録用紙であることを示している。

②GET-THE-FORM

```
((header (GET-THE-FORM ?sp ?hr ?form))
 (prerequisite (know ?hr name&address[?sp] ?n&a))
 (decomposition (introduce-domain-plan-node
                         (pred&cases (predicate okuru)(agent ?hr)(recipient ?sp)
                                     (object?form)(manner ?manner))))
 (add-effect (pred&cases (predicate motsu)(agent ?sp)(object ?form)))
 (belong-to domain-plan))
```

【説明】
▸ (prerequisite (know ?hr name&address[?sp] ?n&a)) :
質問者(?sp)が登録用紙(?form)を手に入れるためには、事務局(?hr)が質問者(?sp)の住所・氏名(name&address[?sp])を知っていなければならない。
▸ (add-effect (pred&cases (predicate motsu)(agent ?sp)(object ?form))) :
GET-THE-FORMが成功すると質問者(?sp)が登録用紙(?form)を持つ(motsu)状態になる。

③FILL-THE-FORM

```
((header (FILL-THE-FORM ?sp ?hr ?form))
 (decomposition (introduce-domain-plan-node
                    (pred&cases (predicate kaku)(agent ?sp)(recipient ?recip) (object
                    ?form) (manner ?manner))))
 (belong-to domain-plan))
```

④RETURN-THE-FORM

```
((header (RETURN-THE-FORM ?sp ?hr ?form))
 (decomposition (introduce-domain-plan-node
                    (pred&cases (predicate okuru)(agent ?sp)(recipient ?hr) (object ?form)
                    (manner ?manner))))
 (add-effect (have ?hr ?form))
 (del-effect (have ?sp ?form))
 (belong-to domain-plan))
```

【説明】

▸ (add-effect (have ?hr ?form)) :
RETURN-THE-FORMが成功すると事務局 (?hr) が登録用紙 (?form) を持つ(motsu)状態になる。

▸ (del-effect (have ?sp ?form)) :
RETURN-THE-FORMが成功すると質問者 (?sp) が登録用紙 (?form) を持つ(motsu)状態であることがなくなる。

3.5 推論規則

推論規則は①応答に関する推論規則と②発話タイプに関する推論規則とプランに関する推論規則に分けることができる。現在のインプリメンテーションでは、応答に関する推論規則と発話タイプに関する推論規則はインターラクション・プランの形で表現され、プランに関する推論規則はプランナの制御機構として表現される。

①応答に関する推論規則

(r1) [間接発話意図に応答する]
→ [発話タイプに対する応答] + [発話意図に対する応答]

(r2) [提供の発話に応答する]
→ [相手が提供の意志を示したことに対する応答]
+ [提供に対する応答]

(r2)はインターラクション・プランで次のように実現される。

```
HEADER:          WILL-DO-ACTION-UNIT(?sp, ?hr , ?clue, ?topoc, ?action)
DECOMPOSITION1 : WILL-DO-ACTION(?sp, ?hr, ?clue , ?topic, ?action2)
                 ACCEPT-OFFER2(?hr, ?sp, nil, nil, ?action3)
                 ACCEPT-OFFER(?hr, ?sp, nil, ?topic, ?action4)
DECOMPOSITION2 : WILL-DO-ACTION(?sp, ?hr, ?clue , ?topic, ?action2)
                 ACCEPT-OFFER2(?hr, ?sp, nil, nil, ?action3)
                 Reject-OFFER(?hr, ?sp, nil, ?topic, ?action4)
```

提供の発話 Will-Do-Action に対しては次の対話対が設定されている。

Demand :	Will-Do-Action	「～しましょうか」
Response :	Accept-Offer	「(はい、)お願いします」
Demand :	Will-Do-Action	「～しましょうか」
Response :	Reject-Offer	「(いいえ、)結構です」

(r2)により、提供の発話に対して、その提供を受諾するか辞退するかを別にして、とりあえず相手が提供を申し出たことにたいする感謝の発話 ACCEPT-OFFER2 (「有り難う御座います」など) をしてから、相手の提供の申し出に対する受諾 Accept-Offer または 辞退 Reject-Offer の発話をする。

②発話タイプに関する推論規則

(s1) Ask-Exist-Value(sp, hr, clue, topic, ((predicate *ARU*)(object *Prop*(*OBJ*)))
→ Ask-Value(sp, hr, clue, topic, ((predicate *DESU*)(object *Prop*(*OBJ*))
(identifier *?Value*))

ObjectのPropertyの存在に関する質問の発話はそのPropertyのValueの質問の発話である可能性がある。

(例) 「参加費はいるのですか」 → 「(もし参加費が要る場合、)参加費はいくらですか」

(s2) Confirm-Value(sp, hr, clue, topic, ((predicate *DESU*)(object *Prop*(*OBJ*))
(identifier *Value*))
→ Ask-Value(sp, hr, clue, topic, ((predicate *DESU*)(object *Prop*(*OBJ*))
(identifier *?Value*))

ObjectのPropertyのValueの確認に関する発話はそのPropertyのValueの質問の発話である可能性がある。

(例)「締切りは12月末ですね」 → 「(もしそうでない場合、)締切りはいつですか」

(s3) Confirm-Action(sp, hr, clue, topic, predicate&cases)
→ Will-Do-Action2(sp, hr, clue, topic, predicate&cases)

Actionの確認の発話は話者がそのActionを実行するつもりであることを伝える発話である可能性がある。

(例)「申込み用紙をお送りすれば良いのですね」 → 「申込み用紙をお送りします」

(s4) Ask-Acceptability1(sp, hr, clue, topic, predicate&cases)
→ Ask-Action(sp, hr, clue, topic, predicate&cases)

Actionの可能性を尋ねる発話はそのActionの内容を尋ねる発話である可能性がある。

(例)「もし都合が悪くなった場合キャンセルということはできますか」 → 「キャンセルするにはどうすればよいのですか」

(s5) Ask-Acceptability2(sp, hr, clue, topic, predicate&cases)
→ Request-Action(sp, hr, clue, topic, predicate&cases)

Actionの可能性を尋ねる発話はそのActionを相手に依頼する発話である可能性がある。

(例)「申込み用紙を21部送って頂くということはできますか」 → 「申込み用紙を21部送って下さい」

(s6) Ask-Ref(sp, hr, clue, topic, predicate&cases)
→ Ask-Value(sp, hr, clue, topic, predicate&cases)

Valueを知っているかどうか尋ねる発話はそのValueを相手に尋ねる発話である可能性がある。

(例)「会議が何時に始まるか知っていますか」 → 「会議は何時に始まりますか」

(s1)~(s6)の発話タイプに関する推論規則は、①の(r1)の発話意図の推論に利用される。

たとえば、(s1)はインターラクション･プランで次のように実現される。

```
HEADER:          ASK-EXIST-VALUE-UNIT(?sp, ?hr , ?clue, ?topoc, ?action)
DECOMPOSITION :  Ask-Exist-Value(?sp, ?hr, ?clue , ?topic, ?action2)
                 Affirmative(?hr, ?sp, nil, nil, ?action3)
                 Inform-Value(?hr, ?sp, nil, ?topic, ?action4)
```

これは次の2つの対話対を(r1),(s1)で結び付けて合成したものである。

Demand : Ask-Exist-Value ----(s1)----> Demand : Ask-Value
Response : Affirmative Response : Inform-Value

(r1) Response → Affirmative + Inform-Value

③ドメイン・プランに関する推論規則

(d1) PREDICATE&CASES predicate&cases → HEADER action
ドメイン・プランのPREDICATE&CASES predicate&casesにユニファイする発話はHEADER actionをゴールとする発話である。

(d2) DECOMPOSITION action-1 → HEADER action
ドメイン・プランのDECOMPOSITION action-1ユニファイするアクションはHEADER actionをゴールとするアクションである。

(d3) EFFECTS predicate&cases → HEADER action
ドメイン・プランのEFFECTS predicate&casesにユニファイする発話はHEADER actionをゴールとする発話である。

(d4) PREREQUISITE predicate&cases → HEADER action
ドメイン・プランのPREREQUISITE predicate&casesにユニファイする発話はHEADER actionをゴールとする発話である。

(d5) HEADER action → EFFECTS predicate&cases
ドメイン・プランのHEADER actionが完了するとEFFECTS predicate&casesが成立する。

3.6 プランナ

3.6.1 構成

プランナの構成を図3に示す。プランナは初めにドメイン･プラン、ディスコース･プラン、インターラクション･プラン、連想テーブル(topic→domain plan)、連想テーブル(utterance-type→discourse plan)をファイルから読み込む。プランナはパーサから渡される発話の解析結果をゴールリスト、旧ゴールリスト、現在の理解状態を参照しながらプランに関する推論規則に従って、解釈を試み、発話の解釈を変換･生成部へ渡す。また必要に応じて発話内容のレベルの次発話の予測を行ない音声認識部へ渡す。

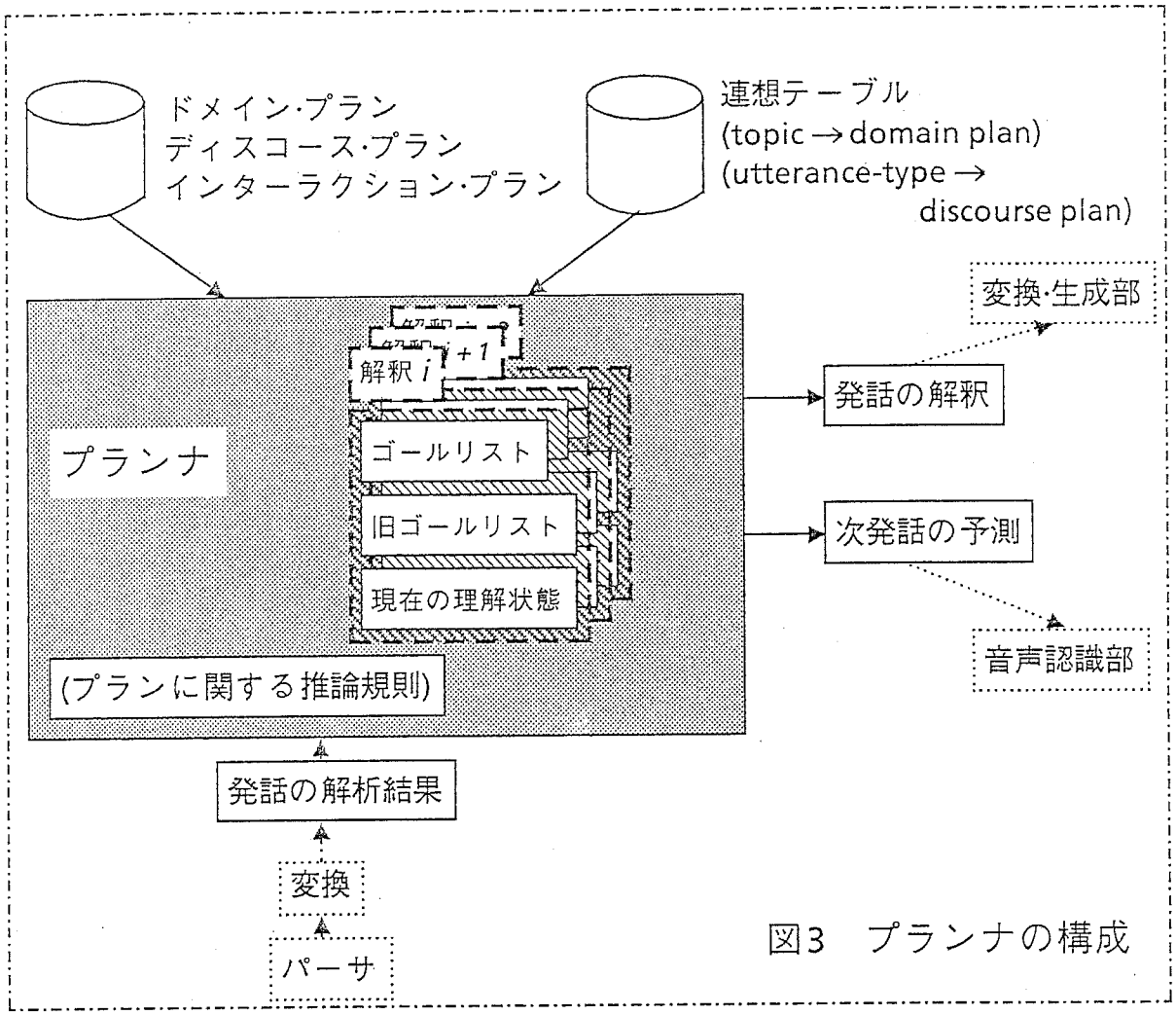


図3 プランナの構成

[解釈にあいまいさがる場合の処理]

対話の解釈にあいまいさがある場合は、そのすべての解釈の可能性を保持する。入力された発話が、各解釈 i のゴールリスト、旧ゴールリストのゴールに適合すれば、その解釈をひき続き保持し、適合しない解釈は破棄する。

3.6.2 基本動作

[プランナの基本的動作]

プランナの基本的動作は以下のとおりである。

step1： ゴールを設定してGOALとする。
　　step2へ
step2： GOALがnilの場合、処理は失敗して終了する。
　　nilでない場合、step3へ
step3： 入力からGOALまでのパスを求める。
　　パスが見つかった場合、処理は成功して終了する。
　　見つからない場合、step1へ

[ゴールの設定方法]
　step1のゴールの設定方法は以下のとおりである。
(step3の入力からGOALまでのパスの探索方法は次節で述べる。)

step1： ゴールリストにあるものを順次GOALとして入力からGOALまでのパスを求める。
　　パスが見つかった場合、処理は成功して終了する。
　　見つからない場合、step2へ
step2： 発話タイプが確認の発話タイプに属する場合、旧ゴールリストにあるものを順次GOALとして入力からGOALまでのパスを求める。
　　パスが見つかった場合、処理は成功して終了する。
　　見つからない場合、step3へ
step3： 発話タイプから連想されるゴールを連想テーブル(発話タイプ→ディスコース・プラン)から求めてそれをGOALとして入力からGOALまでのパスを求める。
　　パスが見つかった場合、処理は成功して終了する。
　　見つからない場合、step4へ
step4： トピックから連想されるゴールを連想テーブル(トピック→ドメイン・プラン)から求めてそれをGOALとして入力からGOALまでのパスを求める。
　　パスが見つかった場合、処理は成功して終了する。
　　見つからない場合、step5へ
step5： 処理は失敗して終了する。

[プランの探索範囲]
　プランの階層性を利用してプランの探索範囲を制限している。各プランのBELONG-TO スロットにそのプランの属するクラスが記述されている。ドメイン・プランは domain-plan、ディスコース・プランは discourse-plan、インターラクション・プランは interaction-plan である。また入力発話は utterance になっている。現在のパスの先頭が入力発話の場合は、それに続くパスのプランの候補はインターラクション・プランとディスコース・プランの中から探す。パスの先頭がインターラクション・プランの場合は、ディスコース・プランの中から探す。パスの先頭がディスコース・プランの場合は、ディスコース・プランとドメイン・プランの中から探す。パスの先頭がドメイン・プランの場合はドメイン・プランの中から探す。

　この関係は大域変数 *plan-class-hierarchy* に保持されている。

(例)　プランのクラス別の探索範囲の記述

```
(setq *plan-class-hierarchy*
    '((utterance      (interaction-plan discourse-plan))
      (interaction-plan (discourse-plan))
      (discourse-plan   (discourse-plan domain-plan))
      (domain-plan      (domain-plan))))
```

プランのパスは次の3つに分けることができる。

(A)インターラクション･プランからディスコース･プランへのパス
(B)ディスコース･プランからディスコース･プランへのパス
(C)ディスコース･プランからドメイン･プランへのパス

(A)と(B)のパスは発話のpredicate&casesに依存しないものが多いので予めパスの候補の可能性を求めておくことができる。しかし(C)のパスでpredicate&casesに依存してダイナミックに変わる場合は、予めパスの候補の可能性を求めておくことができない。ドメイン･プランは数が多いので全数探索を行うと時間がかかることも予想される。(現在のインプリメンテーションでは全数探索を行っている。)

ディスコース･プランからドメイン･プランへのパスは2つに分けられる。

(C-1)　ディスコース･プランからドメイン･プランへ直接パスが張られるもの。
例) アクションの実行に関する発話

(C-2)　ディスコース･プランからドメイン･プランへ連想テーブルを介してパスが張られるもの。
例) オブジェクトのプロパティの値に関する発話　など

(C-1)は上で述べたように全数探索を行うと時間がかかる。しかし(C-2)はテーブル内の少数のドメイン･プランの可能性だけを探索すれば良いので時間はかからない。

一般的に、話題の変化を示すクルー･ワードがない場合は、前の発話で参照されたドメイン･プランの近くのドメイン･プランへしか話題が移らない。そこで(C-1)を次の2つに分けて考える。

(C-1-1)　ディスコース･プランからドメイン･プランへ直接パスが張られるもののうち、話題の変化を示すクルー･ワードがない場合
(C-1-2)　ディスコース･プランからドメイン･プランへ直接パスが張られるもののうち、話題の変化を示すクルー･ワードがある場合

本システムが対象としているタスク･オリエンティッドな対話では、(C-1-1)の場合の関連するドメイン･プランの探索は、前の発話で参照されたドメイン･プランから初めて近くにあるドメイン･プランから順に調べるようにし、かつ関連するドメイン･プランがみつかった時点で探索を中止すれば探索時間はあまりかからない。(C-1-2)の場合の探索範囲については今後の検討課題である。

[プランの探索パス]

プランの探索は基本的にはDECOMPOSIOTIONのつながりで行なう。これは推論規則の(d2)を実現するものである。しかしパスの探索に失敗した場合、プランのINFERENCE-PATHスロットに指定があれば、その指定に従ってADD-EFFECT,

PREREQUISITE などのつながりで探索を行なう。これは推論規則の(d3)、(d4)を実現するものである。

(例) inference-path スロットの記述

```
;;;-------------------------------------------------------------
;;; ASK-TRUTH-PLAN
;;;   decomposition : ASK-TRUTH-UNIT
;;;-------------------------------------------------------------
((header (ask-truth-plan ?action))
 (decomposition (ask-truth-unit ?sp ?hr ?clue ?topic ?action))
 (inference-path (add-effect prerequisite))
 (belong-to discourse-plan))
```

leaf のドメイン・プランの DECOMPOSITION にはアクションの実行に関する記述がなされているので、アクションの実行に関する発話でないものは、ドメイン・プランとは直接 DECOMPOSITION chain では結び付けることができない。この場合は、その発話のトピックとドメイン・プランを結び付ける連想テーブルを介してディスコー・プランとドメイン・プランが結び付けられる。この連想テーブルの参照は goal-effect などによって起動される。

(例) 連想テーブル(トピック → ドメイン・プラン)

```
;                              TOPIC                DOMAIN-PLAN
(setq *topic-plan* '((shimekiri             return-the-form)
                     (kaigi__moushikomi     make-registration)
                     (form                  make-registration)
                     (sankaryou             pay-the-fee)
                     ))
```

(例) goal-effect スロットの記述

```
;;;-------------------------------------------------------------
;;; INTRODUCE-OBJECT-PLAN
;;;   decomposition1 : GET-VALUE-UNIT
;;;   decomposition2 : CONFIRM-VALUE-UNIT
;;;   decomposition3 : CONFIRM-VALUE2-UNIT
;;;   decomposition4 : INTRODUE-OBJET
;;;-------------------------------------------------------------
((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
 (decomposition (get-value-unit ?sp ?hr ?clue ?topic ?action))
 (goal-effect (focus-plan ?topic))
 (belong-to discourse-plan))
```

3.6.3　インターラクション･プランからドメイン･プランへの関係づけ

入力からGOALまでのパスの探索方法を以下に示す。

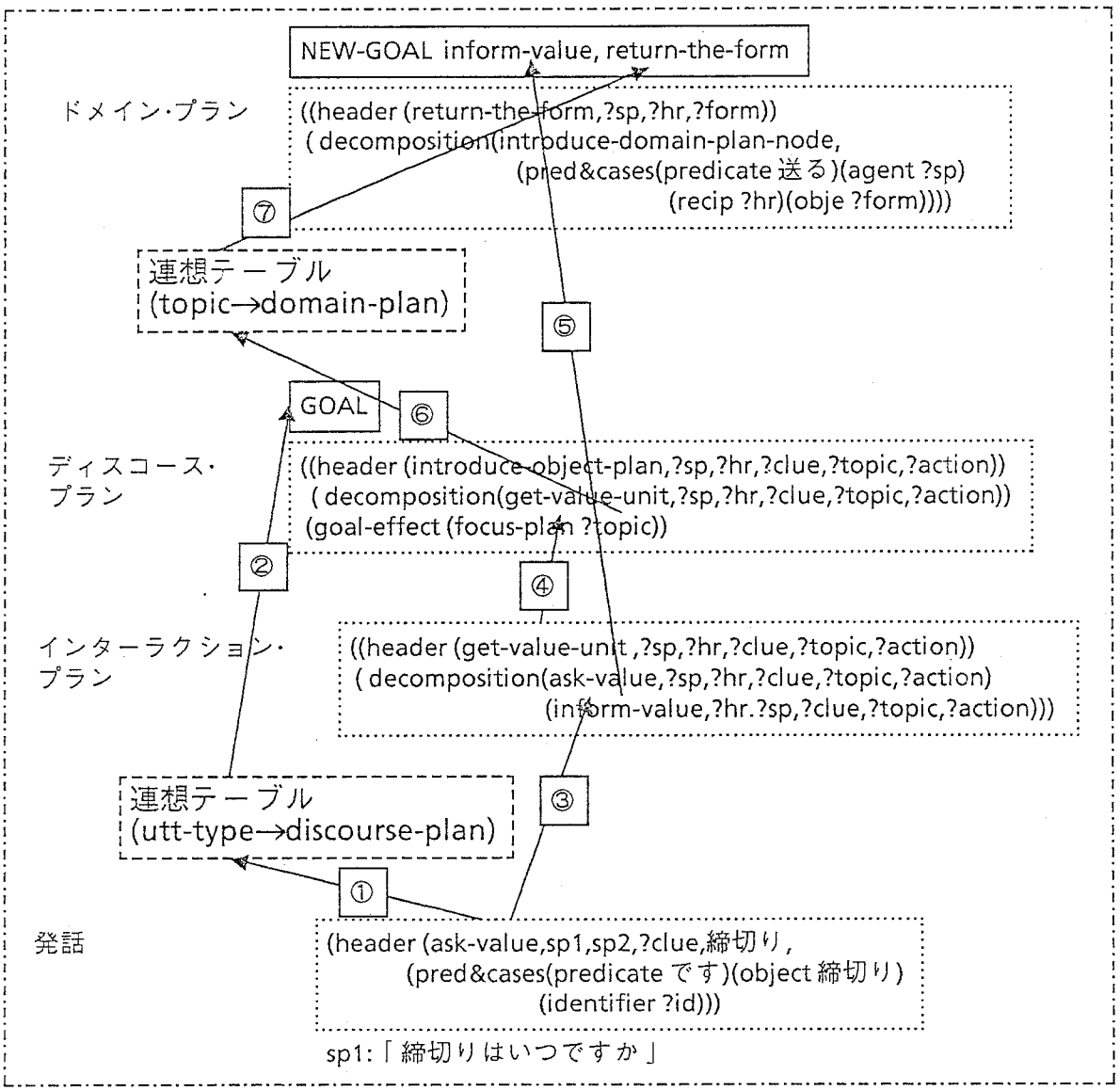
(1)ObjectのPropertyの値に関するプラン
→ObjectのProperty→連想テーブル→ドメイン･プランのPredicate&Cases
例)

①②　発話タイプがask-valueやconfirm-valueのようにObjectのPropertyの値に関するものの場合は、発話タイプに関連するディスコース･プランを連想テーブルから求めて、それをゴールとする。

③④　ゴールに向かってdecomposition chainを求める。

⑤　decompositionが完了していないものは、新しくゴールに登録する。

⑥⑦　ゴールのeffectを実行する。focus-planというeffectの場合は、そのトピックに関連するドメイン･プランを連想テーブルから求めて、それを新しくゴールとする。



