人工知能と対話技法 25－5
（1982. 3. 25）

自然言語処理技術研究の新たな展開に向けて

田中穂積
（電子技術総合研究所）

1. はじめに

自然言語処理技術に対する関心と期待が，我が国でもようやく高まってきている。自然言語処理技術に関する研究のむずかしさとおもしろさは，好むと好まざるとにかかわらず，知識の問題と対峙しなければならない点にある。（この問題の重要性の認識こそ知識工学発想の原点であったと考えられる。）第五世代コンピュータは，別名，知識情報処理システムとよばれている。自然言語処理技術を統合化したシステムの一つに自然言語理解システムがある。これを仔細に観察してみれば，これもまた知識情報処理システムの一つであることが分かる。

自然言語処理技術の研究は，1970年代には，意味の問題が主要な研究課題の一つとして取り上げられていた。そこでまず初めに，同音異義語の処理，特に同一品詞の同音異義語の意味処理の問題を論じ，一つの考え方を提案する。この考え方が機械翻訳にも応用できることを示す。

次にPrologによる自然言語処理の利点を，筆者等が開発した拡張LINGOLと対比させて論じる。

最後に，自然言語による質問応答システムにおける会話のモデルの問題に簡単にふれる。

2. 同音異義語の意味処理と機械翻訳

同音異義語には，
(i) 品詞が同じ：かう（買う，飼う），pen（ペン，囲い）
(ii) 品詞が異なる：よい（良い，宵……），lie（うそ，横たわる），
のニつがある。

このうち(ii)は，構文解析の段階で品詞が認定され，曖昧さの解消が可能なことが多い。これに対して(i)は，構文解析で，曖昧さを解消することは不可能である。曖昧さの解消には意味解析や文脈解析が必要である。

機械翻訳システムでは(i)に関して別の問題が生じる。次の例文を考えてみる。

(イ) I take a cold.
（私は風邪をひく）
(ロ) I take a bus.
（私はバスに乗る）
(ハ) I take my child to her.
（私は彼女のところへ私の子供を連れて行く）
(ニ) I take my dictionary to her.
（私は彼女のところへ私の辞書を持って行く）

(イ)から(ニ)に出現する"take"は，目的語に応じて様々な日本語に訳し分けなければならない。英語の辞典をみると，"take"は一つの項目に収められている。しかし，日本語の辞典をひいてみると，ひく，乗る，連れて行く……，は異義語として異なる項目で説明されている。言い換えれば，日本語の側からみると，英語のtakeは(i)の型の同音異義語にみえるということである。この様なケースを(ii)の型の同音異義語であるとよぶことにしよう。

以下では，(i)と(ii)の型について，曖昧さの解消の問題を論じることにする。

最初に(ii)の型の問題について考察してみよう。

次の例文を考えてみる。

(ホ) 動物をかう
(ヘ) 箱をかう

(ホ)の文だけからは，"かう" の意味が"買う"であるのか"飼う"であるのか決められない．曖昧さの解消は意味解析だけでは不可能である．曖昧さの解消には，さらに文脈解析が必要になる．

これに対し(ヘ)の"かう"は，"買う"の意味に一意に決まる．曖昧さの解消が，この一文だけで可能である．

このことは，"かう"の目的格にどの様な意味をもつ名詞がくるかに依存して，一文のレベルでの曖昧さの解消が可能な場合のあることを示している．

いずれにしても，これは，"かう"の辞書項目として，意味が異なる二つのエントリーを用意して解析を進めなければならない．筆者等のシステムでは，意味表現用言語SRL[1]で，次の様な辞書項目の記述を行っている．

```
(KAW  VHEAD (····)
      '((KAW-U1・買う) unit
        ·····
①·····  (theme ((OR (a BUTTAI) ····) -WO))
        ·····))

(KAW  VHEAD (·····)
      '((KAW-U2・飼う) unit
        ·····
② ·····  (theme ((a DOUBUTU) -WO))
        ····· ))
```