(2)Objectの Propertyの存在に関するプラン
→ObjectのProperty→連想テーブル→ドメイン・プランのPredicate&Cases

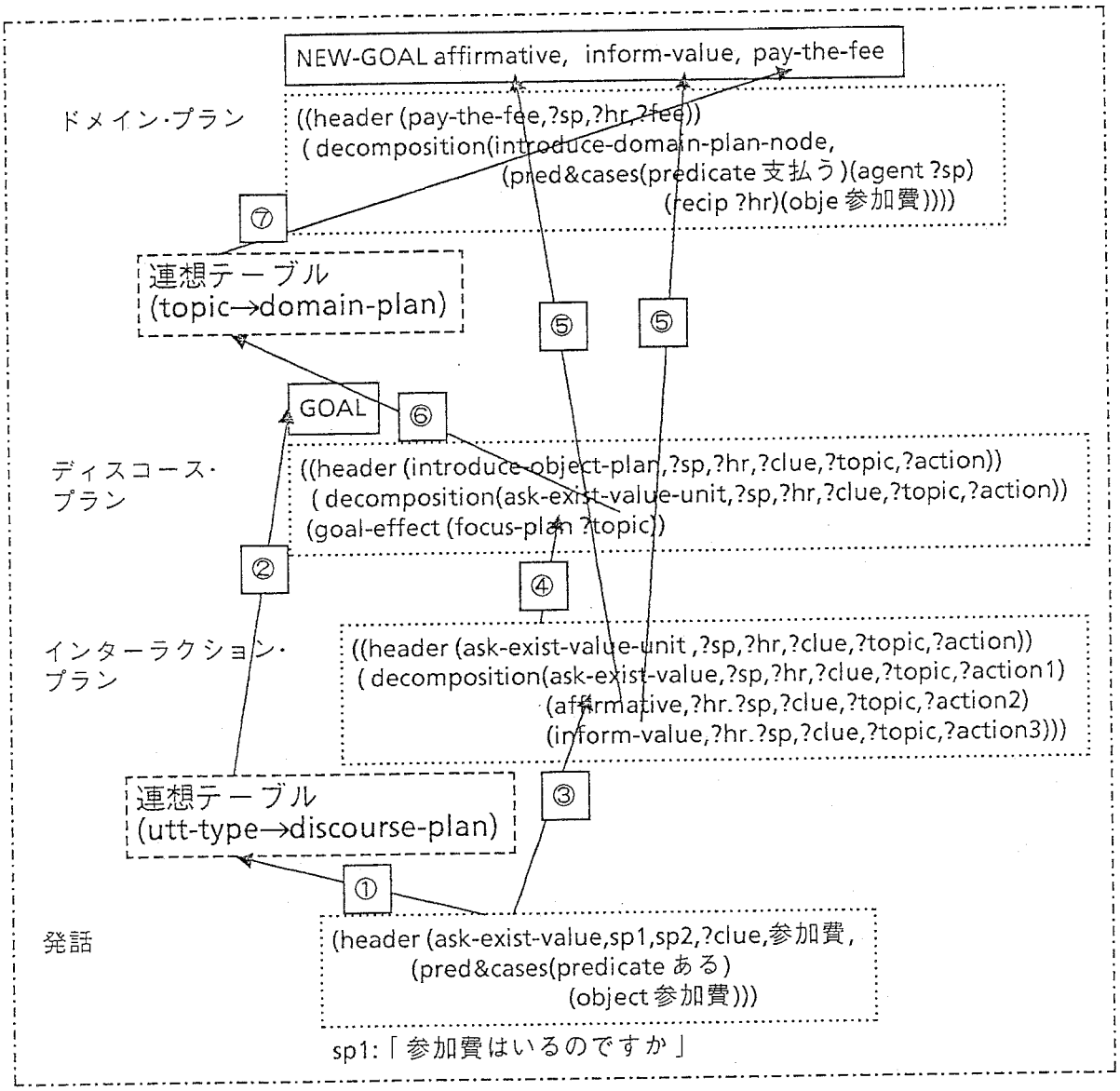
例)

①② 発話タイプが ask-exist-value のようにObjectのPropertyの値の存在に関するものの場合は、発話タイプに関連するディスコース・プランを連想テーブルから求めて、それをゴールとする。

③④ ゴールに向かってdecomposition chainを求める。

⑤ decomposition が完了していないものは、新しくゴールに登録する。

⑥⑦ ゴールの effect を実行する。focus-planというeffectの場合は、そのトピックに関連するドメイン・プランを連想テーブルから求めて、それを新しくゴールとする。



(3)Actionの内容に関するプラン

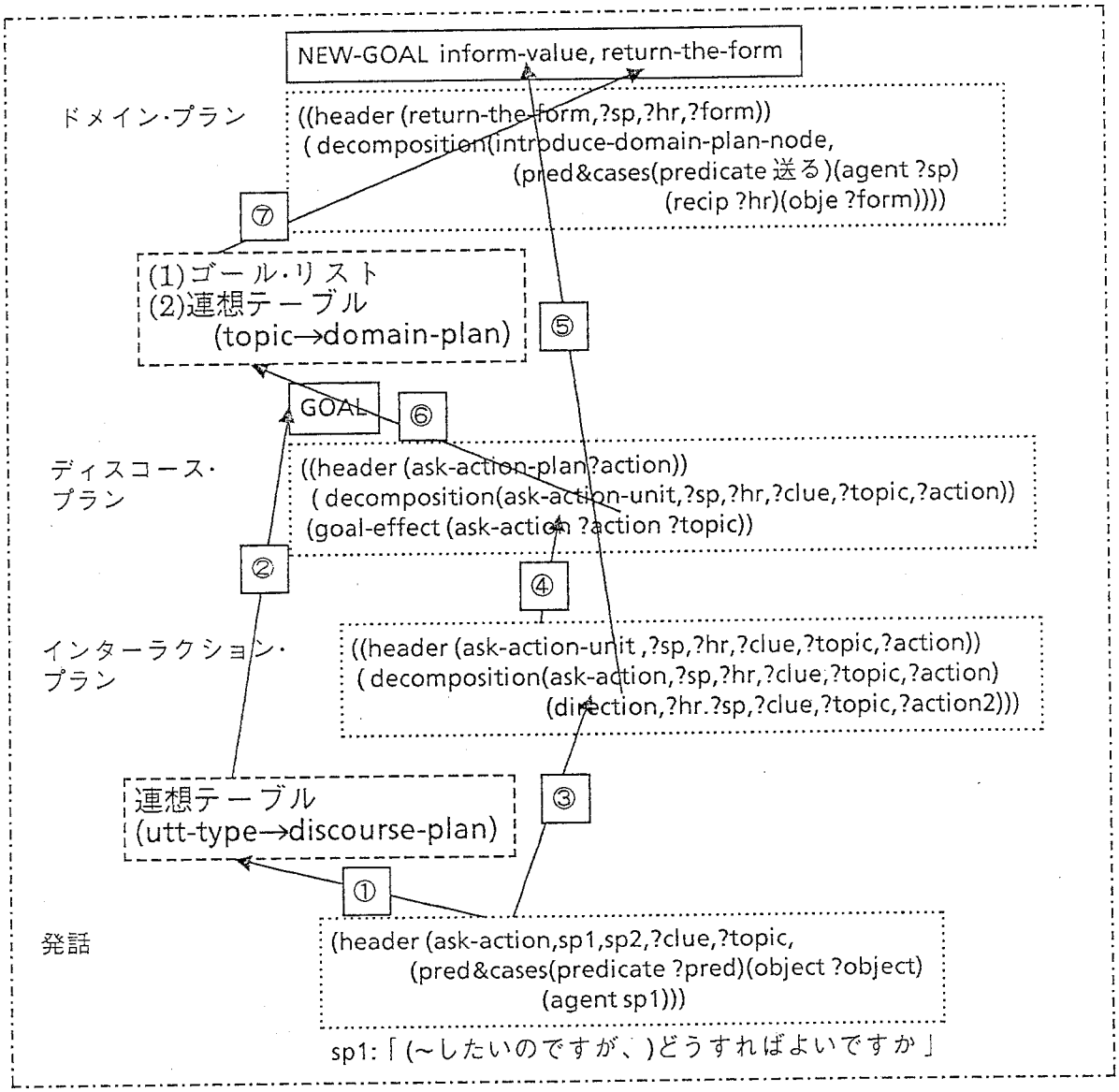
例)

①② 発話タイプがask-actionのようにActionの内容に関するものの場合は、発話タイプに関連するディスコース･プランを連想テーブルから求めて、それをゴールとする。

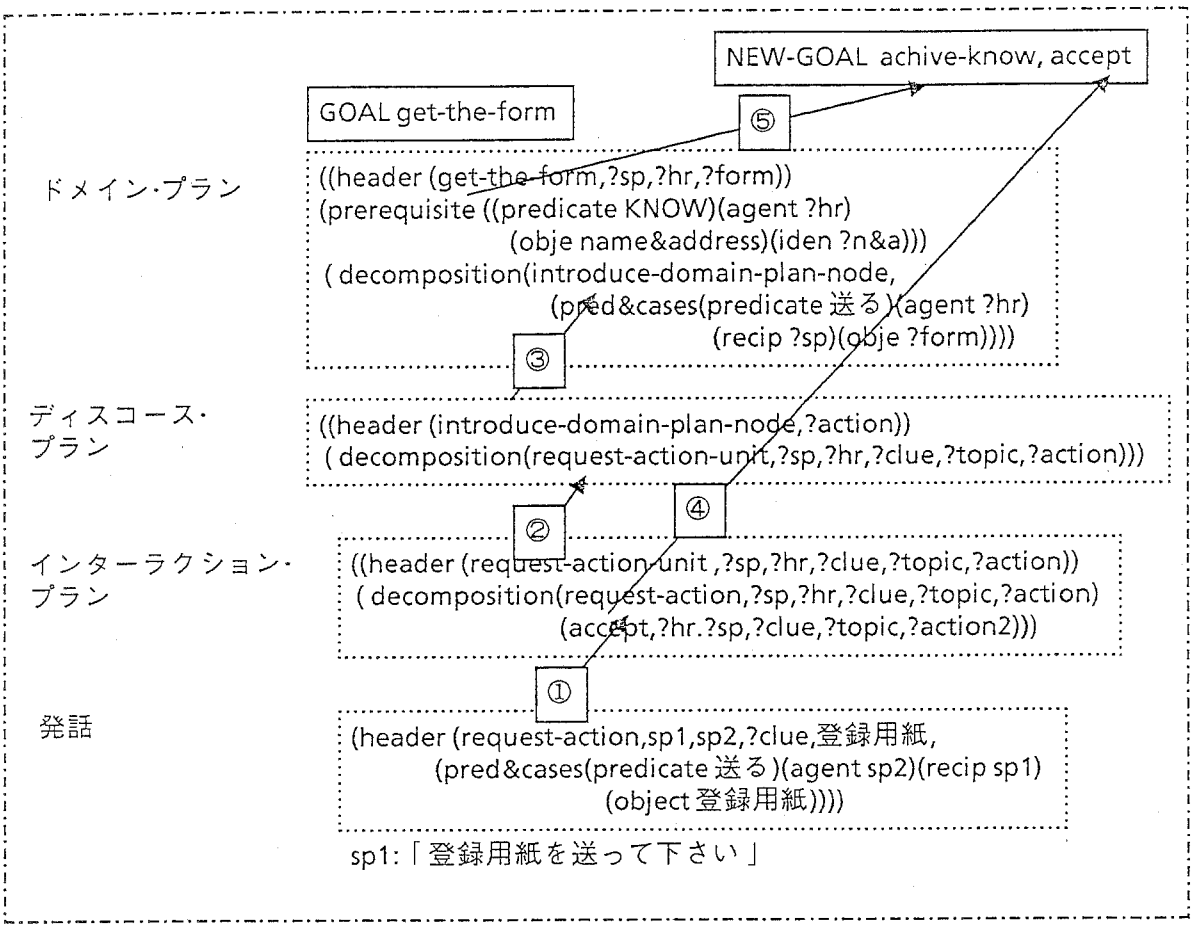
③④ ゴールに向かってdecomposition chainを求める。

⑤ decompositionが完了していないものは、新しくゴールに登録する。

⑥⑦ ゴールのeffectを実行する。ask-actionというeffectの場合は、(1)まず、ゴール･リストあるプランのPREDICATE&CASESスロットとユニファイするものを探す。(PREDICATE&CASESスロットがない場合はDECOMPOSITIONスロットのPredicate&casesの部分とユニファイするものを探す。)ユニファイするものがあれば、それをゴール･リストの先頭に移す。ユニファイするものが場合は、goal-effectのfocus-planと同様にそのトピックに関連するドメイン･プランを連想テーブルから求めて、それを新しくゴールとする。



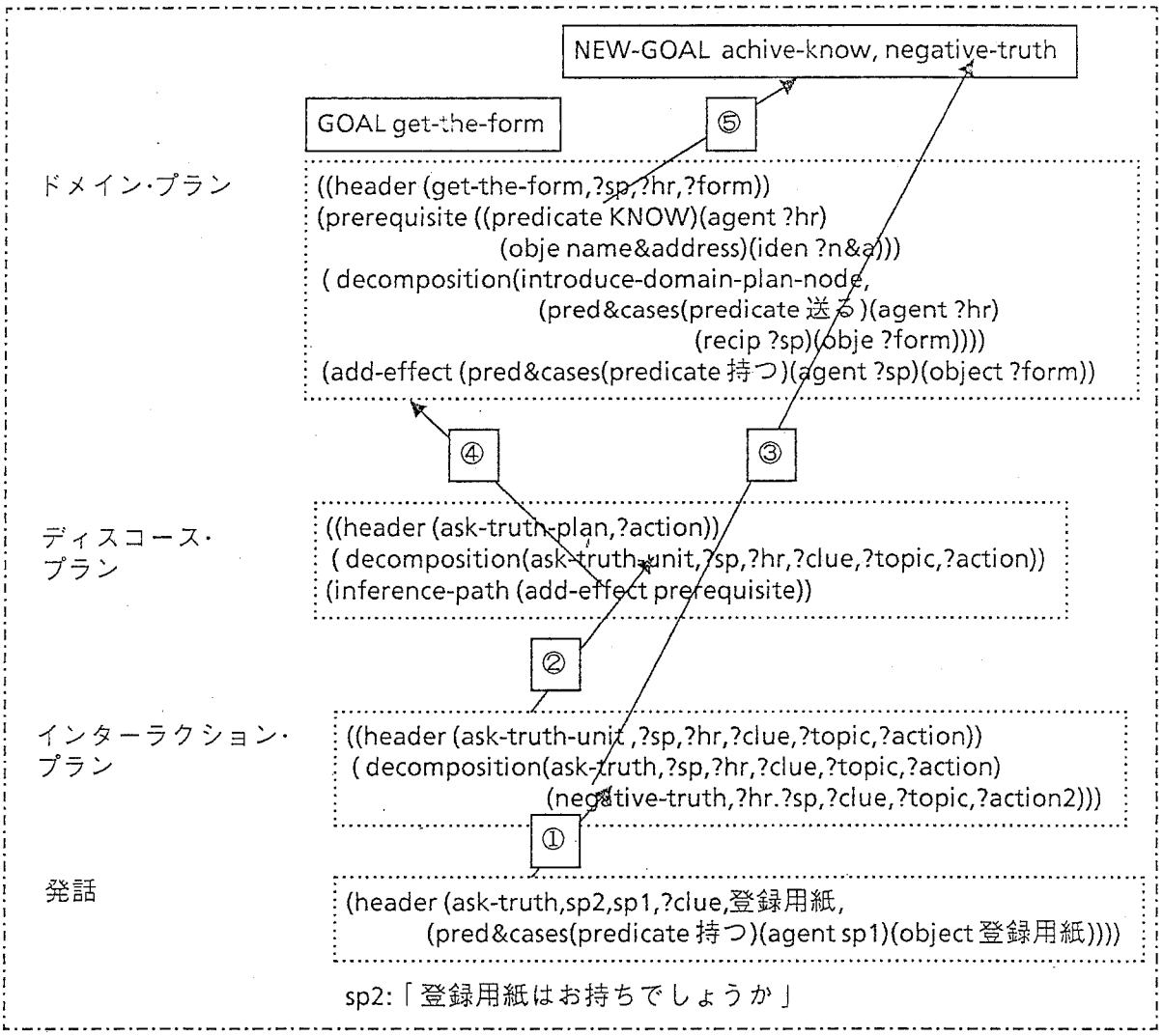
(4)Actionの実行に関するプラン
→ドメイン・プランのPredicate&Cases
例)
①②③　ゴールに向かってdecomposition chainを求める。
④　decompositionが完了していないものは、新しくゴールに登録する。
⑤　prerequisiteの条件が満たされていない場合は、それを満たすことをeffectとするものを新しくゴールに登録する。



(5)命題内容に関するプラン

①② ゴールに向かってdecomposition chainを求める。
③ Ask-Truth-Planのようにinference-pathスロットがあるプラン野ばあい、Decomposition chainによるパスの探索が失敗すると、次にadd-effectのcahinによるパスの探索を行なう。
④ decompositionが完了していないものは、新しくゴールに登録する。
⑤ prerequisiteの条件が満たされていない場合は、それを満たすことをeffectとするものを新しくゴールに登録する。

例

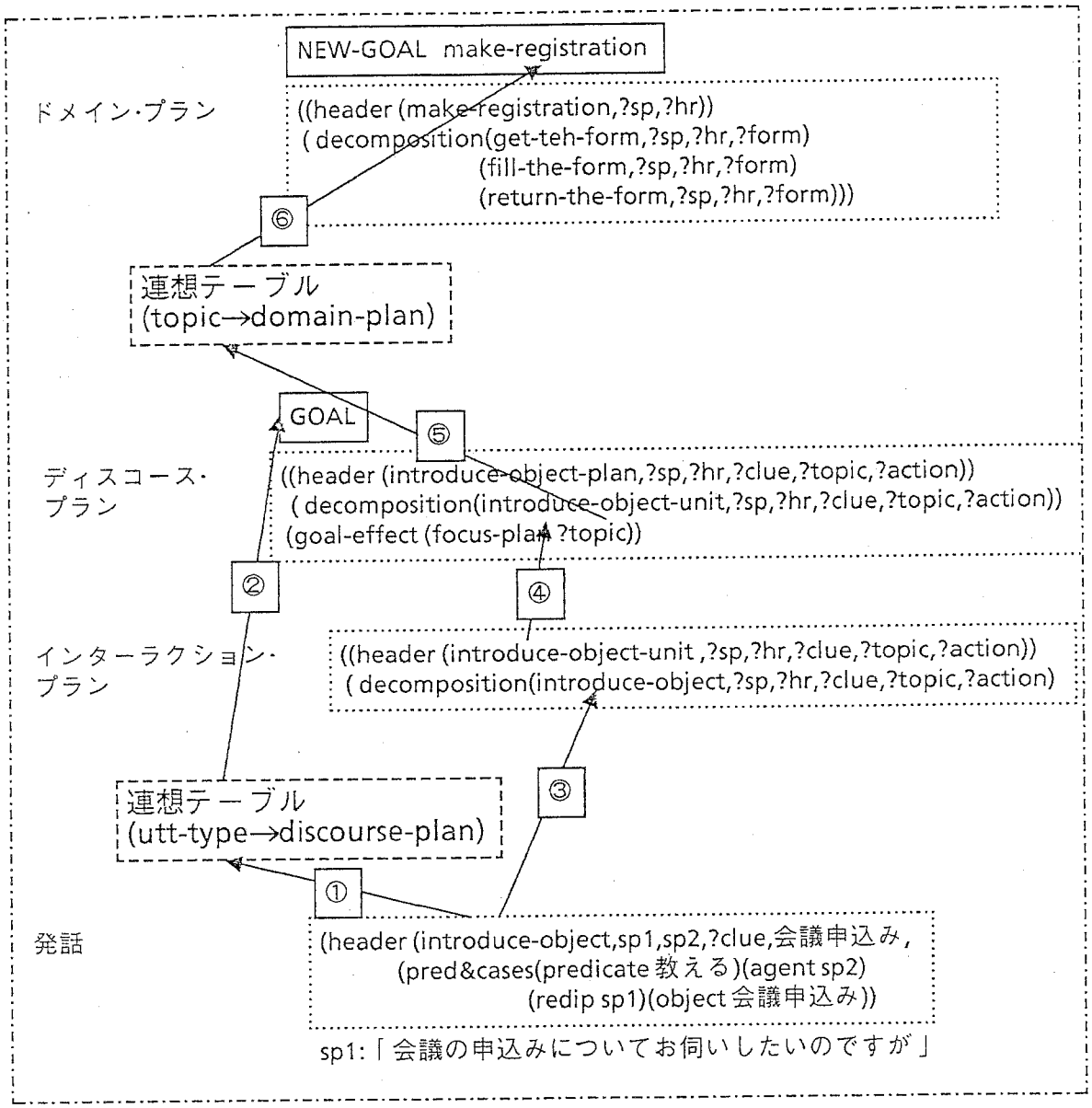


(6)話題導入に関するプラン
→ObjectのProperty→連想テーブル→ドメイン・プランのPredicate&Cases
例)

①② 発話タイプがintroduce-objectのように話題導入に関するものの場合は、発話タイプに関連するディスコース・プランを連想テーブルから求めて、それをゴールとする。

③④ ゴールに向かってdecomposition chainを求める。

⑤⑥ ゴールのeffectを実行する。focus-planというeffectの場合は、そのトピックに関連するドメイン・プランを連想テーブルから求めて、それを新しくゴールとする。



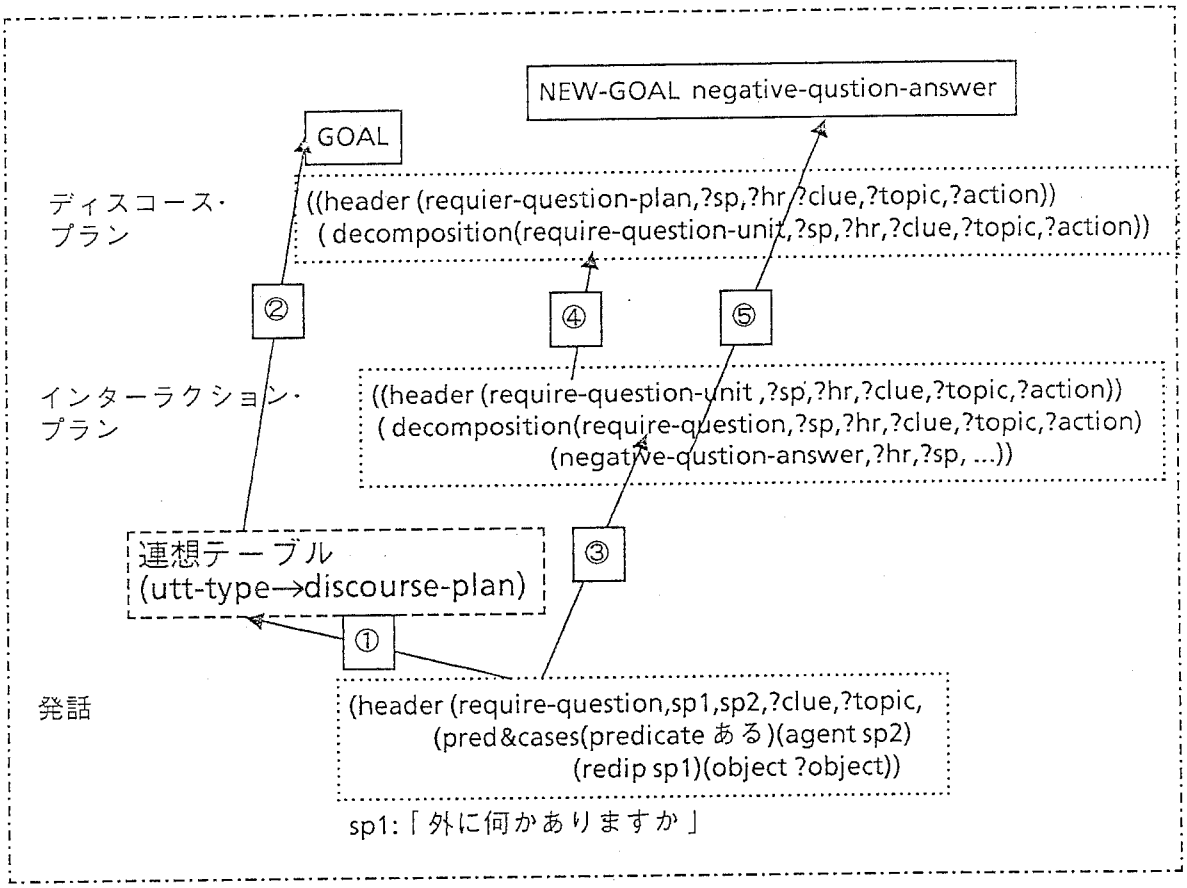
(7)Turn-Takingの起動に関するプラン

例)

①② 発話タイプがrequire-questionのようにTurn-Takingの起動に関するものの場合は、発話タイプに関連するディスコース･プランを連想テーブルから求めて、それをゴールとする。

③④ ゴールに向かってdecomposition chainを求める。

⑤ decompositionが完了していないものは、新しくゴールに登録する。



(8)情報提供に関するプラン

(a)　現在のゴールに到達するパスがみつからない場合

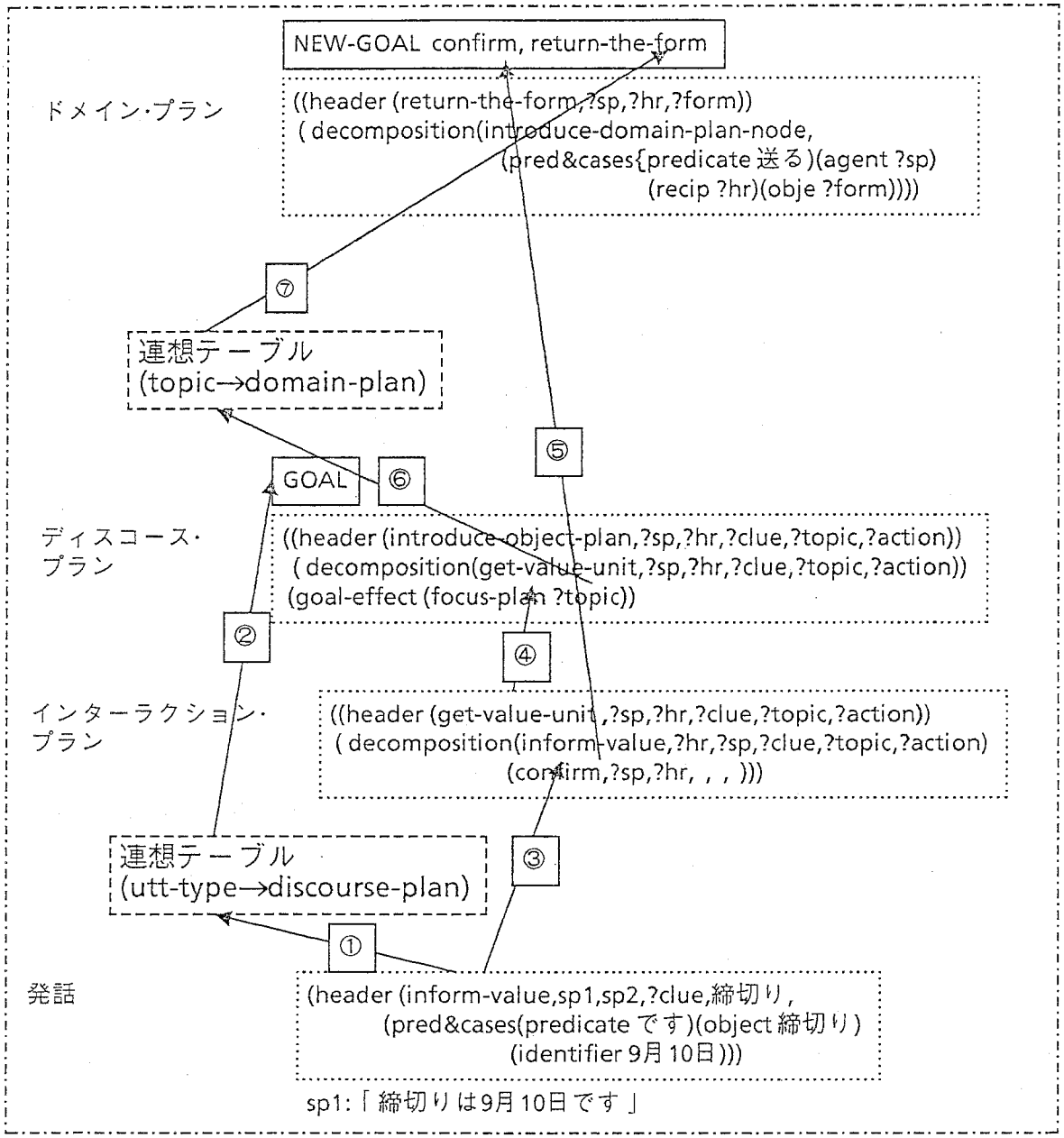
例)

①②　発話タイプがintroduce-objectのように話題導入に関するものの場合は、発話タイプに関連するディスコース・プランを連想テーブルから求めて、それをゴールとする。

③④　ゴールに向かってdecomposition chainを求める。

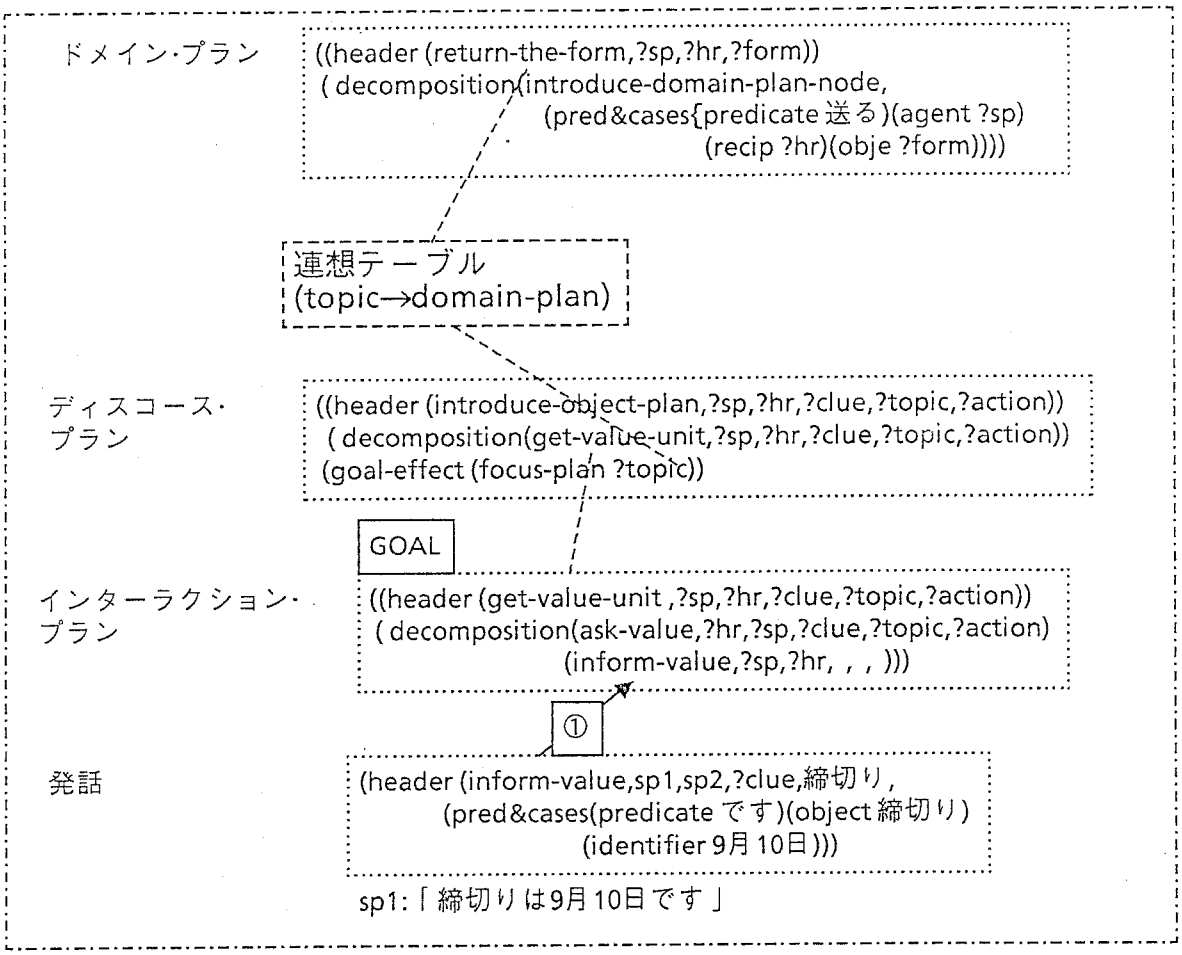
⑤　decompositionが完了していないものは、新しくゴールに登録する。

⑥⑦　ゴールのeffectを実行する。focus-planというeffectの場合は、そのトピックに関連するドメイン・プランを連想テーブルから求めて、それを新しくゴールとする。



(b)　→ObjectのProperty→インターラクション･プラン
現在のゴールに到達するパスがみつかる場合

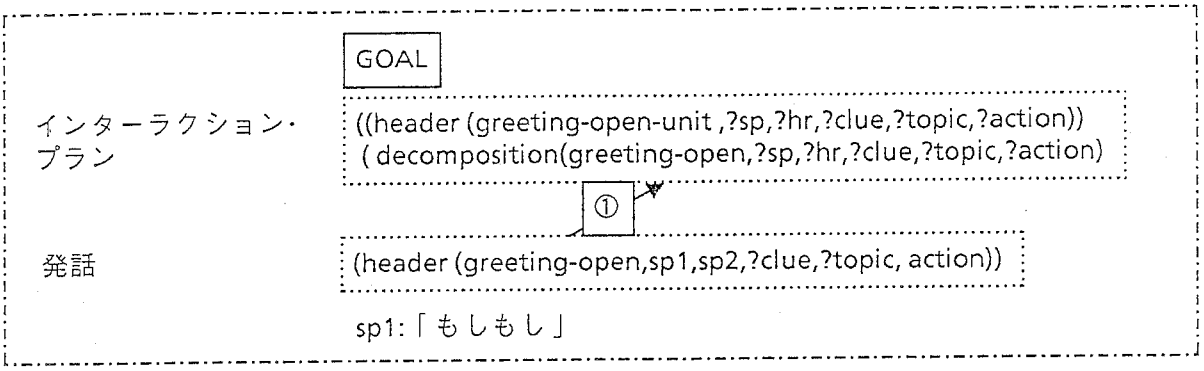
例)
①　ゴールリストにあるゴールまでのパスがみつかれば、解釈が成功する。



(9)対話のチャンネルに関するプラン

例)

① ゴールリストにあるゴールまでのパスがみつかれば、解釈が成功する。(みつからない場合は(7)と同様に連想テーブルからゴールを求めて、そのゴールまでのパスを探す。)



4. 例

以下の対話を例として処理の流れを説明する。

質問者：	登録用紙を送って下さい。	(d6-1)
事務局：	はい、分かりました。	(d6-2)
	住所とお名前をお願いします。	(d6-3)
質問者：	大阪市東区ATR、鈴木真弓です。	(d6-4)
	締切りはいつですか。	(d6-5)
事務局：	12月25日ですので	(d6-6)
	お急ぎください。	(d6-7)

Dialogue 6

システムは次の①から⑦の知識を持っているものとする。

①ドメイン･プラン

```
---------------------------------------------------------------------------------------------------
HEADER:            (MAKE-REGISTRATION ?sp1 ?sp2 ?form)
DECOMPOSITION:     (GET-THE-FORM ?sp1 ?sp2 ?form)
                   (FILL-OUT-THE-FORM ?sp1 ?sp2 ?form)
                   (RETURN-THE-FORM ?sp1 ?sp2 ?form)
PREDICATE&CASES:   ?action
CONSTRAINTS:       ?action =    (pred&cases (predicate MOUSHIKOMU)
                                            (agent ?sp1)
                                            (object ?conference))
---------------------------------------------------------------------------------------------------
HEADER:            (GET-THE-FORM ?sp1 ?sp2 ?form)
PREREQUISITE:      (pred&cases (predicate KNOW)(agent ?sp1)
                                (object address[?sp2])(identfier ?address))
DECOMPOSITION:     (Execute-Domain-Action ?sp1 ?sp2 ?action)
PREDICATE&CASES:   ?action
CONSTRAINTS:       ?action =    (pred&cases (predicate OKURU)
                                            (agent ?sp1)(recipient ?sp2)
                                            (object ?form)(to-loc ?address))
---------------------------------------------------------------------------------------------------
HEADER:            (RETURN-THE-FORM ?sp1 ?sp2 ?form)
DECOMPOSITION:     (Execute-Domain-Action ?sp1 ?sp2 ?action)
PREDICATE&CASES:   ?action
CONSTRAINTS:       ?action =    (pred&cases (predicate OKURU)
                                            (agent ?sp2)(recipient ?sp1)
                                            (object ?form)(to-loc ?address))
---------------------------------------------------------------------------------------------------
```

②ディスコース･プラン

```
---------------------------------------------------------------------------------------------------
HEADER:            (Execute-Domain-Action ?sp1 ?sp2 ?action)
```

DECOMPOSITION1: (REQUEST-ACTION-UNIT ?sp2 ?sp1 ?action)
DECOMPOSITION2: (Will-Do-Action2-Unit ?sp1, ?sp2, ?action)

---

③インターラクション･プラン

HEADER: (GET-VALUE-Unit ?sp1, ?sp2, ?action)
DECOMPOSITION: (ASK-VALUE ?sp1 ?sp2 ?clue ?topic ?action)
(INFORM-VALUE ?sp2 ?sp1 ?clue2 ?topic2 ?action)
CONSTRAINTS: ?action = (pred&cases (predicate DESU)(object ?obje)
(identifier ?ident))
EFFECTS: (pred&cases (predicate KNOW)(agent ?sp1)(object ?obje)
(identifier ?ident))

HEADER: (REQUEST-ACTION-UNIT ?sp1 ?sp2 ?action1)
DECOMPOSITION: (Request-Action ?sp1 ?sp2 ?action1)
(Accept ?sp2 ?sp1 ?action2)

---

④連想テーブル(topic→ドメイン･プラン)

締切り → RETURN-THE-FORM
PAY-THE-FEE

---

⑤連想テーブル(発話タイプ→ディスコース･プラン)

ASK-VALUE → INTRODUCE-OBJECT-PLAN

初期値
先行文脈で次のゴールが設定されているものとする。(これは「会議に登録したいのですが」などの発話によって設定される。)また共通の理解状態はなにもない。
⑥ゴールリスト：
MAKE-REGISTRATION
⑦理解状態：
nil

次に各発話の解釈について説明する。

(1)発話(d6-1)「登録用紙を送ってください」の解釈

(図4.1, 図4.2参照)

発話の解析結果は
(REQUEST-ACTION sp1 sp2 nil form
(pred&cases (predicate OKURU)(agent sp2)(recipient sp1)(object form))
となる。
これからゴール MAKE-REGISTRATION までのdecomposition chain を探索する。

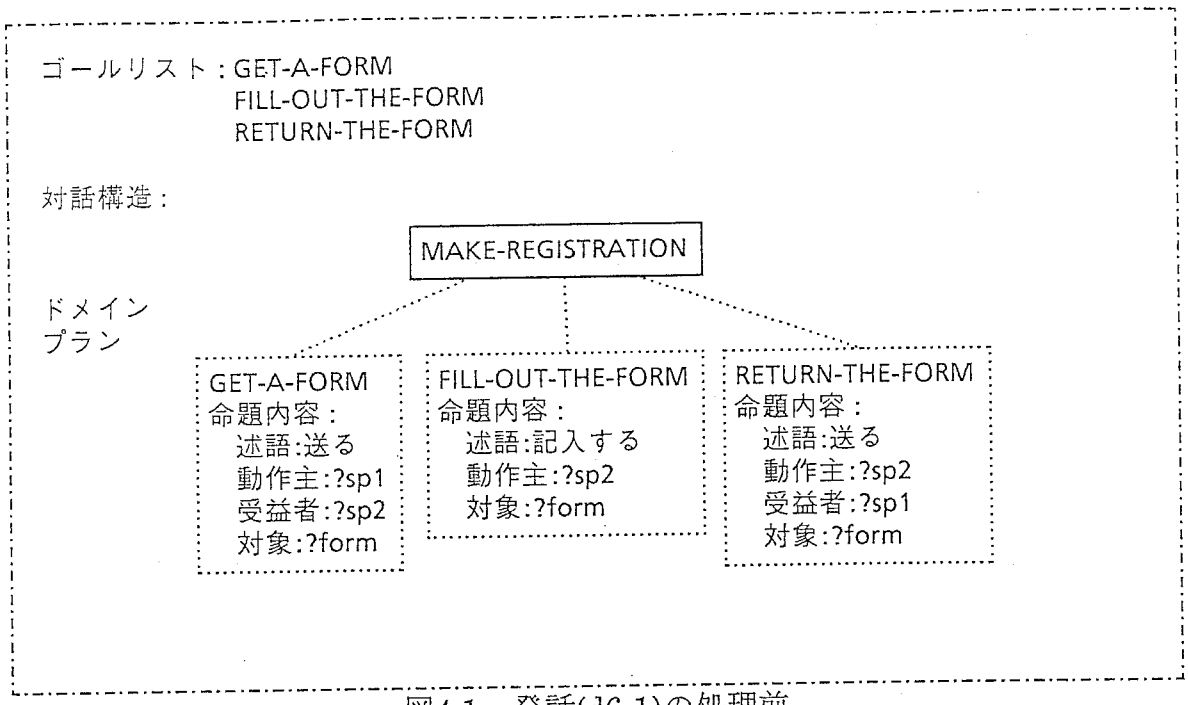


図4.1　発話(d6-1)の処理前

まず入力発話REQUEST-ACTIONはインターラクション･プランREQUEST-ACTION-UNITのDECOMPOSITIONの1つとユニファイする。

(REQUEST-ACTION sp1 sp2 nil form
(pred&cases (predicate OKURU)(agent sp2)(recipient sp1)(object form))
↓ *(decomposition chain)*
(REQUEST-ACTION-UNIT sp1 sp2
(pred&cases (predicate OKURU)(agent sp2)(recipient sp1)(object form))

REQUEST-ACTION-UNITのDECOMPOSITIONのもうひとつのDECOMPOSITION (Accept sp2 sp1 ?action2)はゴールリストに入れられる。
次にインターラクション･プランREQUEST-ACTION-UNITはディスコース･プランExecute-Domain-ActionのDECOMPOSITIONの1つとユニファイする。

(REQUEST-ACTION-UNIT sp1 sp2
(pred&cases (predicate OKURU)(agent sp2)(recipient sp1)(object form))
↓ *(decomposition chain)*
(Execute-Domain-Action sp1 sp2
(pred&cases (predicate OKURU)(agent sp2)(recipient sp1)(object form))

次にディスコース･プランExecute-Domain-Actionはドメイン･プランGET-THE-FORMのDECOMPOSITIONとユニファイする。

(Execute-Domain-Action sp1 sp2
(pred&cases (predicate OKURU)(agent sp2)(recipient sp1)(object form))
↓ *(decomposition chain)*

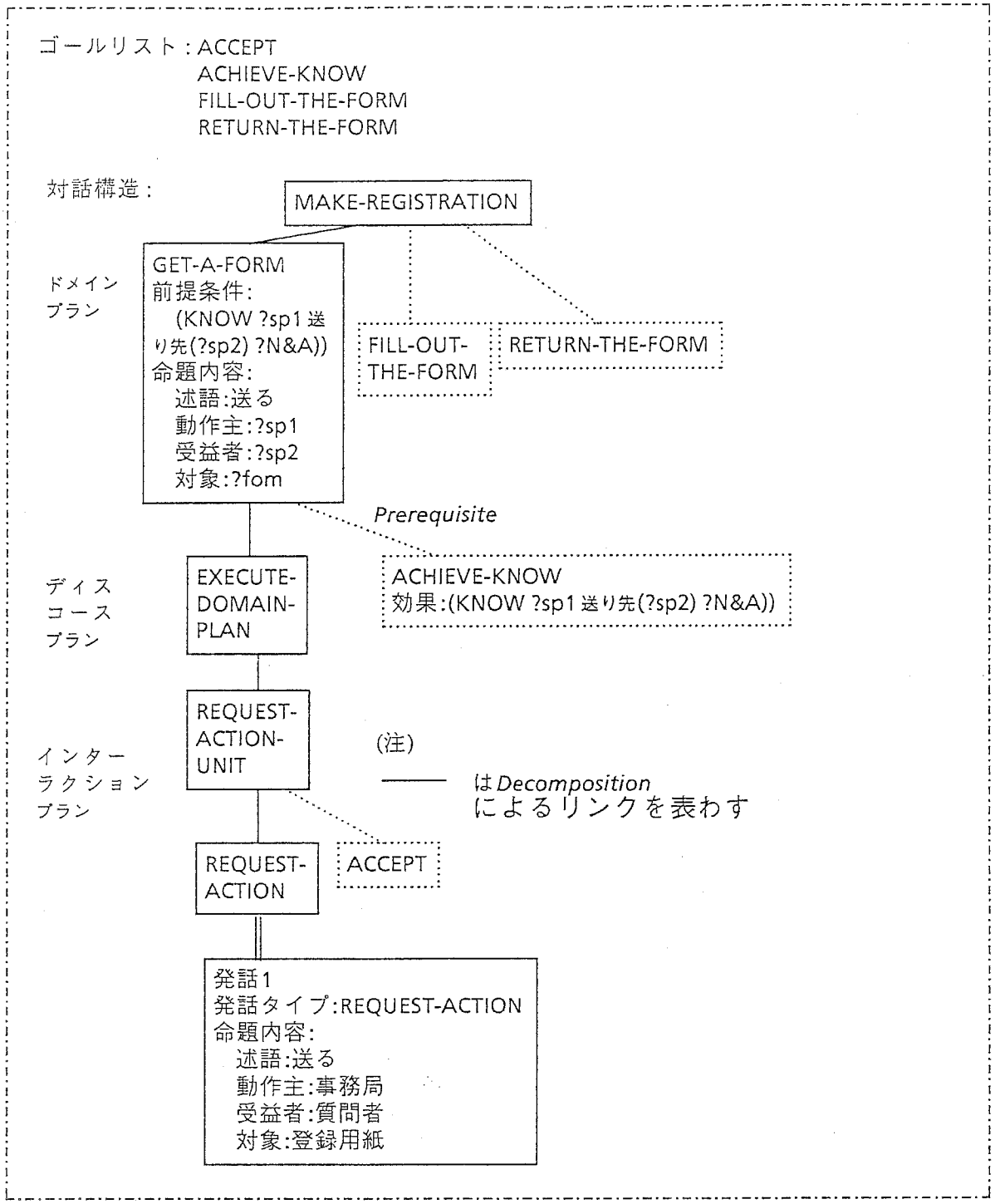


図4.2　発話(d6-1)の処理後

(GET-THE-FORM sp1 sp2 form)

PREREQUISITEを理解状態が満たしているこどうか調べる。満たしていないので

(Achive-know (pred&cases(predicate KNOW)(agent sp2)

(object address)(identifier ?address)))

がゴールリストに入れられる。
次にドメイン･プランGET-THE-FORMはドメイン･プランMAKE-REGISTRATIONのDECOMPOSITIONとユニファイする。

(GET-THE-FORM sp1 sp2 form)
↓ *(decomposition chain)*
(MAKE-REGISTRATION sp1 sp2 form)

MAKE-REGISTRATION の残りのDECOMPOSITION (FILL-OUT-THE-FORM sp2 form) (RETURN-THE-FORM sp2 form) はゴールリストに入れられる。

ゴール MAKE-REGISTRATION までのパスが見つかったので、発話(d6-1)の解釈は成功した。
以上のことから発話(d6-1)の解釈後のシステムの状態は次のようになる。

ゴールリスト：
(Accept sp2 sp1 ?clue ?topic ?action2)
(GET-VALUE-UNIT sp2 sp1 ?clue ?topic
(pred&cases(predicate desu)
(object name&address)(identifier ?ident))
(FILL-OUT-THE-FORM sp2 sp1 form)
(RETURN-THE-FORM sp2 sp1 form)

理解状態： nil

(2)発話(d4-2)「はい、わかりました」の解釈

発話の解析結果は
(Accept sp2 sp1 nil nil (pred&cases (predicate WAKARU))
と
(Confirm sp2 sp1 nil nil (pred&cases (predicate WAKARU))
の2つの可能性が与えられる。Acceptはゴールリストの(Accept sp2 sp1 ?clue ?topic ?action2)とユニファイが成功するのでAcceptの解釈は成功する。Confirmはゴールリストのゴール Accept, GET-VALUE-UNIT, FILL-OUT-THE-FORM, RETURN-THE-FORMのどれにもパスがみつからず、かつAcceptの解釈が成功しているのでConfirmの解釈は失敗する。

以上のことから発話(d6-2)の解釈後のシステムの状態は次のようになる。

ゴールリスト：
(GET-VALUE-UNIT sp2 sp1 ?clue ?topic
(pred&cases(predicate desu)
(object name&address)(identifier ?ident))
(FILL-OUT-THE-FORM sp2 sp1 form)
(RETURN-THE-FORM sp2 sp1 form)

理解状態： nil

(3)発話(d6-3)「住所とお名前をお願いします」の解釈

発話の解析結果は

(ASK-VALUE sp2 sp1 ?clue name&address
(pred&cases (predicate desu)(object name&address)(identifier ?ident))

となる。これからゴールリストのゴールGET-VALUE-UNITまでのdecomposition chain を探索する。

(ASK-VALUE sp2 sp1 ?clue name&address
(pred&cases (predicate desu)(object name&address)(identifier ?ident))

と

(GET-VALUE-UNIT sp2 sp1 ?clue ?topic
(pred&cases(predicate desu)(object name&address)(identifier ?ident))

はユニファイするのでこの解釈は成功する。
GET-VALUE-UNITのdecompositionのもうひとつのdecomposition の (INFORM-VALUE sp1 sp2 ?clue2 ?topic2 (pred&cases(predicate desu)(object name&address) (identifier ?ident)) がゴールリストに入れられる。