SRLによるスロット記述①と②とを比較すると，スロットに記述されている<constraint>が異なっていることが分かる．"買う"の場合には，スロット①に，買えるものとして"物体(a BUTTAI)"，…等があることが記述されている．一方"飼う"の場合には，それは"動物(a DOUBUTU)"でなければならないことが記述されている．動物を物体として見なしうることが，動物のunit記述から別途推論できる（SRLのselfスロットを通じて）とすれば，上記した二つの記述の差から，文(ホ)の場合には曖昧さが解消できず，二通りの意味解釈果が出力される．文(ヘ)に対しては次の注意が必要である．"飼う"のunit記述中のスロット②の<constraint>は，スロット①のそれに比べて条件が厳しい．"箱"は"動物"ではないから，スロット②を満足することができず，"箱を飼う"の解釈が棄却され，"箱を買う"の解釈のみが出力される．この様にして，文(ヘ)では"かう"の意味的曖昧性が解消される．

筆者等の研究室では，こうした考え方に基づく意味解析のプログラムが動作している．

文(ホ)の場合にはさらに，二通りの解釈結果を文脈に照らして曖昧さを解消する必要があるが，これは今後の検討課題である．ただSRLの各unit記述には，inferという名称の特別なスロットがあり，そこに手続きを記述することができる[1]．意味解釈が終了した段階で，この手続きを起動して文脈解析を行う道は残されている．

次に(ii)の型の問題を論じることにする．この問題を処理するための直接的な方法は，"take"を(i)型の同音異義語として，日本語の訳語が異なる毎に辞書に別個のエントリーを用意することであろう．英和辞典を調べてみると分かるように，"take"には様々な日本語への訳語がある．手許の研究社の新英和中辞典では，"take"は他動詞として25個，自動詞として6個の用法があげられている．各用法毎に日本語への訳語が異なっているから，この場合には合計31個の同音異義語"take"を作らなければならないことになる．

どの様な単語に対しても少なくとも数個の用法があるので，この様な直接的な方法は，辞書のエントリー数を非常に増加させる．しかも(ii)の型の同音異義語が極端に増えることになり，同音異義語の意味解析では，複数個の同

音異義語の対を考えねばならないことになる。これは意味解析を非常に困難なものにするので好ましいことではない。

英和辞典で"take"が一つのエントリーとして登録されていることの背景には，この31個の用法の相互間に意味的共通基盤が認められるからに他ならない。この考え方を尊重するとすれば，日本語の訳語毎に"take"を同音異義語に分解する方法をとるべきではない。(iii)の型を(i)の型に還元すべきでなく，"take"を一つの辞書項目として素直に問題を解決することが望ましい。図1は，この立場からの"take"の辞書項目記述例である。

文(1)から(二)を観察すると，文(ホ)と(ヘ)と同様に，動詞を囲む目的語や前置詞句の影響を受けて，"take"の日本語訳が決まることが分かる。これは，筆者等の開発した意味表現用言語SRLで容易に実現することができる。スロットに付加された<action>（手続き）を利用するのである。これを，我々の実験的機械翻訳システム*で用いている図1**の辞書記述を利用して概略説明する。

以前に筆者等は，日本語の意味構造を抽出するプログラムEXPLUSを作成した。そして意味構造の抽出法として，構文解析木の構造に沿って行うユニット間会話会話とよばれる方法を提案した。(1)

ユニット間会話の基本は，2つのユニットのうちの片方が，他のユニット中のいずれかのスロットを満たすことができるかどうかを調べることである。ユニット間会話では，前者をFILLER，後者をORIGINとよんでいた。FILLERがORIGIN中のいずれかのスロットを満たすためには，FILLERはスロットに書かれた意味的ならびに統語的制約条件を満たさねばならない。

ユニット間会話の結果としてFILLERがORIGINのいずれかのスロットを満たすことが知れた時，ORIGINの側から見れば，ORIGINを取りまく環境がFILLERであることを知る。これは訳語選択の機会を与える。