以上のことから発話(d6-3)の解釈後のシステムの状態は次のようになる。

ゴールリスト： (INFORM-VALUE sp1 sp2 ?clue ?topic
(pred&cases(predicate desu)
(object name&address)(identifier ?ident))
(FILL-OUT-THE-FORM sp2 sp1 form)
(RETURN-THE-FORM sp2 sp1 form)

理解状態： nil

(4)発話(d6-4)「大阪市東区ATR、鈴木真弓です。」の解釈

(図4.3参照)

発話の解析結果は

(INFORM-VALUE sp1 sp2 ?clue name&address
(pred&cases (predicate desu)(object name&address)
(identifier Suzuki-MAyumi&Oosaka-ATR))

となる。これからゴールリストのゴールINFORM-VALUEまでのdecomposition chain を探索する。これはユニファイするので解釈は成功する。この時INFORM-VALUEの親のプランGET-VALUE-UNITも成功するので、そのEFFECTが実行される。

以上のことから発話(d6-4)の解釈後のシステムの状態は次のようになる。

ゴールリスト： (FILL-OUT-THE-FORM sp2 sp1 form)
(RETURN-THE-FORM sp2 sp1 form)

理解状態： (pred&cases (predicate KNOW)(agent sp2)
(objectname&address)(identifier Suzuki-MAyumi&Oosaka-ATR))

(5)発話(d6-5)「締切りはいつですか」の解釈

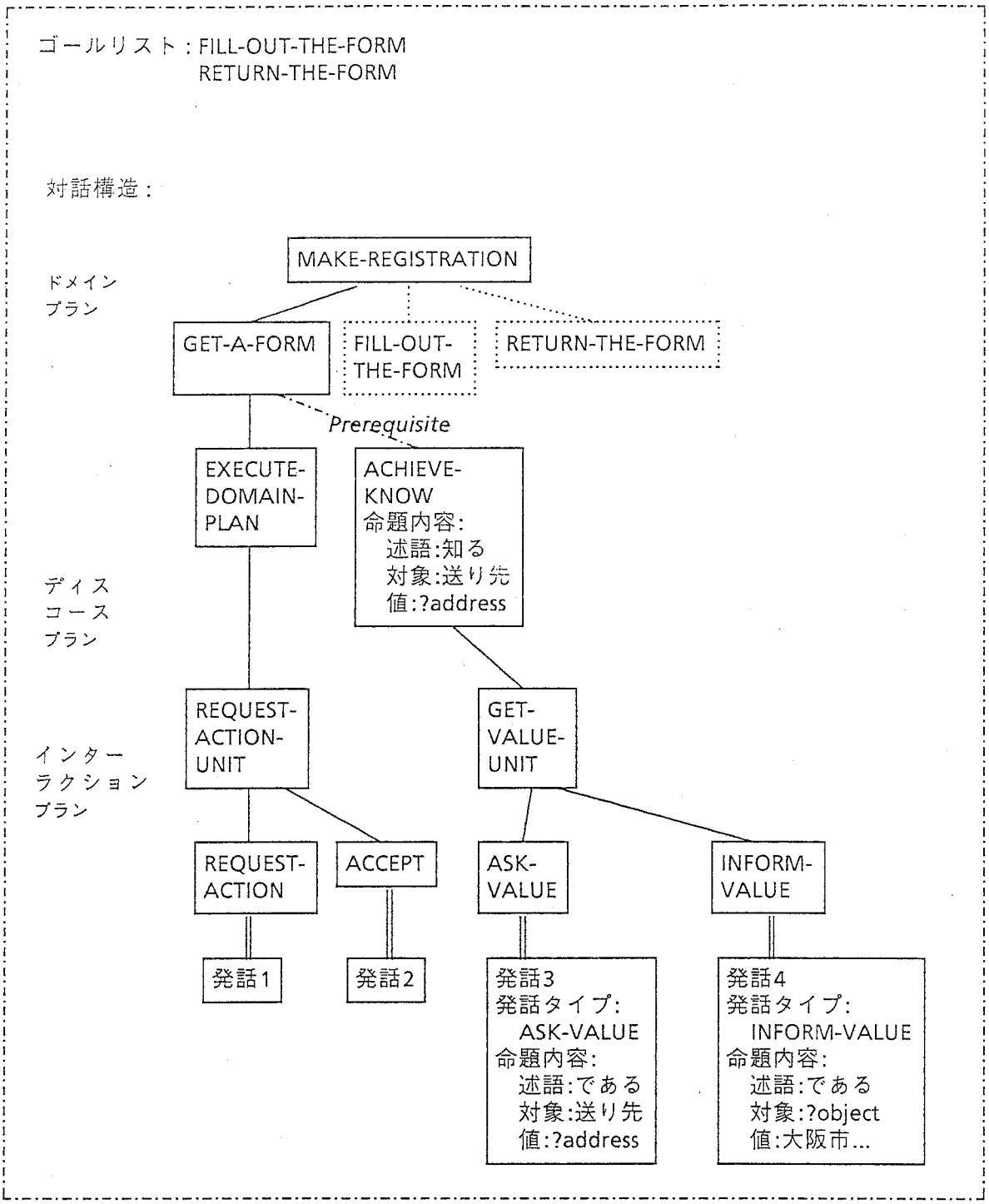


図4.3　発話(d6-4)の処理後

(図4.4参照)

発話の解析結果は

(ASK-VALUE sp1 sp2 ?clue shimekiri

(pred&cases (predicate desu)(object shimekiri)(identifier ?ident))

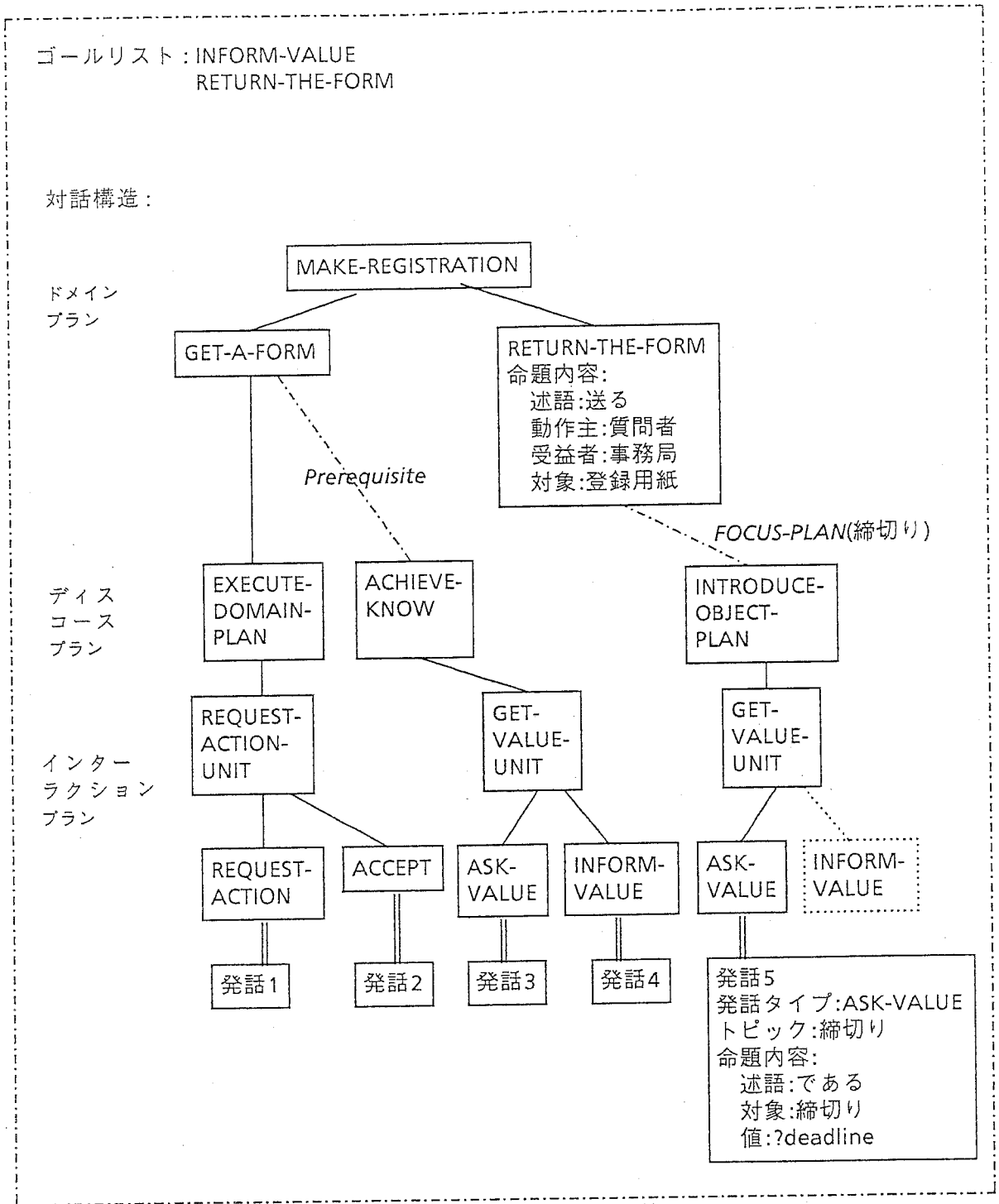


図4.4　発話(d6-5)の処理後

となる。これからゴールリストのゴールFILL-OUT-THE-FORMまでのdecomposition chainを探索するが失敗する。次にゴールリストのゴールRETURN-THE-FORMまでのdecomposition chainを探索するがやはり失敗する。そこで連想テーブル(発話タイプ→ディスコース･プラン)からASK-VALUEに対応するプランINTRODUCE-OBJECT-PLANがゴールとして設定される。ASK-VALUEからINTRODUCE-OBJECT-PLANまでのdecomposition chainは見つけられるのでこの解釈は成功する。

(ASK-VALUE sp1 sp2 ?clue shimekiri
(pred&cases (predicate desu)(object shimekiri)(identifier ?ident))
↓ *(decomposition chain)*
(GET-VALUE-UNIT sp1 sp2
(pred&cases (predicate desu)(object shimekiri)(identifier ?ident))
↓ *(decomposition chain)*
(INTRODUCE-OBJECT-PLAN sp1 sp2
(pred&cases (predicate desu)(object shimekiri)(identifier ?ident))

このときGET-VALUE-UNITのdecomposition INFORM-VALUEがゴールリストに入れられる。

以上のことから発話(d6-4)の解釈後のシステムの状態は次のようになる。

ゴールリスト：
(INFORM-VALUE sp2 sp1 ?clue ?topic
(pred&cases(predicate desu)
(object shimekiri)(identifier ?ident))
(FILL-OUT-THE-FORM sp2 sp1 form)
(RETURN-THE-FORM sp2 sp1 form)

理解状態：
(pred&cases (predicate KNOW)(agent sp2)
(object name&address)(identifier Suzuki-MAyumi&Oosaka-ATR))

(6)発話(d6-6)「12月25日ですので」の解釈

発話の解析結果は
(INFORM-VALUE sp2 sp1 nil nil
(pred&cases (predicate desu)(object ?obje)(identifier December-25))
となる。これからゴールリストのゴールINFORM-VALUEまでのdecomposition chainを探索する。これはユニファイするので解釈は成功する。この時INFORM-VALUEの親のプランGET-VALUE-UNITも成功するので、そのEFFECTが実行される。またINTRODUCE-OBJECT-PLANも成功するのでそのEFFECTが実行され、shimekiriに関連するプランとしてRETURN-THE-FORMがゴールリストの先頭に来る。

以上のことから発話(d6-4)の解釈後のシステムの状態は次のようになる。

ゴールリスト：
(RETURN-THE-FORM sp2 sp1 form)
(FILL-OUT-THE-FORM sp2 sp1 form)

理解状態：
(pred&cases (predicate KNOW)(agent sp2)
(object name&address)(identifier Suzuki-MAyumi&Oosaka-ATR))
(pred&cases (predicate KNOW)(agent sp1)
(object shimekiri)(identifier December-25))

(7)発話(d6-7)「お急ぎ下さい」の解釈

発話の解析結果は

(DIRECTION sp2 sp1 nil nil
(pred&cases (predicate ?pred)(agent sp1)(manner Isoide))

となる。これからゴールリストのゴールRETURN-THE-FORMまでのdecomposition chainを探索する。これはユニファイするので解釈は成功する。その結果、発話の解釈は

(DIRECTION sp2 sp1 nil nil (pred&cases (predicate okuru)(agent sp1)
(recipient sp2)(object form)(manner Isoide))

となり、「(sp1がsp2にformを送ることを)お急ぎ下さい」と解釈できる。

以上のことから発話(d6-4)の解釈後のシステムの状態は次のようになる。

ゴールリスト：　(FILL-OUT-THE-FORM sp2 sp1 form)
理解状態：　(pred&cases (predicate KNOW)(agent sp2)
(object name&address)(identifier Suzuki-MAyumi&Oosaka-ATR))
(pred&cases (predicate KNOW)(agent sp1)
(object shimekiri)(identifier December-25))

5. 自動翻訳電話への応用

5.1 省略の解釈

省略された要素はプラン･スキーマの引数の対応関係から復元することができる。T-Planモデルではドメイン･プランを利用することにより、格要素だけでなく述語の省略も復元できる。

第4章の例で示したように、

sp1:「締切りはいつですか」
sp2:「12月25日ですので、お急ぎください」

のsp2の発話は「(sp1がsp2にformを送ることを)お急ぎ下さい」と解釈できる。

5.2 訳語選択

日本語にはドメインではなく、文脈によって意味の決まる多義句がある。対話には2種類の文脈がある。

①トピックの流れ

"Johen was writing a letter on a plane to IJCAI-87. The ink smeared. He said the quality of this paper is terrible."[Tomabechi 87]

paperの意味がthesisかsheetかはトピックの流れで決まる。これはT-Planモデルではドメイン･プランで認識される。

②発話タイプの流れ

日本語の「わかりました」は英語のI see., Thank you., All right., I understand., That will be fine., OK, Good.,Very well., I will do that., That sounds good. などに対応する。「わかりました」が物理的動作を要求する発話(例 「締切りまでに申込み用紙を返送して下さい。」)の応答である場合はI will do that., All right. などが対応する。相手に情報を尋ねて、その内容を相手が応答したあとに確認の意味で使う場合は、I see., I understand. などが対応する。相手から一連の情報を得て、対話を終わろうとする場合はThank you. が対応する。相手に選択を求めて、その中の一つを相手が選択して、それに対する場合はThat will be fine., OKなどが対応する。

これらの訳し分けのための情報はT-Planモデルではインターラクション･プランから得ることができる。

5.3 次発話の予測 [Arita and Iida 88a 88b]

[次発話の予測手法]

① インターラクション･プランによる予測インターラクション･プランによる予測の例を下に示す。

◇ demand classに属する発話がなされた次には対応するresponse classに属する発話があると予測できる。

② inform classに属する発話の次にはconfirmation classに属する発話があると予測できる。

② ドメイン·プランによる予測

ドメイン·プランによる予測の例を下に示す。

① 話題が大きく変化しなければ、あるアクションが完了すればドメイン·プランに記述されている次のアクションに関連する発話があると予測される。

② アクションの変数が定まっていなければ、その変数を定めるための発話がなされると予測できる。

③ 対話のストラテジーによる発話の組立て

対話は要求の発話とその応答の発話から構成されるのが基本であるが、相手が要求·応答の内容を理解することを助けるために、理由や話題導入の語句(「~についてお伺いしたいのですが。」など)を発話する。要求·応答の発話と理由·話題導入などの発話の間には一定の関係がある。それらを対話のストラテジーとして整理することにより、次発話を予測できる。対話のストラテジーの例を下に示す。

① 応答する-->発話タイプに対する応答+スピーチ·アクトに対する応答
② 質問する-->質問の理由+質問の内容

この予測手法の評価(机上調査)については[Arita and Iida 88b]を参照。

6. 関連研究

関連研究として、[Allen and Perrault 80]、[Pollack 86]、[Litman and Allen 87]、[Kitahashi 88]の研究について簡単に説明する。

① [Allen and Perrault 80]
Allen等は相手のBeliefやgoalのモデルによってHelpful Responseを説明している。(付録 A3参照)

Allen等が対象とした対話を以下に示す。(駅構内での客と駅員の対話)

Example 1 : Providing more information than requested
"When does the train to Windsor leave?"
"The train leaves at 1600. The train leaves from gate 7."
Example 2 : A yes/no question answered no
"Does the Windsor train leave at 4?"
"No. The train leaves at 4:30."
Example 3 : An indirect speech act
"Do you know when the Windsor train leaves?"
"Yes, at 3:15."
Example 4 : a sentence fragment
"The train to Windsor?"
"It leaves at 3:15 from gate 7."

Example 3について簡単に説明する。
(1) A : Do you know when the Windsor train leaves?
(2) S : Yes, at 3:15.

(1)の発話から推論される表層上のゴールは、次のようになる。
(3) A KNOWIF (S KNOWIF 'departure time').
know-positive rule : SBAW(A KNOWIF P) = $i\Rightarrow$ SBAW(P)† を適用すると、次のゴールを得る。
(4)S KNOWIF 'departure time'
precondition-action rule : SBAW(P) = $i\Rightarrow$ SBAW(ACT) if P is a precondition of action ACT から、次のゴールを得る。
(5)INFORMREF(S,A,'departure time')
action-effect rule : SBAW(ACT) = $i\Rightarrow$ SBAW(E) if E is an effect of ACT から、次のゴールを得る。
(6) A KNOWREF 'departure time'
Sは(3)から"Yes"と(6)から"at 3:15"を生成して応答(2)をする。

T-Plan Modelでは(1)は次のように解析される。
(7) (ASK-REF A S nil 'departure time'

(注)† SBAWは S believe A want の略。
The notation SBAW(X) = $i\Rightarrow$SBAW(Y) indicates that if S believes A's plan contains X, then S may infer that A's plan also contains Y.

(pred&cases (predicate DESU)(object 'departure time')(identifier ?id)))

インターラクション・プラン

(8) header: (ask-ref-unit ?sp ?hr ?clue ?topic ?action)
decomposition: (ask-ref ?sp ?hr ?clue ?topic ?action)
(affirmative ?hr ?sp.nil nil ?action2)
(inform ?hr ?sp?nil ?topic ?action)

のdecompositionの1番目と(7)がユニファイし、応答(2)の"Yes"はaffirmativeに、"at 3:15"はinformにユニファイして解釈できる。

② [Pollack 86]

Pollackは"I want to perform an act of β, so I need to find out how to perform an act of α."というようにgoalが明示的な場合について、話者の質問の前提に誤りがある場合の協調的な応答(cooperative response)について研究している。

Pollackが対象とした対話を以下に示す。(電子メールに関する質問ー応答)

"I want to prevent Tom from reading my mail file. How can I set the permission on it to faculty-read only?"

に対して、応答者(R)は質問者(Q)のBeliefを

(1)BEL(R,BEL(Q,EXEC(set-permissions(mmfile,read,faculty),Q)))
(2)BEL(R,BEL(Q,EXEC(prevent(mmfile,read,tom),Q)))
(3)BEL(R,BEL(Q,GEN(set-permissions(mmfile,read,faculty),
prevent(mmfile,read,tom),Q)))

原論文には時間のパラメータがあるが、ここでは省略した。また各関係の意味は次のとおりである。

The relation BEL(G,P,t) should be taken to mean that agent G beliebes proposition P throughout time interval t.

The relation EXEC(α,G,t) is true if and only if the act of G doing α at t is execurable.

The relation GEN(α,β,G,t) is true if and only if the act of G doing α at t genarates the act G doing β at t.

と推論して、それらに誤りがない場合は、

"Type SET PROTECTION."

と答える。(1)に誤りがある場合は、

"There is no way for you to set the permission on a file to faculty-read only. What you can do is move it into a password-protect subdirectory; that will prevent Tom from reading it."

と答える。(1)(2)に誤りがある場合は、

"There is no way for you to set the permission on a file to faculty-read only, nor is there any way for you to prevent Tom from reading it."

と答える。(3)に誤りがある場合は、

"Well, the command is SET PROTECTION, but that won't keep Tom out: file permissions don't apply to the system manager."

と答える。(1)(3)に誤りがある場合は、

"There is no way for you to set the permissions to faculty-read only; and even if you could, it wouldn't keep Tom out: file permissions don't apply to the system manager."

と答える。さらに詳細なモデルで次の答えも説明する。

"Well, the command is SET PROTECTION, but that won't keep Tom out: he's the system manager."

"Well, the command is SET PROTECTION, but that won't keep Tom out: he's the system manager, and file permissions don't apply to the system manager."

T-Plan Modelでは現在のところ、このような質問の前提に誤りがある問題は扱っておらず今後の課題のひとつである。

③ [Litman and Allen 87]

Allen等やPollackは一つの対話対を考察の対象にしていたの対してLitman等はdomain planとdiscourse plan に分けてSubdialogueを考察の対象にしている。各発話とdomain planとの対応を明らかにしている。(付録 A4参照)

Litman等が対象とした対話を以下に示す。(駅構内での客と駅員の対話)

Passenger :	The eight-fifty to Montreal?
Clerk :	Eight-fifty to Montreal. Gate seven.
Passenger :	Where is it?
Clerk :	Down this way to the left. Second one on the left.
Passenger :	OK. Thank you.