SRLでは，FILLERが，あるスロットを満たした時，そのスロットに付加され

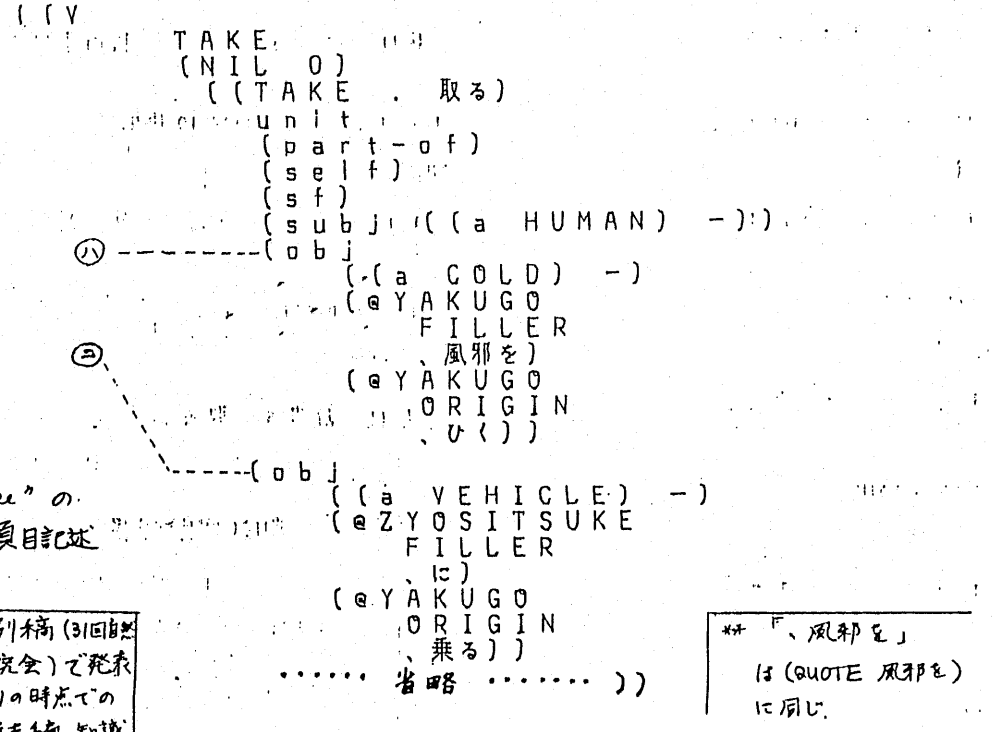

図1. "take"の辞書項目記述

* 詳細は，別稿（31回自然言語処理研究会）で発表する。(2) 2月の終りの時点でのまとめは 田中幸吉 編，知識工学とその応用，昭和56年度文部省報告書。

** 「、風邪を」は(QUOTE 風邪を)に同じ。

ている手続き（<action>）を起動することができる。それにより，適切な訳語を選択することができる。

たとえば図1のスロット㈣は，目的語（obj）として COLD があれば，COLDは FILLERとしてスロット㈡を満たし，その結果，次の<action>

(@YAKUGO FILLER 、風邪を)*

(@YAKUGO ORIGIN 、ひく)

が起動される。これらは FILLER(COLD) の訳語を「風邪を」に，ORIGIN(take) の訳語を「ひく」にする<action>である。

スロット㈢は，FILLERが乗物（a VEHICLE）であれば，FILLERの訳語に助詞「に」を付け，ORIGIN（take）の訳語を「乗る」に変えるべきことを，<action>によって指定している。

以上の例から，我々の意味表現用言語 SRLによる図1の辞書記述は，英和辞典の記述に非常に近いことが理解できよう。文献(1)で指摘したように，各スロットには"take"の例文が記述されていると見なせる。さらに図1の記述では，<action>により，例文の翻訳例が記述されていると見なせる。

図2に，我々の翻訳実験システムの計算機出力結果の例を示す。"take"の訳語として適確なものが選択されていること，また，目的語の助詞としても適確なものが選択されていることが読み取れるであろう。さらに，COLDの訳語として適確な訳し分けが行われていることにも注意されたい。

詳しい説明は別稿に譲るとして，我々が実験している機械翻訳の方式は，意味理解に重点をおき，意味解析（意味構造の抽出）が終了した時点で翻訳文が出力される（意味解析と分離した変換合成過程はない）。意味解析過程と翻訳文変換合成過程とが強く融合した方式である。意味解析には筆者等が

I TAKE A COLD.