T-Plan Model はAllen等の考え方をベースにしているが、その違いはT-Plan Model が対話対というレイヤーを明確に意識し、それに対応するインターラクション･プランを設けていることである。

④ [Kitahashi 88]

北橋等は対話対の間の関係として(1)前提条件、(2)詳細化、(3)結果報告、(4)繰り返し、(5)派生という関係でとらえようとしている。

北橋等が対象とした対話を以下に示す。(研究会の受付での対話)

受付1： 恐れ入りますがお名前をお願いします。
来客1： 山田と申しますが。
受付2： 申込みはお伺いしてますでしょうか。
来客2： 申し込んだはずなんですけどね。
3：大阪大学産研の山田で送ったかな。
受付3： 大阪大学産研の山田
4：ちょっと無いみたいですね。
来客4： すみません。
受付5： 懇親会はどうなされますか。
来客5： 懇親会ですか、どうしようかな。
受付6： 出席なさいますか。
来客6： はい、出席します。
7：いくらですか。
受付7：で は、懇親会費が千円になるんですが。
8：領収書の方は
来客8： はい

受付9： 山田様宛でよろしいでしょうか。
来客9： はい。
受付10： あちらの方のパンフレットと名札、名札にお名前をお願いします。
来客11： どこですか、場所は、
受付11： つきあたり、右手奥になっております。

前提条件: 受付1-来客1の対は、申込みの確認出ある受付2-来客2の発話を可能にしている。また受付1-来客1の対は、懇親会の確認である受付5-来客5の発話も可能にしている。

詳細化: 受付3-来客3は受付2-来客2の詳細化である。

結果報告: 受付4-来客4は申込みの確認を行っている受付2-来客2の結果の報告と確認である。

繰り返し: 受付6-来客6は受付5-来客5と同一内容の発話である。

派生: 受付5-来客5は「懇親会」に関する情報の授受であり、受付7-来客7は「懇親会の会費」の情報の授受である。「懇親会」と「費用」の関係は、対象とその属性という関係である。このように授受されている情報がある対象からその対象の1つの属性に移行する関係を派生と言う。また受付7-来客7から受付8-来客8のように、授受されている情報がある対象からそれに付随する対象に移行する関係も派生と呼ぶ。

この研究は始まったばかりであり、基本的な考え方を示したものである。応用としてはスクリプトの生成を考えている。

各研究の対象･アプローチの違いをまとめると次のようになる。

	1つのturn-taking	subdialogue
質問の前提条件が正しい	[Allen and Perrault 80]	T-Plan Model [Litman and Allen 87] [Kitahashi 88]
質問の前提条件に誤りがある	[Pollack 86]	

	domainの構造と発話の対応がとれる	domainの構造と発話の対応が明確でない
対話対を基本構造としている	T-Plan Model	[Kitahashi 88]
対話対を基本構造としていない	[Litman and Allen 87]	

7. おわりに

対等な話者間で交わされる目標指向型対話の理解のためのプラン認識モデルについて述べた。3層の階層的プラン認識モデルを提案した。3層とは対話のトピックの構造を表わすドメイン･プラン、対話のターン･テイキングを表わすインターラクション･プラン、そしてドメイン･プランとインターラクション･プランを結び付けるディスコース･プランである。ドメイン･プランはドメインに依存する知識であるが、ディスコース･プラン、インターラクション･プランはドメインに依存しない知識である。このモデルは自動翻訳電話において省略の解釈、訳語選択、次発話の予測などに応用できる。

[問題点、今後の課題]

① Demandの前提が間違っている場合

A:「参加費はいつ支払うのですか」
B:「参加費は無料です」

のようにDemandの前提が間違っている対話対も今後、扱えるようにする必要がある。

② 対話対として機械的に処理しにくいもの　ー名詞表現と動詞表現ー

次のような対話対はT-plan Modelでは対話対としてとらえることができない。

(1)参加費はどのようにお支払いしたらよいですか。
(2)参加費は銀行振り込みです。

T-plan Modelで扱えるのは次のような対話対である。

(1')(?)参加費の支払方法は何ですか。
(2)参加費は銀行振り込みです。

(1)参加費はどのようにお支払いしたらよいですか。
(2')参加費は銀行へ振り込んでください。

③ ゴールリストの管理

T-Plan modelでは一度ゴールリストに入れられたゴールは、そのゴールに関連する発話があるまでゴールリストから消去されない。しかし話題が大きく変わった場合など、あきらかにゴールリストから消去してもよい(消去しなければならない)ゴールを見つけなければならない。すなわちゴール間の関係を考慮しなければならない。

④ ドメイン･プランの記述基準

対話の各発話とドメイン･プランのノードがうまく対応するためには、対話の内容の深さとドメイン･プランの記述の深さが同じくらいでなければならない。

⑤ ドメイン･プランの探索

アクションの実行に関する発話のようにディスコース･プランからドメイン･プランへ直接パスが張られるもののうち、話題の変化を示すクルー･ワードがある

場合のドメイン･プランの探索方法を全数探索で行なっていたのでは時間がかかる。

⑥ ドメイン･プラン外の発話の扱い。
登録用紙を得るためのドメイン･プランとして、

Header : Get-The-Form(?sp1 ?sp2 登録用紙)
Prereq : ((Predicate KNOW)(Agent ?sp1)(Object ?Address(?sp2))
Pred&cases : (Pred 送る)(Agent ?sp1)(Recip ?sp2)(Obje 登録用紙))

しかシステムに与えられていない場合に

A1: 請求書を送って頂きたいのですが。
B1: わかりました。
B2: 送り先をお願いします。
A2: 大阪市東区ATRです。

のような対話がなされると、システムは解釈することができない。このような場合を扱うためにドメイン･プランの階層化を行なうことを検討している。たとえば、一般的にある物を送るSend-Somethingというドメイン･プランを設定し、登録用紙を得るためのドメイン･プランはSend-Somethingサブ･ゴールとして持つようにする。Send-SomethingのPREREQUISITEには送り先の住所･氏名を知っていることが必要であると記述しておく。

Header : Get-The-Form(?sp1 ?sp2 登録用紙)
Decomp : Send-Something(?sp2 ?sp1 登録用紙)

Header : Send-Something(?sp1 ?sp2 ?object)
Prereq : ((Predicate KNOW)(Agent ?sp1)(Object ?Address(?sp2))
Pred&cases : (Pred 送る)(Agent ?sp1)(Recip ?sp2)(Obje ?object))
---

⑦ 異なる言語表現の同義解釈の問題
対話においては同じものを指す場合でも違った表層表現となる。たとえば、「登録用紙」は、「参加(の)申込み用紙」、「参加(の)申込み書」、「登録の用紙」、「登録(の)申込み書」などの表層表現として同一の対話の中で出現する[Nogaito 87b]。これらが同じものを指すことが分からなければならない。この問題についてはATRで研究が開始されている[Nogaito 88]。

⑧ 発話解析モジュールとの結合
単一化文法に基づく発話解析のモジュールと結合して、対話翻訳システムに発話理解モジュールを組み入れることは、文脈を考慮しながら翻訳をする一つの手法となる。発話の解析は、言語レベルに閉じた可能な解釈を与えるが、実世界との対応は取れていない。この対応付けを実現することにより、理解モジュールの結合が可能となる。

謝辞

本研究の機会を与えてくださるとともに適切な助言を述べられたATR自動翻訳電話研究所 樽松　明 社長、同 言語処理研究室 相沢輝昭 室長に感謝します。また熱心に討論して下さった言語処理研究室の諸氏に感謝します。

参考文献

[Allen and Perrault 80] J.F.Allen and C.R.Perrault:"Analyzing Intention in Utterances", ARTIFICIAL INTELLIGENCE, Vol.15, p143-178(1980)

[Arita 87] 有田、小暮、野垣内、前田、飯田:「メディアに依存する会話の様式ー電話会話とキーボード会話の比較ー」情報処理学会自然言語処理研究会61-5 (1987)

[Arita and Iida 87] 有田、飯田:「日本語におけるタスクオリエンティッドな対話の構造」電子情報通信学会言語処理とコミュニケーション研究会資料 NLC87-10(1987)

[Arita and Iida 88a] 有田、飯田:「目標指向型対話における次発話の予測」情報処理学会第37回全国大会(1988)

[Arita and Iida 88b] 有田、飯田:「目標指向型対話における次発話の予測」ATRテクニカルレポートTR-I-0042 (1988)

[Cohen84] P.R.Cohen, "The Pragmatics of Referring and the Modality of Communication", Computational Linguistics,Vol. 10, Num. 2, p97-p146(1984)

[Cohen86] P.R.Cohen and S.Fertig, "Discourse Structure and the Modality of Communication", International Symposium on Prospects and Problems of Interpreting Telephony(1986)

[Grosz 85] B.Grosz and C.L.Sidner, "The Structure of Discourse Structure", CSLI Report No.CSLI-85-39(1985)

[Grosz 86] B.Grosz and C.L.Sidner, "Attention, intentions, and structure of discourse", Computational Linguistics, 12(3), pp175-204

[Iida 86] 飯田、野垣内、相沢:「通訳を介した電話会話の特徴分析」電子情報通信学会技術研究報告NLC86-11(1986)

[Iida 87] 飯田、小暮、前田:「端末間の対話通訳システム」情報処理学会自然言語処理研究会64-10 (1987)

[Iida 88] 飯田等:「端末間対話通訳の実験システム構成」情報処理学会第36回全国大会(1988)

[Kaplan 83] S.J.Kaplan : "Cooperative Responses From a Portable Natural Language Database Query System", in COMPUTATIONAL MODELS OF DISCOURSE edited by M.Brady and R.C.Berwick, MIT Press, (1983)

[Katou 88] 加藤、中川：「自然言語インタフェースシステムにおける意図の把握と話題の管理」情報処理学会論文誌 Vol. 29 No. 9 (1988)

[Kitahashi 88] 北橋等：「対話構造のモデル化と知識利用」文部省科学研究費補助金重点領域研究「音声言語によるマン・マシンイターフェイスの高度化」研究報告No. PASL 63-6-4 (1988)

[Kogure 88] K.Kogure, H.Iida, and K.Yoshimoto : "A Method of Analyzing Japanese Speech Act Types", Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language (1988)

[Kume 88] 久米等:「日本語対話文における発話意図の解析の方法」情報処理学会第36回全国大会(1988)

[Litman and Allen 87] D.J.Litman and J.F.Allen:"A Plan Recognition Model for Subdialogues in Conversations", Cognitive Science, Vol.11,p163-200(1987)

[Nogaito 87a] 野垣内:「電話会話における指示・省略の特徴分析」電子情報通信学会全国大会(1987)

[Nogaito 87b] 野垣内、飯田:「照応・指示関係の同一性の分類・解析」人工知能学会研究会資料 SIG-FAI-8701-1(1987)

[Nogaito 88] 野垣内:「名詞句の同一性」ATRテクニカル・レポート(1988) *(to appear)*

[Ogura 88]小倉:「ATRモデル会話」ATR内部資料(1988)

[Pollack 86] M.E.Pollack : "A MODEL OF PLAN INFERENCE THAT DISTINGUISHES BETWEEN THE BELIEFS OF ACTORS AND OBSERVERS", Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics (1986)

[Sagawa 88] 佐川、杉原、杉江：「柔軟な対話制御機構を持ったコンサルテーション・システム」情報処理学会論文誌 Vol. 29 No. 4 (1988)

[Tomabechi 87] H.Tomabechi, "Direct Memory Access Translation", Proceedings of the Tenth International Joint Conference on Artificial Intelligence, p722(1987)

[Yamanashi 84] 山梨正明：「対話理解の基本的側面ー言語学からの方法論と問題点ー」、「対話行動の認知科学的研究」研究会資料(昭和59年2月)

[Yoshimoto 88] 吉本、小暮「日本語端末間対話解析のための句構造文法」情処学会第37回全国大会(1988)

付録

A1.「国際会議の問合せ」におけるドメイン·プランとオブジェクトの例

A2. 発話タイプ、スピーチ·アクト、インターラクション·プランの関係

A2.1 パーサの解析結果と発話タイプの対応

A2.2 スピーチ·アクトとインターラクション·プランの関係

A3. [Allen and Perrault 80] の plan inference rules と推論規則の対応

A4. [Litman and Allen 87] のdomain plan , discourse plan との関係

A5. プロトタイプ·システムの動作例

A5.1 動作例

A5.2 対話構造の例

A5.3 ドメイン·プラン、ディスコース·プラン、インターラクション·プランの例

A1. 「国際会議の問合せ」におけるドメイン･プランとオブジェクトの例

ATRモデル会話[Ogura 88]を処理するのに必要なドメイン知識を示す。見易さのためにカナ漢字表記する。

File-name: ln:~/Domain-Knowledge/domain-knowledge Ver 0.2
Dec. 1, 1988

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊
ＡＴＲモデル会話(A,B,1,2,3,4,5)を処理するのに必要なドメイン知識
＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

****
記述
****
header: アクションの見出し
preconditions: 前提条件
decomposition: アクション・プランへの分解
constraints: 制約条件
effects: アクションが成功した場合の効果

****
プランナ
****
プランナの動作
1．constraintsを満たすアクションを選び出す。
複数ある場合はconstraintsの数が多いものを優先する
2．effectsが成立しているかどうか調べる。
成立している場合は、アクションが成功したとする。
3．preconditionsが成立しているかどうか調べる。
成立している場合はdecompositionをサブ・ゴールとする。
成立していない場合は成立することをサブ・ゴールとする。
4．decompositionsがすべて成功すると。effectsを実行する。
****
ドメイン知識
****
注）X-n,m はその知識が利用されているテキストの場所を示す。
Xは会話番号、n,mはターンの番号（はじめから1,2,3,...とつける。）

==========
A-3 1-3
header: 質問者が会議に参加する
preconditions:
decomposition: 質問者が会議に申し込む
effects:
==========
A-3,4,5 B-3,4 1-3,4,5

header:　　質問者が会議に申し込む
preconditions:
decomposition:　質問者が登録用紙で手続きをする
　　質問者が参加料を支払う
effects:

B-9,10 2-6,7
header:　　質問者xが参加料sankaryouを支払う
preconditions:　質問者xが参加料sankaryouが要ることを知っている
　　質問者xが参加料sankaryouの額kingakuを知っている
　　質問者xが参加料sankaryouの支払い方法houhouを知っている
　　質問者xが参加料sankaryouの支払い期限kigenを知っている
decomposition:　質問者xが参加料sankaryouを支払い方法houhouで支払う
effects:

2-7
header:　　質問者xが参加料sankaryouを支払い方法GINKOU-HIRIKOMIで支払う
preconditions:　質問者xが口座番号bangouを知っている。
---

header:　　質問者が登録用紙で手続きをする
preconditions:
decomposition:　事務局が質問者に登録用紙を送る
　　質問者が登録用紙に記入する
　　質問者が事務局に登録用紙を送る
effects:
==========

A-4,6,7 B-5,6,7 1-6,7
header:　　事務局が質問者に登録用紙を送る
preconditions:　事務局が質問者の住所と名前を知っている
　　事務局が質問者の電話番号を知っている(optional)
decomposition:
effects:　　質問者が登録用紙を持っている
（注）このeffectsは物理的に登録用紙を持たなくてもよい。
==========

3-2
header:　　質問者が論文を発表する
preconditions:　質問者が会議の内容を知っている（論文の内容が会議の内容にふさわしいかどうか知っている）
decomposition:
effects:
==========

4-3
header:　　事務局が質問者に会議について説明する。
preconditions: 事務局が質問者が案内書を持っているかどうか知っている
decomposition: 事務局が質問者に会議について説明する-2。
constraints:
effects:

4-3
header:　　事務局が質問者に会議について説明する-2。
preconditions:
decomposition: 事務局が案内書に従って説明する
constraints:　質問者が案内書を持っている
effects:

4-3
header:　　事務局が質問者に会議について説明する-2。
preconditions:
decomposition: 事務局が案内書なしで説明する
　　事務局が案内書を質問者に送る
constraints:　質問者が案内書を持っていない
effects:
==========
header:　　事務局yが、会議confに対するxの参加取り消し手続きを行なう
preconditions: 事務局yが、xの名前を知る
decompositions:yが、登録料を払い戻す
constraints:　会議confは、参加料を払い戻すことのできる会議である
---
header:　　事務局yが、会議confに対するxの参加取り消し手続きを行なう
preconditions: 事務局yが、xの名前を知る
decompositions:事務局yが、会議confのプログラムと会議confの予稿集をxに送る
constraints:　会議confは、参加料を払い戻すことのできない会議である
---
header:　　事務局yが、会議confに対するxの参加取り消し手続きを行なう
preconditions: 事務局yが、xの名前を知る
decompositions:事務局yが、xの代理人zの名前を知る
constraints:　会議confは、参加料を払い戻すことのできない会議である
　　会議confは、代理人zの参加を認める会議である
==========

ドメインに依存しない一般的な知識
==========
header:　　xがオブジェクトonjectについて知る

decompositions:xがobjectのプロパテイprop-iについて知る
==========
4-3
header:　　xがアクションactionを実行する
decomposition: xがactionの制約条件constraintが成立していることを知る
==========

****
その他の知識
****
==========
ある時期以降のキャンセルには参加費の払い戻しはできない。(5-7)
参加費は申し込み時期によって割引がある場合がある。(2-2)
参加費は資格によって割引がある場合がある。(2-4,5)
会議confの公式言語がlangである場合、
　　langと他の言語lang2の同時通訳が提供されることがある。　(3-5,6,7)
==========

****

<オブジェクトのプロパテイとその値>

(見易さのためにカナ漢字表記する)

****

File Name = atr-ln:/usr3/arita/dis-struct/object

(注)@ＸＸはＸＸと同じであることを示す

オブジェクト (ー>スーパー) (=同義語)	プロパテイ	バリュー
	期限	[日付]、[日付]まで
銀行振込	口座番号	[数字] 2-7
登録費	通貨	円、ドル
	金額	[数字]円、[数字]ドル
	支払い方法	現金、カード
	支払い期限	@期限
	割引	[数字」%、[数字]円、[数字]ドル
	含むもの	予稿集代と歓迎会費用 2-3
支払い	方法	現金、カード
	期限	@期限
発表	言語	日本語、英語
	機器	ＯＨＰ、スライド
	申し込み	
	申し込み用紙	ある、ない
アブストラクト (書類)	言語	@言語
	字数	
	長さ	
書類	締切	@期限
	送り先	[住所]
会議	公式言語	@言語
	同時通訳	ある、ない
	予稿集	
	論文集	
	登録料	
	プログラム	
	場所	
	主催	

	目的 期日	
会議場	場所	
スライド	申し込み用紙	ある、ない
ホテル	宿泊費 手配 リスト 名前	@金額 する、しない
論文集	費用	@金額
分科会	数	[数字]
ツアー	参加者	[人数]
クレジットカード	名称 番号	
スピーカ	受付 申し込み リスト	
発表者 (=スピーカ)		

A2. 発話タイプ、スピーチ･アクト、インターラクション･プランの関係

A2.1 パーサの解析結果と発話タイプの対応

ATRで開発を進めているパーサ[Kogure 88, Yoshimoto 88]の解析結果と発話タイプとの対応の例を以下に示す。現在パーサは開発段階にあり、その解析結果も変更が予想されるため体系的な対応については十分に検討されていない。現時点では下表のパーサの解析結果のアンダーラインの情報に注目して対応する発話タイプを求めることができると考えている。もちろんパーサの解析結果と発話タイプは1対1には対応しない。

下表の発話タイプの列の表現は本対話理解システムの入力となるもので、各引数の意味は次のとおりである。

(発話タイプ, 発話者, 聞き手, クルー･ワード, トピック, 述語とその格要素)

パーサの解析結果	発話タイプ
「もしもし」 [sem[[reln もしもし-hello] [agen .....]	(Greeting-Open .....)
「そちらは国際会議事務局ですか」 [head [[topic *hearer*]] [sem [[reln s-request] [agen *speaker*] [recp *hearer*] [obje [reln informif] [agen *hearer*] [recp *speaker*] [obje [[reln だ-identical] [obje *hearer*] [iden 国際会議事務局]]]	(Confirm-Value, *hearer*,*speaker*,nil,*hearer*, ((predicate です)(obje *hearer*) (iden 国際会議事務局)))
「はい」 [sem [[reln はい-affirmative] 「そうです」 [sem [[reln だ-statement] [obje そう-1]]	(Affirmative .....)
「わたしは会議に申込みたいのですが」 [head [[topic *speaker*]] [sem [[reln が-moderate] [obje [[reln のだ-explicative] [obje [reln たい-desire]	(Introduce-object,sp,hr,clue, 会議に申し込む, ...) この場合のtopicは解析結果のtopic格では

[obje [reln 申込む-1] [sloc 会議-1]]]]	なく、たい-desireのobje格にする。
「登録用紙は既にお持ちでしょうか」 [head [[topic 登録用紙-1]] [sem [[reln s-request] [agen *speaker*] [recp *hearer*] [obje [reln informif] [recp *speaker*] [obje [[reln よう-guess] [obje [reln 持つ-1] [obje 登録用紙-1]]]	(Ask-truth, *hearer*,*speaker*,nil,登録用紙-1, ((predicate持つ) (obje 登録用紙-1)))
「いいえ」 [sem [reln いいえ-negative]] 「まだです」 [sem [reln だ-statement] [obje まだ-1]]	(Negative-Truth, ...) 「まだです」についてはpending
「わかりました」 [sem [[reln perfective] [obje [[reln 分かった-I_see]]	(Accept, ...) (Confirm, ...)
「それでは、こちらからそちらに 登録用紙をお送り致します」 [head [ctype masu]] [sem [[reln 送る-1] [agen *speaker*] [recp *hearer*] [obje 登録用紙-1]]]	(Will-Do-Action, ...)

A2.2 スピーチ･アクトとインターラクション･プランの関係

スピーチアクトを次のように定義する。

```
*  スピーチアクト
-----------------------------------------
REQUSET(sp, hr, action)
precondition: WANT(sp, action)
effect:       WANT(hr,action)
-----------------------------------------
INFORMIF(sp, hr, proposition)
precondition: KNOWIF(sp,proposition)
              WANT(sp,INFORMIF(sp,hr,proposition))
effect:       KNOWIF(hr,proposition)
-----------------------------------------
```

このスピーチアクトを用いて、
「Objctのpropertyはvalueですか？」(a)
「はいそうです。」　　(b)
を解析すると次のようになる。
発話(a)は構文・意味解析されて、
　(2) REQUEST(sp,hr,INFORMIF(hr,sp,EQUAL(property(object),value)))
となる。これはスピーチアクトREQUEST(sp,hr,action)とマッチし話者spのプランがそのpreconditionから、
　KNOWIF(sp,EQUAL(property(object),value))
と、認識される。
また、REQUESTのeffectから話者hrは
　(4) INFORMIF(hr,sp,EQUAL(property(object),value))
というプランを持つ。
発話(b)が
　INFORMIF(hr,sp,_)

と、解析できると(4)とマッチし、そのeffectから
(5)KNOWIF(sp,EQUAL(property(object),value))
となり、話者spのはじめのプランが達成される。

```
(1)WANT(sp,KNOWIF(sp,EQUAL(property(object),value)))      ;; goal of sp
  |
  V precondition -> action
(2)REQUEST(sp,hr,INFORMIF(hr,sp,EQUAL(property(object),value)))
  | ||
  | ||
  | ++================発話計画=(condition:?a?)=>Objctのpropertyはvalueですか？
  |                            (condition:?b?)=>Objctのpropertyはvalueですね？
  |
  |.............> ?turn-taking?
  |
  V action -> effect
(3)WANT(hr,INFORMIF(hr,sp,EQUAL(property(object),value))) ;; goal of hr
  | precondition:KNOWIF(hr,EQUAL(property(object),value))
  |
  V precondition -> action
(4)INFORMIF(hr,sp,EQUAL(property(object),value)))
  | ||
  | ++================発話計画=(condition:?c?)=>Objctのpropertyはvalueです。
  |                            (condition:?turn-taking?
  |                             constraint:
  |                              EQUAL(property(onject),value))=>はいそうです。
  |
  V action -> effect
(5)KNOWIF(sp,EQUAL(property(object),value))     ;; completion of sp's goal
```

ここで、次のような問題点がある。
(*1)発話(b)「はいそうです」が
　　INFORNIF(hr,sp,_)
　と解析することができるか？
(*2)逆に
　(4)INFORMIF(hr,sp,EQUAL(property(object),value)))
　から発話をしようとする場合、
　　「Objctのpropertyはvalueです。」 (c)
　ではなく、
　　「はいそうです」(b)
　とすることができるか？

(*2)についてはspがREQUESTをした時点でなんらかの文脈的effectができて（それを?turn-taking?とする）そのconditionによって発話(c)でなく発話(b)が選択されると考えることができる。

(*1),(*2)に対処するために、以下のようなインターラクションプラン、発話タイプを導入する。

インターラクションプラン

```
-----------------------------------------------------
CONFIRM-VALUE-PLAN(sp,hr,EQUAL(property(object),value))
precondition: WANT(sp,KNOWIF(sp,EQUAL(property(object),value)))
              KNOWIF(hr,EQUAL(property(object),value))
constraint:   EQUAL(property(object),value)
effect:       KNOWIF(sp,EQUAL(property(object),value))
decomposition: CONFIRM-VALUE(sp,hr,EQUAL(property(object),value))
               AFFIRMATIVE(hr,sp,EQUAL(property(object),value))
-----------------------------------------------------
```

発話タイプ

```
-----------------------------------------------------
CONFIRM-VALUE(sp,hr,EQUAL(property(object),value))
[precondition:  WANT(sp,KNOWIF(sp,EQUAL(property(object),value)))      ]
[               KNOWIF(hr,EQUAL(property(object),value))               ]
decomposition: REQUEST(sp,hr,INFORMIF(hr,sp,EQUAL(property(object),value)))
[effect:        WANT(hr,INFORMIF(hr,sp,EQUAL(property(object),value))) ]
-----------------------------------------------------
AFFIRMATIVE(sp,hr,EQUAL(property(object),value))
[precondition:  WANT(sp,INFORMIF(sp,hr,EQUAL(property(object),value))) ]
[constraint:    EQUAL(property(object),value) is true                  ]
decomposition: INFORM(sp,hr,Da((obje Sou)))
[effect:        KNOWIF(hr,EQUAL(property(object),value))               ]
-----------------------------------------------------
```

?turn-taking?に相当する文脈的effectはインターラクションプランのdecompositionで表わされる。またインターラクションプランのdecompositionを発話の表層表現に近い発話タイプで表わすことにより、スピーチアクトの解析を省略できる。発話タイプのprecondition,constraint,effectはインターラクションプランで記述される。また

```
  CONFIRM-VALUE                                          AFFIRMATIVE
          effect --> WANT(sp,INFORMIF(sp,hr,          --> precondition
                          EQUAL(property(object),value))
```

の推論過程が省略できる。

A3. [Allen and Perrault 80] の plan inference rules と推論規則の対応

[Allen and Perrault 80]	T-Plan Model
[Precondition-Action Rule] SBAW(P) = *i*⇒SBAW(ACT) if P is a precondition of action ACT.	PREREQUISITE → HEADER
[Body-Action Rule] SBAW(B) = *i*⇒SBAW(ACT) if B is part of the body of ACT.	DECOMPOSITION → HEADER
[Action-Effect Rule] SBAW(ACT) = *i*⇒SBAW(E) if E is an effect of ACT.	
[Want-Action Rule] SBAW(*n*W(ACT)) = *i*⇒SBAW(ACT) if *n* is the agent of the intentional action ACT.	
[Know-positive Rule] SBAW(A KNOWIF *P*) = *i*⇒SBAW(*P*)	(Ask-Acceptability1 → Ask-Action) (Ask-Acceptability2 → Request-Action) (Confirm-Action → Will-Do-Action2)
[Know-negative Rule] SBAW(A KNOWIF *P*) = *i*⇒SBAW(not*P*)	否定については考慮していない。
[Know-value Rule] SBAW(A KNOWIF *P*(*a*)) = *i*⇒ SBAW(A KNOWREF the *x* : *P*(*x*))	Confirm-Value → Ask-Value (Ask-Exist-Value → Ask-Value)
[Know-term Rule] SBAW(A KNOWREF the *x* : *D*(*x*)) = *i*⇒SBAW(*P*(the *x* : *D*(*x*)) where *P*(the *x* : *D*(*x*) is a goal or action involving the description (or its referent)	Objectからプランへの 連想テーブル
	PREDICATE&CASES → HEADER
(plan construction rule)	EFFECTS → HEADER

注) The notation SBAW(*X*) = *i*⇒SBAW(*Y*) indicates that if S believes A's plan contains *X*, then S may infer that A's plan also contains *Y*.