翻訳結果

私は 風邪を ひく.

I TAKE A COLD IN THE COLD.

翻訳結果

私は その寒さの中で 風邪を ひく.

I TAKE A BUS.

翻訳結果

私は バスに 乗る.

I TAKE A BATH.

翻訳結果

私は 風呂に 入る.

I TAKE MY CHILD TO HER.

翻訳結果

私は 彼女のところへ 私の子供を 連れて行く

I TAKE MY DICTIONARY TO HER.

翻訳結果

私は 彼女のところへ 私の辞書を 持って行く

I TAKE A DICTIONARY.

翻訳結果

私は 辞書を 取る.

図2. 翻訳実験結果の計算機出力例

開発した日本語の意味構造抽出システムEXPLUSで用いたユニット間会話の考え方をそのまま利用する。また，意味記述にも，SRLがそのまま利用され予想以上の有効性が確認されている。我々は，機械翻訳システムを言語理解システムの一環としてとらえ，その場しのぎ解決策を極力排した（願わくば正統的で一般的な）解法を探っていきたいと考えている。

機械翻訳システムは，拡張LINGOLシステム[1]の中に埋め込まれて動作する。次節で述べる様に，これは，Prolog上のDCGでも大きな変更を伴うことなく記述可能である。なぜなら，拡張LINGOLとDCGとは極めて似た発想があるからであるが，最近，意味解析の制御の柔軟さの点で，Prologをベースにした DCGに優れた長所があることが明らかになってきている。そこで次節で両者の比較を行い，原理的な利害得失を論じる。

3. Prolog & DCG : Lisp & 拡張LINGOL

述語論理プログラミング言語Prologを用いて自然言語処理を行うことができる。これはDCGとよばれており，DCGで記述された文法規則は簡単なトランスレータにより，Prologのプログラムに変換される。このPrologのプログラムを実行することがトップダウンに構文解析することに相当し，各文法規則に付加された述語により，構文解析過程の制御と意味解析を行うことができる。

DCGは，筆者がLisp上で開発した拡張LINGOLと極めて似た発想がある。文献(3)で両者の比較を試みたが，その後，幾つかの修正点が明らかになり，現在では，DCGにより，これまでのシステムを全て書き変えてみようと考えている。紙面の都合で，両者の動作原理を説明するわけにはいかない。興味ある読者は文献(1,5)を参照されたい。

表1 DCGと拡張LINGOLの比較

	DCG	拡張LINGOL
A)	文法規則の記述即Prologプログラム，?- sentence（入力文）．というゴールの実行が，Prologプログラムの実行による構文解析と意味解析に直結，したがって構文解析用の特別なシステムを作る必要がないだけでなく，構文解析過程と意味解析過程の制御を，Prologの非決定論的なプログラムの実行機構に任せることができるので，オーバヘッドが少ない。	構文解析を行い意味解析過程を制御するシステムをLisp上に作成しなければならない。そのため実行に若干のオーバヘッドがある。作成労力はLispで約800行程度。
B)	ユニフィケーションによる非終端記号間での自然なメッセージ転送と，付加述語（{,}で囲まれたterm）の実行によるメッセージ検査で，文法規則の適用可能条件の判定を行う。	<advice>部に書かれたLisp関数の評価による非終端記号間でのメッセージ転送とメッセージ検査による文法規則の適用可能条件の判定。
C)	DCGによる文法記述例： n_subj(SJ) → det(DET), (adj(ADJ); []), noun(N), p(P), {member(P, [ga, wa])}.	拡張LINGOLによる文法記述例： (n_subj (@(det NIL) @(adj NIL) (noun NIL) (p (MEMBER (MH) '(ga wa)))) <cog> <sem>)
	（つづく）	（つづく）