A4. [Litman and Allen 87] のdomain plan , discourse plan との関係

[Litman and Allen 87]	[T-Plan Model]
[入力] "The eight-fifty to Montreal ?" SURFACE-REQUEST(Person1, Clerk1, INFORMREF(Clerk1, Person1,?term, Gate(dtrain,eight-fifty to Montreal)	[入力] "The eight-fifty to Montreal ?" Ask-Value(Person1, Clerk1,nil,Gate, (pred&cases (predicate desu) (object Gate) (identifier ?term))
[ディスコース･プラン] Introduce-Plan	[ディスコース･プラン] Introduce-Domain-Plan-Node
[ディスコース･プラン] Correct-Plan	対応するものなし
[ディスコース･プラン] Identify-Parameter	[インターラクション･プラン] Get-Value-Unit
対応するものなし	[インターラクション･プラン] Turn-Takingの起動に関するもの
対応するものなし	[インターラクション･プラン] 対話のチャンネルに関するもの
[ドメインプラン] ドメインプランは[T-Plan Model]と同様である。	[ドメインプラン] ドメインプランは[Litman and Allen 87]と同様である。

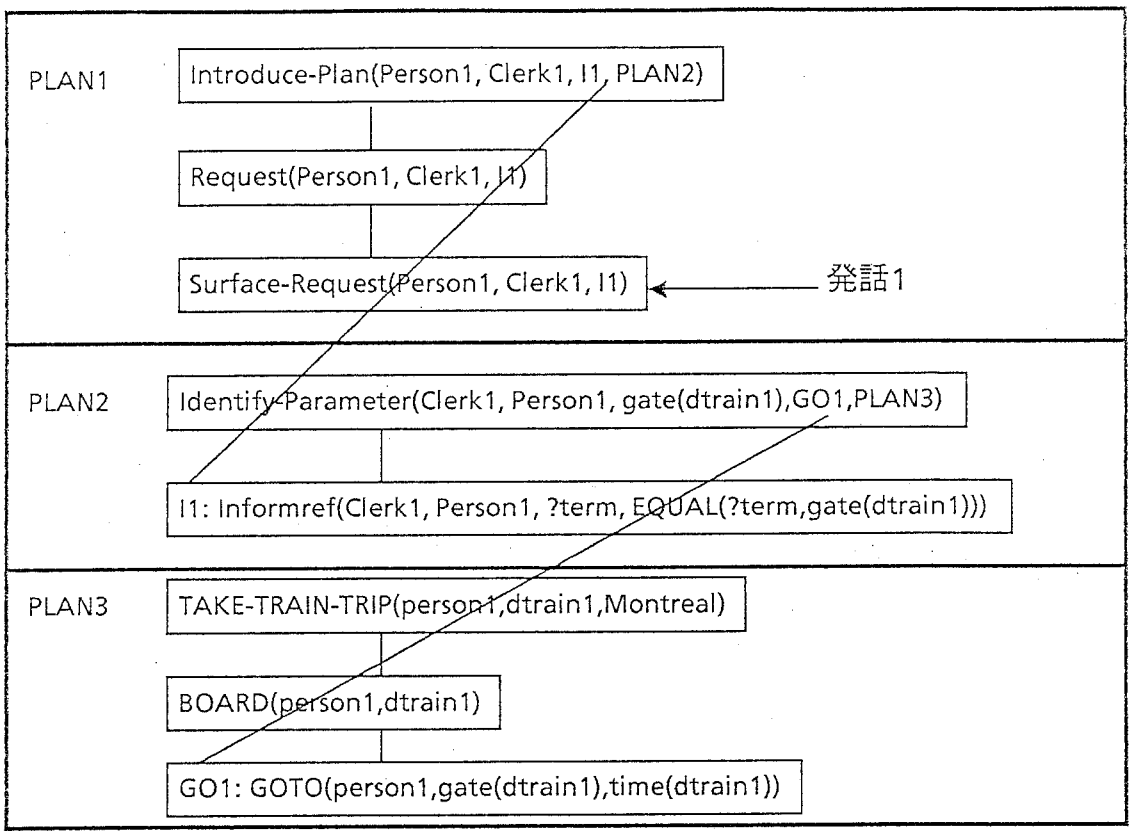
T-Plan Model はsub-dialogueだけでなく、対話全体の解釈を目指しているため、ディスコース･プラン、インターラクション･プランの分類がより詳細になっている。

次の対話の処理を比較する。

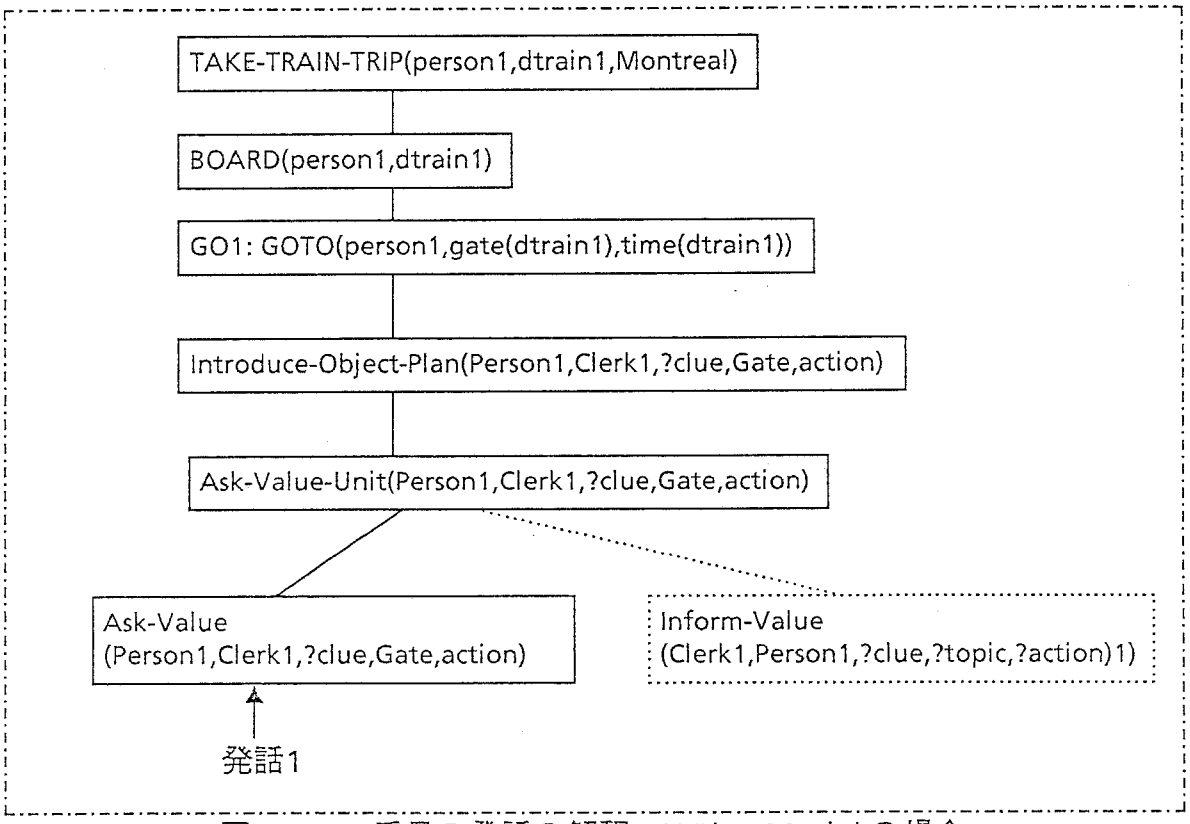
Passenger: The eight-fifty to Montreal ?
Clerk: Gate seven.

(a)1番目の発話の解釈
Litman and Allen の場合を図A4-1に示す
T-Plan Model の場合を図A4-2に示す
(b)2番目の発話の解釈
Litman and Allen の場合を図A4-3に示す
T-Plan Model の場合を図A4-4に示す

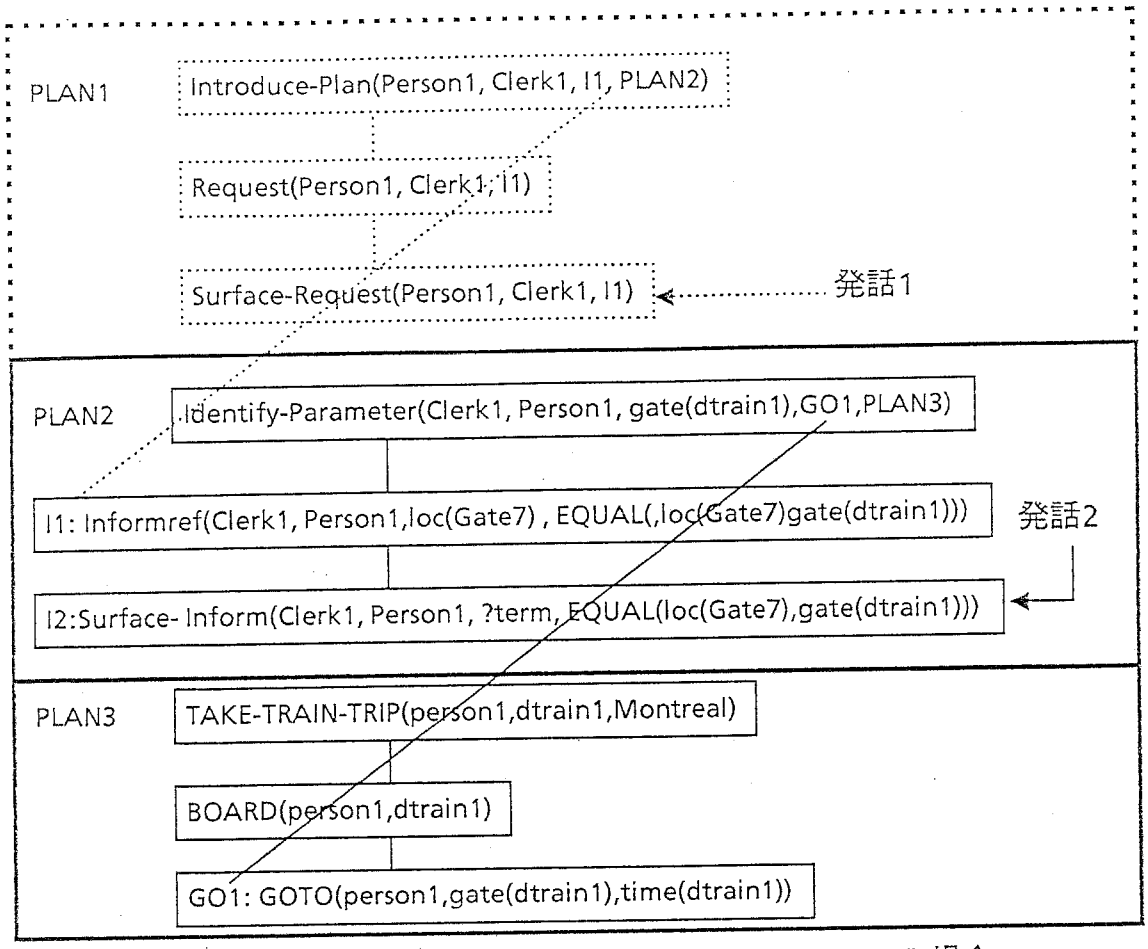
T-Plan Model のほうが2つの発話の関係をよく表わしている。



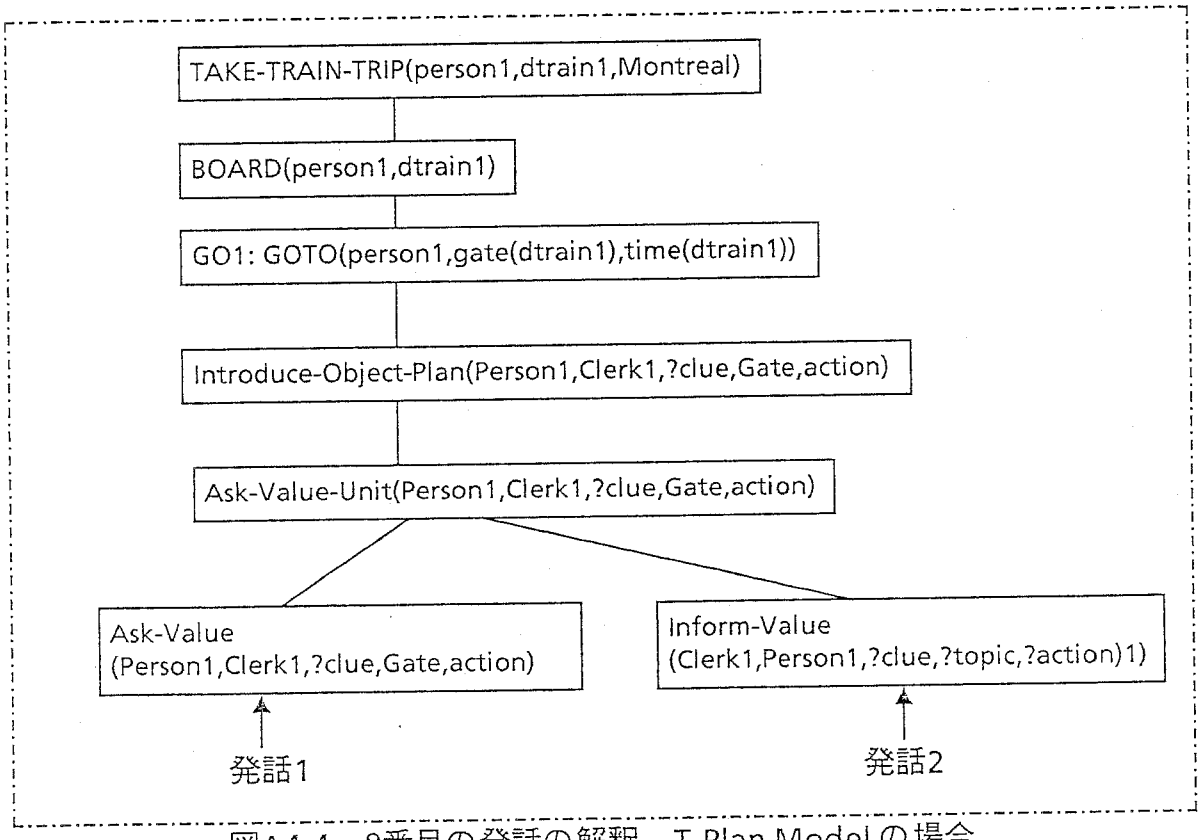
図A4-1　1番目の発話の解釈　Litman and Allen の場合



図A4-2　1番目の発話の解釈　T-Plan Model の場合



図A4-3　2番目の発話の解釈　Litman and Allen の場合



図A4-4　2番目の発話の解釈　T-Plan Model の場合

A5. プロト･タイプシステムの動作例

本報告で述べた対話理解モデルは、リスプマシンSymbolics 3620上に一部の機能を除いて実現されている。
図A5-1に動作例の画面のハードコピーを示す。
コマンドの意味は次のとおりである。

発話の解釈：	対話の次の発話の解釈をする。
次発話の予測：	現時点での文脈から予測できる次の発話を求める。
現在の理解状態の表示：	現時点での対話の理解状態を表示する、
プランのパスの表示：	直前に処理した発話からゴールのプランまでのパスを表示する。
ゴールの表示：	現時点での対話のゴールを表示する。
対話構造の表示：	現時点での対話の構造を表示する。
内部表現の表示：	システムの内部表現をそのまま表示する。
対話の読み込み：	処理の対象とする対話をファイルから読み込む。
対話の表示：	処理の対象とする対話を表示する。
プランの読み込み：	処理に利用するプランをファイルから読み込む。
プランの表示：	処理に利用するを表示する。
初期化：	初期化を行なう。
終了：	処理を終了してLISPのトップレベルに戻る。

図A5-1　動作例の画面のハードコピー

A5.1　動作例

(run)

******< Thu Dec 1 0:53:3 1988 >******

初期化して下さい。

[初期化します]... [初期化しました]

[対話の表示]

対話番号 : 1

(1)sp1: もしもし。
(2)sp1: 通訳電話国際会議事務局ですか。
(3)sp2: はい、そうです。
(4)sp2: どのようなご用件でしょうか。
(5)sp1: 会議に申込たいのですが、
(6)sp1: どのような手続きをすればよろしいのでしょうか
(7)sp2: 所定の登録用紙で手続きをして下さい。
(8)sp2: 登録用紙はすでにお持ちでしょうか。
(9)sp1: いいえ、まだです。
(10)sp2: 分かりました。
(11)sp2: それでは、登録用紙をお送り致します。
(12)sp2: ご住所とお名前をお願いします。
(13)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。
(14)sp2: 分かりました。
(15)sp2: 登録用紙を至急送らせて頂きます。
(16)sp1: よろしくお願いします。

(17)sp1: それでは失礼します。

[対話の表示終了]

[発話の解釈]

入力発話 : (1)sp1: もしもし。

解析結果(1/1): (GREETING-OPEN SP1 SP2 ?CLUE2-1 ?TOPIC2-1 (PRED&CASES (PREDICATE MOSHIMOSHI) (AGENT SP1)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (GREETING-OPEN SP1 SP2 ?CLUE0-1 ?TOPIC0-1 (PRED&CASES (PREDICATE MOSHI MOSHI) (AGENT SP1)))

[発話の解釈 終了]

[発話の解釈]

入力発話 : (2)sp1: 通訳電話国際会議事務局ですか。

解析結果(1/2): (CONFIRM-VALUE-A SP1 SP2 ?CLUE2-2 ?TOPIC2-2
(PRED&CASES (PREDICATE DESU) (OBJECT SP2) (IDENTIFIER KOKUSAIKAIGI_JIMUKYOKU)))

解析結果(2/2): (CONFIRM-VALUE-N SP1 SP2 ?CLUE2-2 ?TOPIC2-2
(PRED&CASES (PREDICATE DESU) (OBJECT SP2) (IDENTIFIER KOKUSAIKAIGI_JIMUKYOKU)))

```
 現在のゴールリストで解釈を試みます。
 処理対象 : 解析結果(1/2)
 処理対象 : 理解状態(1/1)
 Found the path (find-plan-path)
 > 解釈 : (CONFIRM-VALUE-A SP1 SP2 ?CLUE1$2 ?TOPIC1$2
          (PRED&CASES (PREDICATE DESU) (OBJECT SP2) (IDENTIFIER KOKUSAIKAIGI_JIM
UKYOKU)))
 処理対象 : 解析結果(2/2)
 処理対象 : 理解状態(1/1)

 [ 発話の解釈 終了]

 [ 発話の解釈 ]

 入力発話 : (3)sp2: はい、そうです。

 解析結果(1/1): (AFFIRMATIVE SP2 SP1 ?CLUE2-3 ?TOPIC2-3
                 (PRED&CASES (PREDICATE DESU) (OBJECT ?OBJ$1-2) (IDENTIFIER ?ID$1
-2)))

 現在のゴールリストで解釈を試みます。
 処理対象 : 解析結果(1/1)
 処理対象 : 理解状態(1/1)
 Found the path (find-plan-path)
 > 解釈 : (AFFIRMATIVE SP2 SP1 ?CLUE1$2 ?TOPIC1$2
          (PRED&CASES (PREDICATE DESU) (OBJECT SP2) (IDENTIFIER KOKUSAIKAIGI_JIM
UKYOKU)))

 [ 発話の解釈 終了]

 [ 発話の解釈 ]
```

```
入力発話 : (4)sp2: どのようなご用件でしょうか。

解析結果(1/1): (ASK-TOPIC SP2 SP1 ?CLUE2-4 ?TOPIC2-4
                (PRED&CASES (PREDICATE DESU) (OBJECT ?OBJECT2-4) (IDENTIFIER ?ID
2-4)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (ASK-TOPIC SP2 SP1 ?CLUE0-2 ?TOPIC0-2
          (PRED&CASES (PREDICATE DESU) (OBJECT ?OBJECT$22) (IDENTIFIER ?ID$22)))

[ 発話の解釈 終了]

[ 発話の解釈 ]

入力発話 : (5)sp1: 会議に申込たいのですが、

解析結果(1/1): (INTRODUCE-OBJECT SP1 SP2 ?CLUE2-5 KAIGI_MOUSHIKOMI
                (PRED&CASES (PREDICATE OSHIERU) (AGENT SP2) (RECIPIENT SP1) (OBJ
ECT KAIGI_MOUSHIKOMI)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (INTRODUCE-OBJECT SP1 SP2 ?CLUE0-2 KAIGI_MOUSHIKOMI
          (PRED&CASES (PREDICATE OSHIERU) (AGENT SP2) (RECIPIENT SP1) (OBJECT KA
IGI_MOUSHIKOMI)))

[ 発話の解釈 終了]
```

[発話の解釈]

入力発話 : (6)sp1: どのような手続きをすればよろしいのでしょうか

解析結果(1/1):
(ASK-ACTION SP1 SP2 ?CLUE2-6 ?TOPIC2-6 (PRED&CASES (PREDICATE ?PRED2-6) (AGENT S
P1) (OBJECT ?OBJECT2-6)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)

すでに参照されたのゴールリストで解釈を試みます。
解析結果(1/1):

発話タイプから連想されるゴールで解釈を試みます。

解析結果(1/1):

ASK-ACTION-PLAN をゴールとして解釈を試みます。(utt-type)
理解状態(1/1):
Found the path (find-plan-path)
> 解釈 : (ASK-ACTION SP1 SP2 ?CLUE$35 ?TOPIC$35 (PRED&CASES (PREDICATE MOUSHIKO
MU) (AGENT SP1) (OBJECT KAIGI)))

[発話の解釈 終了]

[発話の解釈]

入力発話 : (7)sp2: 所定の登録用紙で手続きをして下さい。

```
 解析結果(1/1): (DIRECTION SP2 SP1 ?CLUE2-7 ?TOPIC2-7
                (PRED&CASES (PREDICATE TETSUDUKIWOSURU) (AGENT SP1) (OBJECT ?OBJ
ECT2-7)))

 現在のゴールリストで解釈を試みます。
 処理対象 : 解析結果(1/1)
 処理対象 : 理解状態(1/1)
 Found the path (find-plan-path)
 > 解釈 : (DIRECTION SP2 SP1 ?CLUE$35 ?TOPIC$35
          (PRED&CASES (PREDICATE TETSUDUKIWOSURU) (AGENT SP1) (OBJECT ?OBJECT2-7
)))

 [ 発話の解釈 終了]

 [ 発話の解釈 ]

 入力発話 : (8)sp2: 登録用紙はすでにお持ちでしょうか。

 解析結果(1/1): (ASK-TRUTH-N SP2 SP1 ?CLUE2-8 FORM (PRED&CASES (PREDICATE MOTSU)
 (AGENT SP1) (OBJECT FORM)))

 現在のゴールリストで解釈を試みます。
 処理対象 : 解析結果(1/1)
 処理対象 : 理解状態(1/1)
 Found the path (find-plan-path)
 > 解釈 : (ASK-TRUTH-N SP2 SP1 ?CLUE$41 FORM (PRED&CASES (PREDICATE MOTSU) (AGEN
T SP1) (OBJECT FORM)))

 [ 発話の解釈 終了]

 [ 発話の解釈 ]
```

```
入力発話 : (9)sp1: いいえ、まだです。

解析結果(1/1): (NEGATIVE-TRUTH SP1 SP2 ?CLUE2-9 ?TOPIC2-9 (PRED&CASES (PREDICAT
E DESU)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (NEGATIVE-TRUTH SP1 SP2 ?CLUE$41 FORM (PRED&CASES (PREDICATE DESU)))

[ 発話の解釈 終了]

[ 発話の解釈 ]

入力発話 : (10)sp2: 分かりました。

解析結果(1/2): (ACCEPT SP2
                       SP1
                       ?CLUE2-10
                       ?TOPIC2-10
                       (PRED&CASES (PREDICATE WAKARU) (AGENT SP2) (RECIPIENT ?RE
CIP2-10) (OBJECT ?OBJ2-10)
                        (MANNER ?MANNER2-10B)))

解析結果(2/2): (CONFIRM SP2 SP1 ?CLUE2-10 ?TOPIC2-10
                (PRED&CASES (PREDICATE WAKARU) (AGENT SP2) (RECIPIENT ?RECIP2-10
) (OBJECT ?OBJ2-10)
                 (MANNER ?MANNER2-10B)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/2)
```

```
処理対象 : 理解状態(1/1)
処理対象 : 解析結果(2/2)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (CONFIRM SP2 SP1 ?CLUE$41 FORM
         (PRED&CASES (PREDICATE WAKARU) (AGENT SP2) (RECIPIENT ?RECIP2-10) (OBJ
ECT ?OBJ2-10) (MANNER ?MANNER2-10B)
          ))

[ 発話の解釈 終了]

[ 発話の解釈 ]

入力発話 : (11)sp2: それでは、登録用紙をお送り致します。

解析結果(1/1): (WILL-DO-ACTION-A SP2 SP1 ?CLUE2-11 FORM
                (PRED&CASES (PREDICATE OKURU) (AGENT SP2) (RECIPIENT SP1) (OBJEC
T FORM) (MANNER ?MANNER2-11)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (WILL-DO-ACTION-A SP2 SP1 ?CLUE$54 FORM
         (PRED&CASES (PREDICATE OKURU) (AGENT SP2) (RECIPIENT SP1) (OBJECT FORM
) (MANNER ?MANNER$56)))

[ 発話の解釈 終了]

[ 発話の解釈 ]

入力発話 : (12)sp2: ご住所とお名前をお願いします。
```

```
 解析結果(1/1): (ASK-VALUE SP2 SP1 ?CLUE2-12 NAME&ADDRESS
                 (PRED&CASES (PREDICATE DESU) (OBJECT NAME&ADDRESS) (IDENTIFIER ?
ID2-12)))

 現在のゴールリストで解釈を試みます。
 処理対象 : 解析結果(1/1)
 処理対象 : 理解状態(1/1)
 Found the path (find-plan-path)
 > 解釈 : (ASK-VALUE SP2 SP1 ?CLUE$76 NAME&ADDRESS
          (PRED&CASES (PREDICATE DESU) (OBJECT NAME&ADDRESS) (IDENTIFIER ?N&A$42
)))

 [ 発話の解釈 終了]

 [ 発話の解釈 ]

 入力発話 : (13)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。

 解析結果(1/1): (INFORM-VALUE SP1 SP2 ?CLUE2-13 ?TOPIC2-13
                 (PRED&CASES (PREDICATE DESU) (OBJECT ?OBJECT2-13) (IDENTIFIER SU
ZUKI_MAYUMI)))

 現在のゴールリストで解釈を試みます。
 処理対象 : 解析結果(1/1)
 処理対象 : 理解状態(1/1)
 Found the path (find-plan-path)
 > 解釈 : (INFORM-VALUE SP1 SP2 ?CLUE$76 NAME&ADDRESS
          (PRED&CASES (PREDICATE DESU) (OBJECT NAME&ADDRESS) (IDENTIFIER SUZUKI_
MAYUMI)))

 [ 発話の解釈 終了]
```

```
[ 発話の解釈 ]

入力発話 : (14)sp2: 分かりました。

解析結果(1/2): (ACCEPT SP2
                      SP1
                      ?CLUE2-14
                      ?TOPIC2-14
                      (PRED&CASES (PREDICATE WAKARU) (AGENT SP2) (RECIPIENT ?RE
CIP2-14) (OBJECT ?OBJ2-14)
                       (MANNER ?MANNER2-14B)))

解析結果(2/2): (CONFIRM SP2 SP1 ?CLUE2-14 ?TOPIC2-14
                (PRED&CASES (PREDICATE WAKARU) (AGENT SP2) (RECIPIENT ?RECIP2-14
) (OBJECT ?OBJ2-14)
                 (MANNER ?MANNER2-14B)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/2)
処理対象 : 理解状態(1/1)
処理対象 : 解析結果(2/2)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (CONFIRM SP2 SP1 ?CLUE$76 NAME&ADDRESS
          (PRED&CASES (PREDICATE WAKARU) (AGENT SP2) (RECIPIENT ?RECIP2-14) (OBJ
ECT ?OBJ2-14) (MANNER ?MANNER2-14B)
           ))

[ 発話の解釈 終了]

[ 発話の解釈 ]
```

```
入力発話 : (15)sp2: 登録用紙を至急送らせて頂きます。

解析結果(1/1): (WILL-DO-ACTION-A SP2 SP1 ?CLUE2-15 FORM
                (PRED&CASES (PREDICATE OKURU) (AGENT SP2) (RECIPIENT SP1) (OBJEC
T FORM) (MANNER ?MANNER2-15)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)

すでに参照されたのゴールリストで解釈を試みます。
解析結果(1/1):
理解状態(1/1):

(CONFIRM ?SP$75 ?HR$75 ?CLUE$75 ?TOPIC$75 ?ACTION$75) をゴールとして解釈を試み
ます。(old-goals)

(INFORM-VALUE ?HR$75 ?SP$75 ?CLUE$75 ?TOPIC$75
 (PRED&CASES (PREDICATE DESU) (OBJECT ?OBJECT$75) (IDENTIFIER ?ID$75))) をゴー
ルとして解釈を試みます。(old-goals)

(ACHIEVE-KNOW ?HR$42 NAME&ADDRESS ?N&A$42) をゴールとして解釈を試みます。(old-g
oals)

(GET-THE-FORM ?SP$42 ?HR$42 ?FORM$42) をゴールとして解釈を試みます。(old-goals)
Found the path (find-plan-path)
> 解釈 : (WILL-DO-ACTION-A SP2 SP1 ?CLUE$202 FORM
         (PRED&CASES (PREDICATE OKURU) (AGENT SP2) (RECIPIENT SP1) (OBJECT FORM
) (MANNER ?MANNER$204)))

[ 発話の解釈 終了]
```

```
[ 発話の解釈 ]

入力発話 : (16)sp1: よろしくお願いします。

解析結果(1/1): (ACCEPT-OFFER SP1 SP2 ?CLUE2-16 ?TOPIC2-16 (PRED&CASES (PREDICAT
E NEGAU) (AGENT SP1)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/1)
Found the path (find-plan-path)
> 解釈 : (ACCEPT-OFFER SP1 SP2 ?CLUE$202 FORM (PRED&CASES (PREDICATE NEGAU) (AG
ENT SP1)))
Found the path (find-plan-path)
> 解釈 : (ACCEPT-OFFER SP1 SP2 ?CLUE$54 FORM (PRED&CASES (PREDICATE NEGAU) (AGE
NT SP1)))

[ 発話の解釈 終了]

[ 発話の解釈 ]

入力発話 : (17)sp1: それでは失礼します。

解析結果(1/1): (GREETING-CLOSE SP1 SP2 ?CLUE2-17 ?TOPIC2-17 (PRED&CASES (PREDIC
ATE SHITSUREISURU) (AGENT SP1)))

現在のゴールリストで解釈を試みます。
処理対象 : 解析結果(1/1)
処理対象 : 理解状態(1/2)
Found the path (find-plan-path)
> 解釈 : (GREETING-CLOSE SP1 SP2 ?CLUE0-3 ?TOPIC0-3 (PRED&CASES (PREDICATE SHIT
```

```
SUREISURU) (AGENT SP1)))
 処理対象 : 理解状態(2/2)
 Found the path (find-plan-path)
 > 解釈 : (GREETING-CLOSE SP1 SP2 ?CLUE0-3 ?TOPIC0-3 (PRED&CASES (PREDICATE SHIT
SUREISURU) (AGENT SP1)))

 [ 発話の解釈 終了]

 [ 発話の解釈 ]
 interpret-planner0: 対話終了

 [ 発話の解釈 終了]

 [ 対話構造の表示 ]

 ----- 理解状態 [ 1/2 ] -----
+--TOP-OF-DISCOURSE-STRUCTURE
       |--OPEN-DIALOGUE
       |      |--GREETING-OPEN-UNIT
       |      |      +--(GREETING-OPEN (1)sp1: もしもし。)
       |      +--CONFIRM-PARTNER-UNIT
       |             |--(CONFIRM-VALUE-A (2)sp1: 通訳電話国際会議事務局ですか。)
       |             +--(AFFIRMATIVE (3)sp2: はい、そうです。)
       |--ESTABLISH-TOPIC-UNIT
       |      |--(ASK-TOPIC (4)sp2: どのようなご用件でしょうか。)
       |      +--MAKE-REGISTRATION
       |             |--INTRODUCE-OBJECT-PLAN
       |             |      +--(INTRODUCE-OBJECT (5)sp1: 会議に申込たいのですが、)
       |             |--ASK-ACTION-PLAN
       |             |      +--ASK-ACTION-UNIT
       |             |             |--(ASK-ACTION (6)sp1: どのような手続きをすればよ
ろしいのでしょうか)
```

```
      |           |           +--(DIRECTION (7)sp2: 所定の登録用紙で手続きをして
下さい。)
      |           |--GET-THE-FORM
      |           |     |--ASK-TRUTH-PLAN
      |           |     |     +--ASK-TRUTH-UNIT
      |           |     |           |--(ASK-TRUTH-N (8)sp2: 登録用紙はすでにお持
ちでしょうか。)
      |           |     |           |--(NEGATIVE-TRUTH (9)sp1: いいえ、まだです
。)
      |           |     |           +--(CONFIRM (10)sp2: 分かりました。)
      |           |     |--INTRODUCE-DOMAIN-PLAN-NODE
      |           |     |     +--WILL-DO-ACTION-UNIT
      |           |     |           |--(WILL-DO-ACTION-A (11)sp2: それでは、登録
用紙をお送り致します。)
      |           |     |           +--ACCEPT-OFFER
      |           |     |--ACHIEVE-KNOW
      |           |     |     +--GET-VALUE-UNIT
      |           |     |           |--(ASK-VALUE (12)sp2: ご住所とお名前をお願
いします。)
      |           |     |           |--(INFORM-VALUE (13)sp1: 大阪市北区城見2-1-
61, 鈴木真弓です。)
      |           |     |           +--(CONFIRM (14)sp2: 分かりました。)
      |           |     +--INTRODUCE-DOMAIN-PLAN-NODE
      |           |           +--WILL-DO-ACTION-UNIT
      |           |                 |--(WILL-DO-ACTION-A (15)sp2: 登録用紙を至急
送らせて頂きます。)
      |           |                 +--(ACCEPT-OFFER (16)sp1: よろしくお願いしま
す。)
      |           |--FILL-THE-FORM
      |           +--RETURN-THE-FORM
      +--CLOSE-DIALOGUE
            +--GREETING-CLOSE-UNIT
```

```
                |--(GREETING-CLOSE (17)sp1: それでは失礼します。)
                +--GREETING-CLOSE

 ----- 理解状態 [ 2/2 ] -----
+--TOP-OF-DISCOURSE-STRUCTURE
      |--OPEN-DIALOGUE
      |     |--GREETING-OPEN-UNIT
      |     |     +--(GREETING-OPEN (1)sp1: もしもし。)
      |     +--CONFIRM-PARTNER-UNIT
      |           |--(CONFIRM-VALUE-A (2)sp1: 通訳電話国際会議事務局ですか。)
      |           +--(AFFIRMATIVE (3)sp2: はい、そうです。)
      |--ESTABLISH-TOPIC-UNIT
      |     |--(ASK-TOPIC (4)sp2: どのようなご用件でしょうか。)
      |     +--MAKE-REGISTRATION
      |           |--INTRODUCE-OBJECT-PLAN
      |           |     +--(INTRODUCE-OBJECT (5)sp1: 会議に申込たいのですが、)
      |           |--ASK-ACTION-PLAN
      |           |     +--ASK-ACTION-UNIT
      |           |           |--(ASK-ACTION (6)sp1: どのような手続きをすればよ
ろしいのでしょうか)
      |           |           +--(DIRECTION (7)sp2: 所定の登録用紙で手続きをして
下さい。)
      |           |--GET-THE-FORM
      |           |     |--ASK-TRUTH-PLAN
      |           |     |     +--ASK-TRUTH-UNIT
      |           |     |           |--(ASK-TRUTH-N (8)sp2: 登録用紙はすでにお持
ちでしょうか。)
      |           |     |           |--(NEGATIVE-TRUTH (9)sp1: いいえ、まだです
。)
      |           |     |           +--(CONFIRM (10)sp2: 分かりました。)
      |           |     |--INTRODUCE-DOMAIN-PLAN-NODE
      |           |     |     +--WILL-DO-ACTION-UNIT
```

```
      |           |     |           |--(WILL-DO-ACTION-A (11)sp2: それでは、登録
用紙をお送り致します。)
      |           |     |           +--(ACCEPT-OFFER (16)sp1: よろしくお願いしま
す。)
      |           |     |--ACHIEVE-KNOW
      |           |     |     +--GET-VALUE-UNIT
      |           |     |           |--(ASK-VALUE (12)sp2: ご住所とお名前をお願
いします。)
      |           |     |           |--(INFORM-VALUE (13)sp1: 大阪市北区城見2-1-
61, 鈴木真弓です。)
      |           |     |           +--(CONFIRM (14)sp2: 分かりました。)
      |           |     +--INTRODUCE-DOMAIN-PLAN-NODE
      |           |           +--WILL-DO-ACTION-UNIT
      |           |                 +--(WILL-DO-ACTION-A (15)sp2: 登録用紙を至急
送らせて頂きます。)
      |           |--FILL-THE-FORM
      |           +--RETURN-THE-FORM
      +--CLOSE-DIALOGUE
            +--GREETING-CLOSE-UNIT
                  |--(GREETING-CLOSE (17)sp1: それでは失礼します。)
                  +--GREETING-CLOSE

 [ 対話構造の表示 終了]
END
```

A5.2　対話構造の例

「会議に登録するには、(1)質問者が登録用紙を手に入れて、(2)登録用紙に記入して、(3)登録用紙を返送する」というドメイン･プランに係わる対話の構造を以下に示す。

システムが持っているドメイン･プラン:
　　HEADER :　　MAKE-REGISTRATION
　　DECOMPOSITION : GET-THE-FORM
　　FILL-THE-FORM
　　RETURN-THE-FORM

システムが初期状態として持っているゴールリスト:
　　OPEN-DIALOGUE　　(対話のチャネルを開く)
　　ESTABLISH-TOPIC-UNIT　　(話題を確立する)
　　CLOSE-DIALOGUE　　(対話のチャネルを閉じる)

ドメインに依存しない対話のバリエーションを以下に示す。

対話番号 : A-1
　　発話番号(4)　質問者(sp1)が対話の目的が会議に申し込むことであるという発話を初めにしている。
　　発話番号(8)　事務局(sp2)が「登録用紙を質問者(sp1)に送る」という発話をしている。

対話番号 : 1-1
　　発話番号(4)　事務局(sp2)が対話の目的が何であるかという発話を初めにしている。
　　発話番号(11) 事務局(sp2)が「登録用紙を質問者(sp1)に送る」という発話をしている。

対話番号 : 0-1
　　発話番号(4)　質問者(sp1)が対話の目的が会議に申し込むことであるという発話を初めにしている。
　　発話番号(7)　質問者(sp1)が「登録用紙を質問者(sp1)に送る」という発話をしている。

対話番号 : B-1
　　発話番号(2)　事務局(sp2)が自分が会議の事務局であるという発話を初めにしている。

発話番号(3)　質問者(sp1)が対話の目的が会議に申し込むことであるという発話を初めにしている。
発話番号(8)　質問者(sp1)が「登録用紙を質問者(sp1)に送る」という発話をしている。

図A5-2に動作例の画面のハードコピーを示す。

図A5-2　動作例の画面のハードコピー(対話構造)

[対話の表示]

対話番号 : A-1

(1)sp1: もしもし。	; Hello.
(2)sp1: 通訳電話国際会議事務局ですか。	; Is that the office for the Conference ?
(3)sp2: はい、そうです。	; Yes. Thats right.
(4)sp1: 会議に申込たいのですが、	; I would like to make a registration ; for the conference.
(5)sp2: 登録用紙はすでにお持ちでしょうか。	; Do you already have a registration form ?
(6)sp1: いいえ、まだです。	; No, not yet.
(7)sp2: 分かりました。	; I see.
(8)sp2: それでは、登録用紙をお送り致します。	; Then I will send you an appliation form.
(9)sp2: ご住所とお名前をお願いします。	; Would you give me your name and address ?
(10)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。	; My address is 23 Chaya-machi, Kita-ku, Osaka ; My name is Mayumi Suzuki
(11)sp2: 分かりました。	; I see.
(12)sp1: それでは失礼します。	; Goodbye.
(13)sp2: それでは失礼します。	; Goodbye.

[対話の表示終了]

```
----- 理解状態 [ 1/1 ] -----
+--TOP-OF-DISCOURSE-STRUCTURE
      |--OPEN-DIALOGUE
      |     |--GREETING-OPEN-UNIT
      |     |     +--(GREETING-OPEN (1)sp1: もしもし。)
      |     +--CONFIRM-PARTNER-UNIT
      |           |--(CONFIRM-VALUE-A (2)sp1: 通訳電話国際会議事務局ですか。)
      |           +--(AFFIRMATIVE (3)sp2: はい、そうです。)
      |--MAKE-REGISTRATION
      |     |--ESTABLISH-TOPIC-UNIT
      |     |     +--(INTRODUCE-OBJECT (4)sp1: 会議に申込たいのですが、)
      |     |--GET-THE-FORM
      |     |     |--ASK-TRUTH-PLAN
      |     |     |     +--ASK-TRUTH-UNIT
      |     |     |           |--(ASK-TRUTH-N (5)sp2: 登録用紙はすでにお持ちでしょうか。)
      |     |     |           |--(NEGATIVE-TRUTH (6)sp1: いいえ、まだです。)
      |     |     |           +--(CONFIRM (7)sp2: 分かりました。)
      |     |     |--INTRODUCE-DOMAIN-PLAN-NODE
      |     |     |     +--WILL-DO-ACTION-UNIT
      |     |     |           |--(WILL-DO-ACTION-A (8)sp2: それでは、登録用紙を お送り致します。)
      |     |     |           +--ACCEPT-OFFER
      |     |     +--ACHIEVE-KNOW
      |     |           +--GET-VALUE-UNIT
      |     |                 |--(ASK-VALUE (9)sp2: ご住所とお名前をお願いします。)
      |     |                 |--(INFORM-VALUE (10)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。)
      |     |                 +--(CONFIRM (11)sp2: 分かりました。)
      |     |--FILL-THE-FORM
      |     +--RETURN-THE-FORM
      +--CLOSE-DIALOGUE
            +--GREETING-CLOSE-UNIT
                  |--(GREETING-CLOSE (12)sp1: それでは失礼します。)
                  +--(GREETING-CLOSE (13)sp2: それでは失礼します。)
```

対話番号 : 1-1

(1)sp1: もしもし。 ; Hello.
(2)sp1: 通訳電話国際会議事務局ですか。 ; Is that the office for the Conference ?
(3)sp2: はい、そうです。 ; Yes.
(4)sp2: どのようなご用件でしょうか。 ; May I help you ?
(5)sp1: 会議に申込たいのですが、 ; I would like to make a registration
; for the Conference.
(6)sp1: どのような手続きをすればよろしいのでしょうか ; How can I apply ?
(7)sp2: 所定の登録用紙で手続きをして下さい。 ; There is an application form we can send you.
(8)sp2: 登録用紙はすでにお持ちでしょうか。 ; Do you have one ?
(9)sp1: いいえ、まだです。 ; No, not yet.
(10)sp2: 分かりました。 ; I see.
(11)sp2: それでは、登録用紙をお送り致します。 ; Then, I'll send you an application form.
(12)sp2: ご住所とお名前をお願いします。 ; Would you give me your name and address ?
(13)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。 ; My address is 23 Chaya-machi, Kita-ku, Osaka
; My name is Mayumi Suzuki
(14)sp2: 分かりました。 ; I see.
(16)sp1: それでは失礼します。 ; Goodbye.

[対話の表示終了]

```
 ----- 理解状態 [ 1/1 ] -----
+--TOP-OF-DISCOURSE-STRUCTURE
      |--OPEN-DIALOGUE
      |     |--GREETING-OPEN-UNIT
      |     |     +--(GREETING-OPEN (1)sp1: もしもし。)
      |     +--CONFIRM-PARTNER-UNIT
      |           |--(CONFIRM-VALUE-A (2)sp1: 通訳電話国際会議事務局ですか。)
      |           +--(AFFIRMATIVE (3)sp2: はい、そうです。)
      |--ESTABLISH-TOPIC-UNIT
      |     |--(ASK-TOPIC (4)sp2: どのようなご用件でしょうか。)
      |     +--MAKE-REGISTRATION
      |           |--INTRODUCE-OBJECT-PLAN
      |           |     +--(INTRODUCE-OBJECT (5)sp1: 会議に申込たいのですが、)
      |           |--ASK-ACTION-PLAN
      |           |     +--ASK-ACTION-UNIT
      |           |           |--(ASK-ACTION (6)sp1: どのような手続きをすればよ ろしいのでしょうか)
      |           |           +--(DIRECTION (7)sp2: 所定の登録用紙で手続きをして下さい。)
      |           |--GET-THE-FORM
      |           |     |--ASK-TRUTH-PLAN
      |           |     |     +--ASK-TRUTH-UNIT
      |           |     |           |--(ASK-TRUTH-N (8)sp2: 登録用紙はすでにお持ちでしょうか。)
      |           |     |           |--(NEGATIVE-TRUTH (9)sp1: いいえ、まだです 。)
      |           |     |           +--(CONFIRM (10)sp2: 分かりました。)
      |           |     |--INTRODUCE-DOMAIN-PLAN-NODE
      |           |     |     +--WILL-DO-ACTION-UNIT
      |           |     |           |--(WILL-DO-ACTION-A (11)sp2: それでは、登録用紙をお送り致します。)
      |           |     |           +--ACCEPT-OFFER
      |           |     +--ACHIEVE-KNOW
      |           |           +--GET-VALUE-UNIT
      |           |                 |--(ASK-VALUE (12)sp2: ご住所とお名前をお願 いします。)
      |           |                 |--(INFORM-VALUE (13)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。)
      |           |                 +--(CONFIRM (14)sp2: 分かりました。)
```

```
|           |--FILL-THE-FORM
|           +--RETURN-THE-FORM
+--CLOSE-DIALOGUE
      +--GREETING-CLOSE-UNIT
            |--(GREETING-CLOSE (16)sp1: それでは失礼します。)
            +--GREETING-CLOSE
```

[対話の表示]

対話番号 : 0-1

(1)sp1: もしもし。 ; Hello.
(2)sp1: 通訳電話国際会議事務局ですか。 ; Is that the Conference office ?
(3)sp2: はい、そうです。 ; Yes. That's right.
(4)sp1: 会議に申込たいのですが、 ; I would like to make a registration
; for the conference.
(5)sp1: どのような手続きをすればよろしいのでしょうか ; What should I do ?
(6)sp2: 所定の登録用紙で手続きをして頂ます。 ; There is an application form we can send you.
(7)sp1: 登録用紙を送って下さい。 ; Please send me a form
(8)sp2: ご住所とお名前をお願いします。 ; Would you give me your name and address?
(9)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。 ; My address is 23 Chaya-machi, Kita-ku, Osaka
; My name is Mayumi Suzuki
(10)sp2: 分かりました。 ; I see.
(11)sp1: 締切はいつですか? ; When is the deadline ?
(12)sp2: 9月19日です。 ; September 19.
(13)sp2: お急ぎ下さい。 ; Please be hurry.
(14)sp1: それでは失礼します。 ; Goodbye.
(15)sp2: どうも失礼します。 ; Goodbye.