	DCG（つづき）	拡張LINGOL（つづき）
(D)	Prologのプログラム実行制御に従うためトップダウン・縦型探索による構文解析，したがって強制バックトラックにより他の構文解析結果を得る．	ボトムアップとトップダウン法の併用と横型探索による構文解析，横型探索のため，構文解析終了時に複数個の構文解析結果が一度に得られる．
(E)	省略可能な非終端記号は，文法規則毎に指定可能なだけでなく，全文法規則にわたるグローバルな指定も可能： question → n_subj(SJ), adj(ADJ), da(?). n_subj → det(DET), (adj(ADJ); []), noun(N), p(P), {member(P, [ga, wa])}. det(sono) → [sono]. det([]) → []. 下線 ~~~~ が個別的，下線 ——— がグローバルな省略可能指定．	省略可能な非終端記号は文法規則毎に記号@で指定する： (question ((n_subj NIL) (adj NIL) (da NIL) <cog> <sem>) (n_subj (@(det NIL) @(adj NIL) (noun NIL) (p (MEMBER (MM) '(ga wa))) <cog> <sem>) 記号@で，det，adjが省略可能を指定．
(F)	構文解析過程と意味解析過程との融合を基本とする．いずれの過程も，Prologの非決定論的なプログラムの制御機構でバックトラック制御がなされる．これにより得られる利点は，一つの構文解析木から複数個の意味解析結果が容易に得られることである．（これについては，本節後半で実例を示す）	<sem>，<cog>，<advice>部の役割分担により，構文解析過程と意味解析過程との結合度合いの調節が可能であるが，両者のルーズな結合を基本とする．意味解析結果は，一つの構文解析木から一つ得られるだけで，DCGと比較して意味解析の柔軟さに欠ける．
(G)	構文解析結果(木)の優先順位付けは，文法規則の配列法により解決可能な場合もあるが一般的で柔軟な解決策はない．これは(D)項で述べた様に，縦型探索による構文解析であり，解析中途で解析結果の出力順序を制御できないことによる．	横型探索による構文解析だから，複数個の解析結果が構文解析中途でも得られており，<cog>部の起動により，それら相互間での優先順位付けが可能
(H)	誰かはやく自動分かち書きの機能を組み込んでほしい！	辞書項目と文法規則を用いた自動分かち書き機能が組み込まれている．
(I)	誰かはやく，漢字Prologを作ってほしい！	漢字Lisp（図1，図2参照）使用により，ASCII文字と同じレベルで漢字を扱うことができる(P)！

筆者等はかって，少数の構文解析結果が得られるように，文法規則の形式に工夫をこらすことの利点を指摘したことがある[10]．その時，得られた構文解析木の構造が，妥当な意味解析を妨げるものであってはならないことを述べた．この時，得られる構文解析木の数を少なくするために，一つの構文解析木に対して，複数個の意味解析結果が得られる可能性をも考慮した文

法規則の開発を行うべきことが示唆されていた。しかし，これをLispベースの拡張LINGOL上で行うためには，意味解析過程そのものを横型探索方式にするか，縦型探索としてバックトラック方式にすることが必要になる。前者は，意味解析過程を複雑なものにするし，そのために従来の意味解析法（1つの構文解析木に1つの意味解釈が対応）を大幅に変更しなければならない。後者はまさにPrologのプログラム実行機構の一部を具体化しなければならないことを意味する。これが表1(5)にまとめたことの意味である。

これに対して，DCGはPrologのプログラムとして実働化されているため，拡張LINGOLで生じた（意味解析での）問題は起きない。これは，Prologがnondeterministic programming言語であることの自然な帰結である。

ここで，1つの構文解析木から複数個の意味解析結果を得る必要性を論じてみる。やや技巧的な文ではあるが，「猫を買う女は美しい」という文は，

(i) 女が猫を(誰かに)買う

(ii) (誰かが)猫を女に買う

の両様に解釈できる（「買い与える」の方が「買う」よりも良かったかも知れない）。これは，「買う」で連体修飾される名詞「女」の助詞が，美しいの格であり，「買う」の格と何ら関係ないために，「女」が「買う」とどの様な格関係にあるかが決まらないことによる曖昧さである。この二様の解釈は意味のレベルの問題であり，構文解析の

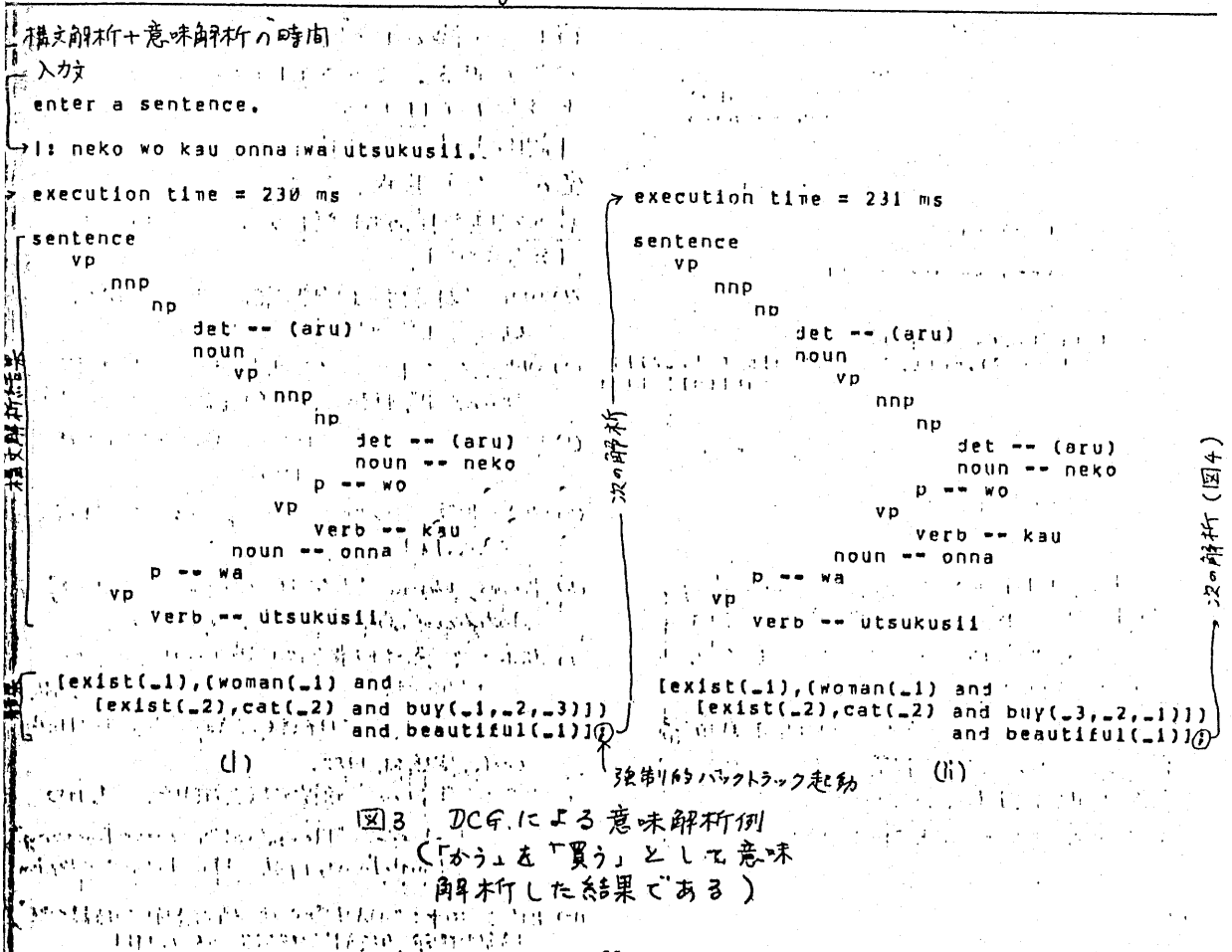

図3 DCGによる意味解析例
（「かう」を「買う」として意味解析した結果である）

レベルの問題ではない。構文解析では，「女」が「買う」に連体修飾される構造を作り出すだけで十分である。

前頁図3にDCGによる意味解析例を示す。これから，1つの構文解析木から2つの意味解析結果が得られていることが分かる。以上は「かう」を「買う（buy）」として解析しているが，実は「育てる，飼う（raise）」の意味も辞書に記述されており，第2の意味解析結果に更に強制的バックトラックを行なうと，第3の意味解析結果（図4）が得られる。この場合には，「買う」の場合のような二様の解釈は得られない。

```
execution time = 236 ms

sentence
   vp
      nnp
         np
            det -- (aru)
            noun
               vp
                  nnp
                     np
                        det -- (aru)
                        noun -- neko
                     p -- wo
                  vp
                     vero -- kau
               noun -- onna
         p -- wa
      vp
         vero -- utsukusii


[exist(_1),(woman(_1) and
   [exist(_2),cat(_2) and raise(_1,_2)])
                      and beautiful(_1)]
```

図4　「かう」を「飼う」として意味解析した結果