[対話の表示終了]

```
 ----- 理解状態 [ 1/4 ] -----
+--TOP-OF-DISCOURSE-STRUCTURE
      |--OPEN-DIALOGUE
      |     |--GREETING-OPEN-UNIT
      |     |     +--(GREETING-OPEN (1)sp1: もしもし。)
      |     +--CONFIRM-PARTNER-UNIT
      |           |--(CONFIRM-VALUE-A (2)sp1: 通訳電話国際会議事務局ですか。)
      |           +--(AFFIRMATIVE (3)sp2: はい、そうです。)
      |--MAKE-REGISTRATION
      |     |--ESTABLISH-TOPIC-UNIT
      |     |     +--(INTRODUCE-OBJECT (4)sp1: 会議に申込たいのですが、)
      |     |--ASK-ACTION-PLAN
      |     |     +--ASK-ACTION-UNIT
      |     |           |--(ASK-ACTION (5)sp1: どのような手続きをすればよろしい のでしょうか)
      |     |           +--(DIRECTION (6)sp2: 所定の登録用紙で手続きをして頂ます。)
      |     |--GET-THE-FORM
      |     |     |--INTRODUCE-DOMAIN-PLAN-NODE
      |     |     |     +--REQUEST-ACTION-UNIT
      |     |     |           |--(REQUEST-ACTION-A (7)sp1: 登録用紙を送って下さ い。)
      |     |     |           +--(ACCEPT (10)sp2: 分かりました。)
      |     |     +--ACHIEVE-KNOW
      |     |           +--GET-VALUE-UNIT
      |     |                 |--(ASK-VALUE (8)sp2: ご住所とお名前をお願いします。)
      |     |                 |--(INFORM-VALUE (9)sp1: 大阪市北区城見2-1-61, 鈴 木真弓です。)
      |     |                 +--CONFIRM
      |     |--RETURN-THE-FORM
      |     |     |--INTRODUCE-OBJECT-PLAN
      |     |     |     +--GET-VALUE-UNIT
      |     |     |           |--(ASK-VALUE (11)sp1: 締切はいつですか?)
      |     |     |           |--(INFORM-VALUE (12)sp2: 9月19日です。)
      |     |     |           +--CONFIRM
```

```
|      |      +--INTRODUCE-DOMAIN-PLAN-NODE
|      |           +--REQUEST-ACTION-UNIT
|      |                |--(REQUEST-ACTION-A (13)sp2: お急ぎ下さい。)
|      |                +--ACCEPT
|      +--FILL-THE-FORM
+--CLOSE-DIALOGUE
       +--GREETING-CLOSE-UNIT
            |--(GREETING-CLOSE (14)sp1: それでは失礼します。)
            +--(GREETING-CLOSE (15)sp2: どうも失礼します。)
```

[対話の表示]

対話番号 : B-1

(1)sp1: もしもし。 ; Hello.
(2)sp2: こちらは会議事務局です。 ; This is the office for the Conference.
(3)sp1: 会議に申込たいのですが、 ; I'd like to make a registration for the conference.
(4)sp1: どのような手続きをすればよろしいのでしょうか ; How can I apply ?
(5)sp2: 所定の登録用紙で手続きをして下さい。 ; There is an application form we can send you.
(6)sp2: 登録用紙はすでにお持ちでしょうか。 ; Do you have one ?
(7)sp1: いいえ、まだです。 ; No, not yet.
(8)sp1: 登録用紙を送って下さい。 ; Please send me a form.
(9)sp2: ご住所とお名前をお願いします。 ; Would you give me your name and address ?
(10)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。 ; My address is 2-1-61 Kita-ku Shiromi Osaka.
; My name is Mayumi Suzuki.
(11)sp2: 分かりました。 ; I see.
(16)sp1: どうもありがとうございました。 ; Thank you.
(17)sp2: それでは失礼します。 ; Goodbye.
[対話の表示終了]

```
 ----- 理解状態 [ 1/3 ] -----
+--TOP-OF-DISCOURSE-STRUCTURE
      |--OPEN-DIALOGUE
      |     |--GREETING-OPEN-UNIT
      |     |     +--(GREETING-OPEN (1)sp1: もしもし。)
      |     +--CONFIRM-PARTNER-UNIT
      |           +--(INFORM-VALUE (2)sp2:こちらは会議事務局です。)
      |--MAKE-REGISTRATION
      |     |--ESTABLISH-TOPIC-UNIT
      |     |     +--(INTRODUCE-OBJECT (3)sp1: 会議に申込たいのですが、)
      |     |--ASK-ACTION-PLAN
      |     |     +--ASK-ACTION-UNIT
      |     |           |--(ASK-ACTION (4)sp1: どのような手続きをすればよろしい のでしょうか)
      |     |           +--(DIRECTION (5)sp2: 所定の登録用紙で手続きをして下さい。)
      |     |--GET-THE-FORM
      |     |     |--ASK-TRUTH-PLAN
      |     |     |     +--ASK-TRUTH-UNIT
      |     |     |           |--(ASK-TRUTH-N (6)sp2: 登録用紙はすでにお持ちでしょうか。)
      |     |     |           |--(NEGATIVE-TRUTH (7)sp1: いいえ、まだです。)
      |     |     |           +--CONFIRM
      |     |     |--INTRODUCE-DOMAIN-PLAN-NODE
      |     |     |     +--REQUEST-ACTION-UNIT
      |     |     |           |--(REQUEST-ACTION-A (8)sp1: 登録用紙を送って下さ い。)
      |     |     |           +--(ACCEPT (11)sp2: 分かりました。)
      |     |     +--ACHIEVE-KNOW
      |     |           +--GET-VALUE-UNIT
      |     |                 |--(ASK-VALUE (9)sp2: ご住所とお名前をお願いします。)
      |     |                 |--(INFORM-VALUE (10)sp1: 大阪市北区城見2-1-61, 鈴木真弓です。)
      |     |                 +--CONFIRM
      |     |--FILL-THE-FORM
      |     +--RETURN-THE-FORM
      +--CLOSE-DIALOGUE
```

```
+--GREETING-CLOSE-UNIT
      |--(GREETING-CLOSE (16)sp1: どうもありがとうございました。)
      +--(GREETING-CLOSE (17)sp2: それでは失礼します。)
```

A5-3　ドメイン･プラン、ディスコース･プラン、インターラクション･プランの例

```
;;; -*- Mode: LISP; Base: 10; Package: USER; Syntax: Common-lisp -*-

;;;
;;;  スキーマの表現
;;;
;;; ((HEADER         (header-name variables))
;;;  (PREREQUISITE   (predicate variables)
;;;                  ....                    )
;;;  (DECOMPOSITION (predicate variables)
;;;                  .....                   )
;;;  (ADD-EFFECT     (predicate variables)
;;;                  .....                   )
;;;  (DEL-EFFECT     (predicate variables)
;;;                  .....                   )
;;;  (CONSTRAINT
;;;  (BELONG-TO           literal)  ;; utterance-type | interaction-plan |
;;;                                    domain-plan | discourse-plan
;;;  )
;;;
```

```
;;;;;;;;;;;;;;;;;;;;;
;;;                 ;;;
;;;  ドメインプラン;;;
;;;                 ;;;
;;;;;;;;;;;;;;;;;;;;;

;;;-----------------------------------------------------------------
;;;   MAKE-REGISTRATION
;;;      |--- GET-THE-FORM
;;;      |--- FILL-THE-FORM
;;;      +--- RETURN-THE-FORM
;;;-----------------------------------------------------------------
((header (MAKE-REGISTRATION ?sp ?hr))
 (decomposition (get-the-form ?sp ?hr ?form)
                (fill-the-form ?sp ?hr ?form)
                (return-the-form ?sp ?hr ?form))
 (pred-structure (pred&cases (predicate moushikomu)(agent ?sp)(object kaigi)))
 (belong-to domain-plan))

((header (GET-THE-FORM ?sp ?hr ?form))
 (prerequisite (know ?hr name&address ?n&a))
 (decomposition (introduce-domain-plan-node
                  (pred&cases (predicate okuru)(agent ?hr)(recipient ?sp)(object
 ?form)
                              (manner ?manner))))
 (add-effect (pred&cases (predicate motsu)(agent ?sp)(object ?form)))
 (belong-to domain-plan))

((header (FILL-THE-FORM ?sp ?hr ?form))
; (prerequisite (have ?sp ?form))
 (decomposition (introduce-domain-plan-node
                  (pred&cases (predicate kaku)(agent ?sp)(recipient ?dummy)(obje
```

```
ct ?form)
                                (manner ?manner))))
 (belong-to domain-plan))

((header (RETURN-THE-FORM ?sp ?hr ?form))
; (prerequisite (have ?sp ?form))
 (decomposition (introduce-domain-plan-node
                  (pred&cases (predicate okuru)(agent ?sp)(recipient ?hr)(object
 ?form)
                                (manner ?manner))))
 (add-effect (have ?hr ?form))
 (del-effect (have ?sp ?form))
 (belong-to domain-plan))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                      ;;;
;;; ディスコースプラン ;;;
;;;                      ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;-------------------------------------------------------------
;;; INTRODUCE-DOMAIN-PLAN-NODE
;;;    decomposition1 : REQUEST-ACTION-UNIT
;;;    decomposition2 : WILL-DO-ACTION-UNIT
;;;-------------------------------------------------------------
((header (introduce-domain-plan-node ?action))
 (decomposition (request-action-unit ?sp ?hr ?clue ?topic ?action))
 (belong-to discourse-plan))
((header (introduce-domain-plan-node ?action))
 (decomposition (WILL-DO-action-unit ?sp ?hr ?clue ?topic ?action))
 (belong-to discourse-plan))

;;;-------------------------------------------------------------
;;; ASK-ACTION-PLAN
;;;    decomposition1 : ASK-ACTION-UNIT
;;;    decomposition1 : REQUEST-ACTION-UNIT
;;;-------------------------------------------------------------

((header (ask-action-plan ?action))
 (decomposition (ask-action-unit ?sp ?hr ?clue ?topic ?action))
 (goal-effect (ask-action ?action ?topic))
 (belong-to discourse-plan))
((header (ask-action-plan ?action))
 (decomposition (request-action-unit ?sp ?hr ?clue ?topic ?action))
 (goal-effect (ask-action ?action ?topic))
 (belong-to discourse-plan))
```

```
;;;-------------------------------------------------------------
;;; REQUIRE-QUESTION-PLAN
;;;     decomposition : REQUIRE-QUESTION-UNIT
;;;-------------------------------------------------------------

((header (require-question-plan ?action))
 (decomposition (require-question-unit ?sp ?hr ?clue ?topic ?action))
 (belong-to discourse-plan))

;;;-------------------------------------------------------------
;;; ASK-TRUTH-PLAN
;;;     decomposition : ASK-TRUTH-UNIT
;;;-------------------------------------------------------------

((header (ask-truth-plan ?action))
 (decomposition (ask-truth-unit ?sp ?hr ?clue ?topic ?action))
 (inference-path (add-effect prerequisite))
 (belong-to discourse-plan))

;;;-------------------------------------------------------------
;;; INTRODUCE-OBJECT-PLAN
;;;     decomposition1 : GET-VALUE-UNIT
;;;     decomposition2 : CONFIRM-VALUE-UNIT
;;;     decomposition3 : CONFIRM-VALUE2-UNIT
;;;     decomposition4 : INTRODUE-OBJET
;;;-------------------------------------------------------------
((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
 (decomposition (get-value-unit ?sp ?hr ?clue ?topic ?action))
 (goal-effect (focus-plan ?topic))
 (belong-to discourse-plan))
((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
```

```
 (decomposition (confirm-value-unit ?sp ?hr ?clue ?topic ?action))
 (goal-effect (focus-plan ?topic))
 (belong-to discourse-plan))
((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
 (decomposition (confirm-value2-unit ?sp ?hr ?clue ?topic ?action))
 (goal-effect (focus-plan ?topic))
 (belong-to discourse-plan))
((header (introduce-object-plan ?sp ?hr ?clue ?topic ?action))
 (decomposition
               (introduce-object ?sp ?hr ?clue ?topic ?action))
 (goal-effect (focus-plan ?topic))
 (belong-to discourse-plan))

;;;-------------------------------------------------------------
;;; OPEN-DIALOGUE
;;;    decomposition : GREETING-OPEN-UNIT
;;; CLOSE-DIALOGUE
;;;    decomposition : GREETING-CLOSE-UNIT
;;;-------------------------------------------------------------
((header (open-dialogue ?sp ?hr ?clue ?topic ?action))
 (decomposition (greeting-open-unit ?sp ?hr ?clue ?topic ?action)        ;;; cha
nnel の接続
               (confirm-partner-unit ?sp ?hr ?clue1 ?topic1              ;;; 相
手の確認
                                     (pred&cases (predicate desu)(object ?hr)(ident
ifier ?id))))
 (belong-to discourse-plan))
((header (close-dialogue ?sp ?hr ?clue ?topic ?action))
 (decomposition (greeting-close-unit ?sp ?hr ?clue ?topic ?action))
 (belong-to discourse-plan))

;;;-------------------------------------------------------------
```

```
;;; ACHIEVE-KNOW
;;;    decomposition : GET-VALUE-UNIT
;;;-------------------------------------------------------------
((header (achieve-know ?agent ?object ?value))
 (decomposition (get-value-unit ?agent ?hr ?clue ?topic
                      (pred&cases (predicate desu)(object ?object)(identifier ?
value))))
 (belong-to discourse-plan))

;;;-------------------------------------------------------------
;;; ESTABLISH-TOPIC-UNIT  ;;; 話題の確立
;;;    decomposition1 : introduce-object    -- についてお伺いしたいのですが
;;;
;;;    decomposition2 : ask-topic          何かご用でしょうか
;;;                     introduce-object    -- についてお伺いしたいのですが
;;;-------------------------------------------------------------
((header (establish-topic-unit ?sp ?hr ?clue ?topic  (pred&cases (predicate oshi
eru)
                                                    (agent ?hr)(recipient ?sp)(obj
ect ?topic))))

 (decomposition (introduce-object ?sp ?hr ?clue ?topic (pred&cases (predicate os
hieru)
                                                    (agent ?hr)(recipient ?sp)(obj
ect ?topic))))
 (goal-effect (focus-plan ?topic))
 (belong-to discourse-plan))
((header (establish-topic-unit ?sp ?hr ?clue ?topic   (pred&cases (predicate osh
ieru)
                                                    (agent ?hr)(recipient ?sp)(obj
ect ?topic))))
```

```
 (decomposition (ask-topic ?hr ?sp ?clue ?topic (pred&cases (predicate desu)
                                                            (object ?object)(ident
ifier ?id)))
                (introduce-object-plan ?sp ?hr ?clue ?topic (pred&cases (predica
te oshieru)
                                                            (agent ?hr)(recipient ?sp)(obj
ect ?topic))))
 (goal-effect (focus-plan ?topic))
 (belong-to discourse-plan))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;                          ;;;
;;; インターラクションプラン ;;;
;;;                          ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;--------------------------------------------------------------
;;; CONFIRM-PARTNER-UNIT  ;;; 相手の確認
;;;    decomposition1 : confirm-value-a    ?hr は ?id ですか
;;;                     affirmative        はい、そうです
;;;    decomposition2 : inform-value       ?hr は ?id です
;;;--------------------------------------------------------------
((header (confirm-partner-unit ?sp ?hr ?clue ?topic
                               (pred&cases (predicate desu)(object ?hr)(identifi
er ?id))))
 (decomposition (confirm-value-a ?sp ?hr ?clue ?topic (pred&cases (predicate des
u)
                                                       (object ?hr)(identifie
r ?id)))
                (affirmative ?hr ?sp ?clue ?topic  (pred&cases (predicate desu)
                                                       (object ?hr)(identifie
r ?id))))
 (add-effect (know ?sp ?hr ?id))
 (belong-to interaction-plan))
((header (confirm-partner-unit ?sp ?hr ?clue ?topic
                               (pred&cases (predicate desu)(object ?hr)(identifi
er ?id))))
 (decomposition (inform-value ?hr ?sp ?clue ?topic (pred&cases (predicate desu)
                                                       (object ?hr)(identifie
r ?id))))
 (belong-to interaction-plan))
```

```
;;;-----------------------------------------------------------------
;;; GET-VALUE-UNIT
;;;    decomposition1 : ask-value        ?object は wh ですか
;;;                     inform-value     ?id です
;;;                     confirm          分かりました
;;;
;;;    decomposition2   inform-value     ?id です
;;;-----------------------------------------------------------------
((header (get-value-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(objec
t ?object)
                                                          (identifier ?id))))
 (decomposition (ask-value ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(obj
ect ?object)
                                                          (identifier ?id)))
                (inform-value ?hr ?sp ?clue ?topic (pred&cases (predicate desu)
                                                          (object ?object)(iden
tifier ?id)))
                (confirm ?sp ?hr ?clue ?topic ?action))
 (add-effect (know ?sp ?object ?id))
 (belong-to interaction-plan))

((header (get-value-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(objec
t ?object)
                                                          (identifier ?id))))
 (decomposition (inform-value ?hr ?sp ?clue ?topic (pred&cases (predicate desu)
                                                          (object ?object)(iden
tifier ?id))))
 (add-effect (know ?sp ?object ?id))
 (belong-to interaction-plan))

;;;-----------------------------------------------------------------
```

```
;;; ASK-EXIST-VALUE-UNIT
;;;    decomposition1 : ask-exist-value-a        ?object はありますか / ?objet
は要るのですか
;;;                     affirmative-exist        はい、あります / はい、要ります

;;;    decomposition2 : ask-exist-value-n        ?object はありますか / ?objet
は要るのですか
;;;                     negative-exist           いいえ、ありません / いいえ、要
りません
;;;
;;;-----------------------------------------------------------------
((header (ask-exist-value-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (ask-exist-value-a ?sp ?hr ?clue ?topic ?action)
                (affirmative-exist ?hr ?sp ?clue ?topic ?action2))
 (belong-to interaction-plan))

((header (ask-exist-value-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (ask-exist-value-n ?sp ?hr ?clue ?topic ?action)
                (negative-exist ?hr ?sp ?clue ?topic ?action2))
 (belong-to interaction-plan))

;;;-----------------------------------------------------------------
;;; REQUEST-ACTION-UNIT
;;;    decomposition1 : request-action-a ~してください
;;;                     accept          わかりました
;;;    decomposition1 : request-action-r ~してください
;;;                     reject          申し訳ありませんが、ーできません
;;;-----------------------------------------------------------------
((header (request-action-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (request-action-a ?sp ?hr ?clue ?topic ?action)
                (accept ?hr ?sp ?clue ?topic (pred&cases (predicate wakaru)(agen
t ?hr)
```

```
                                                    (recipient ?recip)
                                                    (object ?action)
                                                    (manner ?manner))))
 (belong-to interaction-plan))
((header (request-action-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (request-action-r ?sp ?hr ?clue ?topic ?action)
                (reject ?hr ?sp ?clue ?topic ?action2))
 (belong-to interaction-plan))

;;;-------------------------------------------------------------
;;; ASK-TRUTH-UNIT
;;;    decomposition1 : ask-truth-n  (proposition)か
;;;                     negative-truth いいえ,(proposition)
;;;                     confirm      分かりました
;;;    decomposition2 : ask-truth-a  (proposition)か
;;;                     affirmative-truth はい,(proposition)
;;;    decomposition3 : ask-truth-i  (proposition)か
;;;                     inform     (proposition)
;;;-------------------------------------------------------------
((header (ask-truth-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (ask-truth-n ?sp ?hr ?clue ?topic ?action)
                (negative-truth ?hr ?sp ?clue ?topic (pred&cases (predicate ?pre
d)))
                (confirm ?sp ?hr ?clue ?topic ?action2) )
 (belong-to interaction-plan))
((header (ask-truth-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (ask-truth-a ?sp ?hr ?clue ?topic ?action)
                (affirmative-truth ?hr ?sp ?clue ?topic (pred&cases (predicate ?
pred))))
 (belong-to interaction-plan))
((header (ask-truth-unit ?sp ?hr ?clue ?topic ?action2))
 (decomposition (ask-truth-i ?sp ?hr ?clue ?topic ?action)
```

```
                (inform ?hr ?sp ?clue ?topic ?action2))
 (belong-to interaction-plan))


;;;-------------------------------------------------------------
;;; WILL-DO-ACTION-UNIT
;;;    decomposition1 : will-do-action-a ~します
;;;                     accept-offer     はい、お願いします
;;;    decomposition2 : will-do-action-n ~します
;;;                     reject-offer     いいえ、結構です。
;;;-------------------------------------------------------------
((header (will-do-action-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (will-do-action-a ?sp ?hr ?clue ?topic ?action)
                (accept-offer ?hr ?sp ?clue ?topic (pred&cases (predicate negau)
(agent ?hr)
                                                    (recipient ?sp)
                                                    (object ?action)
                                                    (manner ?manner))))
 (belong-to interaction-plan))
((header (will-do-action-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (will-do-action-n ?sp ?hr ?clue ?topic ?action)
                (reject-offer ?hr ?sp ?clue ?topic ?action2))
 (belong-to interaction-plan))

;;;-------------------------------------------------------------
;;; ASK-ACTION-UNIT
;;;    decomposition : ask-action ~するにはどうすればよいですか
;;;                    direction  ~してください
;;;-------------------------------------------------------------
((header (ask-action-unit ?sp ?hr ?clue ?topic ?action))
 (decomposition (ask-action ?sp ?hr ?clue ?topic ?action)
                (direction ?hr ?sp ?clue ?topic ?action2))
```

```
 (belong-to interaction-plan))

;;;-----------------------------------------------------------------
;;; CONFIRM-VALUE-UNIT
;;;    decomposition : confirm-value-a ?object は ?id ですか
;;;                    affirmative     はい、そうです
;;;    decomposition : confirm-value-n ?object は ?id ですか
;;;                    negative        いいえ、ちがいます
;;;    decomposition : confirm-value-ai              ?object は ?id ですか
;;;                    affirmative-inform-value      はい、?idです
;;;
;;;    decomposition : confirm-value-ni              ?object は ?id ですか
;;;                    negative-inform-value         いいえ、?id2です
;;;-----------------------------------------------------------------
((header (confirm-value-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(o
bject ?object)
                                                    (identifier ?id))))
 (decomposition (confirm-value-a ?sp ?hr ?clue ?topic (pred&cases (predicate des
u)
                                                   (object ?object)(ident
ifier ?id)))
                (affirmative ?hr ?sp ?clue ?topic  (pred&cases (predicate desu)
                                                   (object ?object)(ident
ifier ?id))))
 (belong-to interaction-plan))
((header (confirm-value-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(o
bject ?object)
                                                    (identifier ?id))))
 (decomposition (confirm-value-n ?sp ?hr ?clue ?topic (pred&cases (predicate des
u)
                                                   (object ?object)(ident
ifier ?id)))
```

```
                (negative ?hr ?sp ?clue ?topic  (pred&cases (modal negative)(pre
dicate desu)
                                                                (object ?object)(ident
ifier ?id))))
 (belong-to interaction-plan))
((header (confirm-value-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(o
bject ?object)
                                                                  (identifier ?id))))
 (decomposition (confirm-value-ai ?sp ?hr ?clue ?topic (pred&cases (predicate de
su)
                                                                (object ?object)(ident
ifier ?id)))
                (affirmative-inform-value ?hr ?sp ?clue ?topic  (pred&cases (pre
dicate desu)
                                                                (object ?object)(ident
ifier ?id))))
 (belong-to interaction-plan))
((header (confirm-value-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(o
bject ?object)
                                                                  (identifier ?id2))))
 (decomposition (confirm-value-ni ?sp ?hr ?clue ?topic (pred&cases (predicate de
su)
                                                                (object ?object)(ident
ifier ?id)))
                (negative-inform-value ?hr ?sp ?clue ?topic
                                        (pred&cases (modal negative)(predicate de
su)
                                                                (object ?object)(ident
ifier ?id2))))
 (belong-to interaction-plan))

;;;-----------------------------------------------------------------
```

```
;;; CONFIRM-VALUE2-UNIT
;;;    decomposition : confirm-value2 ?object は ?id1 ですか ?id2 ですか
;;;                    inform-value    ?id1/?id2 です
;;;    decomposition : confirm-value2-2 ?object は ?id1 ですか ?id2 ですか
;;;                    inform-value2    両方 です
;;;-------------------------------------------------------------
((header (confirm-value2-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(
object ?object)
                                                    (identifier ?id))))
 (decomposition (confirm-value2 ?sp ?hr ?clue ?topic (pred&cases (predicate desu
)
                                                    (object ?object)(ident
ifier ?id1)))
                (inform-value ?hr ?sp ?clue ?topic  (pred&cases (predicate desu)
                                                    (object ?object)(ident
ifier ?id))))
 (belong-to interaction-plan))
((header (confirm-value2-unit ?sp ?hr ?clue ?topic (pred&cases (predicate desu)(
object ?object)
                                                    (identifier ?id))))
 (decomposition (confirm-value2-2 ?sp ?hr ?clue ?topic (pred&cases (predicate de
su)
                                                    (object ?object)(ident
ifier ?id)))
                (inform-value2 ?hr ?sp ?clue ?topic  (pred&cases (predicate desu
)
                                                    (object ?object)(ident
ifier ?id))))
 (belong-to interaction-plan))

;;;-------------------------------------------------------------
;;; GREETING-CLOSE-UNIT
```

```
;;;    decomposition : greeting-close    失礼します
;;;                    greeting-close    失礼します
;;;-----------------------------------------------------------
((header (greeting-close-unit ?sp ?hr ?clue ?topic (pred&cases (predicate ?pred)
                                                              (agent ?sp))))
 (decomposition (greeting-close ?sp ?hr ?clue ?topic (pred&cases (predicate ?pre
d)
                                                                (agent ?sp)))
                (greeting-close ?hr ?sp ?clue1 ?topic1 (pred&cases (predicate ?p
red1)
                                                                  (agent ?hr)))
)
 (belong-to interaction-plan))

;;;-----------------------------------------------------------
;;; GREETING-OPEN-UNIT
;;;    decomposition : greeting-open     もしもし
;;;-----------------------------------------------------------
((header (greeting-open-unit ?sp ?hr ?clue ?topic (pred&cases (predicate ?pred)(
agent ?sp))))
 (decomposition (greeting-open ?sp ?hr ?clue ?topic (pred&cases (predicate ?pred
)(agent ?sp))))
 (belong-to interaction-plan))

;;;-----------------------------------------------------------
;;; REQUIRE-QUESTION-UNIT
;;;    decomposition : require-question  他に何かありますか
;;;                                      はい、何でしょうか
;;;-----------------------------------------------------------
((header (require-question-unit ?sp ?hr ?clue ?topic
                (pred&cases (predicate ?pred)(agent ?hr)(recipient ?sp)(object ?
youken))))
```

```
 (decomposition (require-question ?sp ?hr ?clue ?topic
                (pred&cases (predicate ?pred)(agent ?hr)(recipient ?sp)(object ?
youken))))
 (belong-to interaction-plan))
```