図3，図4の結果は，推論機構研究で開発を進めている日本語モンテギュー文法の研究成果の一端を示したものである。日本語モンテギュー文法の実働化に部分的に成功したのはこれが最初であろう。この研究では，語順が比較的自由な日本語の特性を考慮し，SRL的なフレームを導入した意味的整合性の検査を行なっている。この様な試みも，これまでのモンテギュー文法の研究者から比較的無視されてきたことのように思える。この研究については，文献(6,7)を参照してほしい。

4. 会話のモデル

ここまで書いてきて，制限頁が6頁で，学会から送られてきた原稿用紙が尽きたことに気付いた。余白も限られてきたので簡単に説明する。問題意識は文献(4)で説明した。ここではそれをまとめてみる。

知的なインタフェース・システムとして自然言語理解システムの備えるべき要件として，会話のモデルを持つ必要がある。それを利用して(A)質問者の発した質問が事実に反する仮設に基づくことを検知したらそれを質問者に教える，(B)質問者に関する知識を獲得し，知識のレベルに応じた応答をすべきである。この方向の研究は文献(9)を参照してほしい。

［謝辞］本稿の準備で，推論機構研究室の元吉文男氏，松本裕二氏，松下電器の安川秀樹氏のお世話になった。感謝します。

［参考文献］

(1) 田中："計算機による自然言語の意味処理に関する研究"，電総研研究報告797，1979.

(2) 田中，元吉，安川："意味表現用言語SRLの機械翻訳への応用"，情報処理学会自然言語処理研究会(投稿中

(3) 田中・松本："Prologと自然言語処理"，情報処理学会第23回大会，6G-1，1981.

(4) 田中・諏訪："コンサルテーション・システム"，数理科学第5世代計算機特集，4月号，1982.

(5) Pieiar, Warren："Definite Clause Grammar", Artificial Intelligence, 13, 2, 1980.

(6) 松本："意味的整合性を考慮した日本語モンタギュー文法"，自然言語の論理的分析・モンタギュー文法と関連領域，上智大

(7) 松本："Prologによる日本語モンタギュー文法"，第1回Prolog Conf.，電総研，1982.

(8) 元吉："漢字Lisp"，情報処理学会24回大会，2L-1，1982.

(9) Joshi et.al. eds.："Elements of Discourse Processing", Oxford Univ. Press，1981. 特にLehnertとKaplanの論文.

(10) 井佐原・田中："日本語埋め込み文の構文解析における諸問題"，情報処理学会，自然言語処理研究会26-4，1981.