自然言語対話のモデルと CAI への応用†

大槻説乎†† 竹内 章††

CAI コースウェアのように，取り扱う分野の範囲と内容が明確な領域を対象として，自然言語を用いて質問応答を行う場合の対話システムについて考察する．まず自然言語を用いた対話による質問応答の際に解決せねばならない問題点を提起する．この問題点を解決するために，対話システムがどのような構造と機能をもつべきかを CAI コースウェアを例にとって検討し，自然言語対話のためのモデルを構成する．次にこのモデルを実現するために用いた言語 Prolog-s の特徴とモデルとのかかわりについて述べ，最後に実現したシステムにおける DCG (definite clause grammar by F. C. N. Pereira) を基本にした自然言語処理の方法に関して報告する．付録として，このシステムを用いた文章の解析例，生成例および対話例を付記した．

1. はじめに

ここ数年，計算機科学の急速な進展と普及に伴い，CAI に関する研究や実践も全世界で著しい高まりと広がりを見せている．この流れは二つに分類できる．

その一つはいわゆる traditional CAI と呼ばれるもので，シミュレーション型，ゲーム型，演習問題型，データベース型等，多くの実践の結果定着した技術を組み合わせて実用的なコースウェアを開発利用するもので，authoring system も含めて，すでに多数の実践例が報告されている．この型の CAI は長期間にわたる経験の集積によって，現在では実用に耐えるよいコースウェアを実現できる状況になっている．しかし traditional CAI は，学習者がイニシャティブをもって自由に質問をすることができないという，教育システムとしてはかなり本質的な欠点をもっている．

他の一つの流れは intelligent CAI と呼ばれるもので，traditional CAI のもつ上記の欠点を解決しようとする一つの試みであり，知識ベースと推論機能を基本として，学習途上で生じる学習者のさまざまな疑問に柔軟に対応できることを目的として現在研究が進められており，いくつかの authoring system とコースウェアが発表されている[1)-3),5)]．この技術は将来の CAI の普及によい影響を及ぼすであろうと思われるが，現状では解決しなければならない多くの問題点がある．主要な問題点は次の二点であろう．

（1） 自然言語の理解

学習者に自由な質問を許すためには自然言語は不可欠である．最近，自然言語の解析，生成技術には長足の進歩がみられる．解析と生成に関しては，多義語やあいまいな表現を別にすれば，十分実用になる段階に至っている．多義語やあいまい性は知識ベースを用いた推論によって解決されねばならない自然言語理解の問題である．したがって自然言語の解析，生成は自然言語理解と不可分の関係にあり，知識ベースや推論機構と共通の方式で処理される必要がある．

（2） 学習者の理解度の把握と会話の主導性

学習者と自然言語で対話を行う場合，学習者の質問に答えるだけでは不十分で，対話のなかから学習者の理解度を把握して最適な学習内容を提供できるような方向づけが必要である．学習者の理解度に応じた教育目標の設定に関してはすでに多くの研究例がある[7),12)]．したがって，問題はこれらの方式と対話によって理解した内容との整合性に帰着される．

われわれは上記の問題点を解決するために，対話システムがどのような構造と機能をもつべきかを CAI コースウェアを例にとって検討し，自然言語対話のためのモデルを構成した．本論文では，最初に対話処理系のモデルと，それを実現するための記述言語について述べ，次にこのモデルの中で行われる英語に関する自然言語処理の方法について報告する．

2. 対話処理系

2.1 対話処理系のモデル

2.1.1 概念知識と言語知識の関係

知識工学で通常用いられる知識，すなわち規則や事実などを概念知識と呼ぶ．一方，自然言語処理で用いられる文法規則や単語の辞書的知識を言語知識と呼ぶ．概念知識と言語知識は次の観点から，分離して管

† A Model of Natural Language Dialogue and Its Application to CAI by SETSUKO OTSUKI and AKIRA TAKEUCHI (University Computation Center, Kyushu University).

†† 九州大学中央計数施設

理されるべきであると考える.

(1) 同じ語に対する知識でも，言語知識と概念知識とはまったく別の働きをする．たとえば，"WHY" という語の言語知識は品詞や接続関係を示すのに対して，概念知識は話題の根拠を推論する手続きを示す.

(2) 言語知識は，用いる自然言語の種類ごとに異なった文法規則と辞書を必要とするが，概念知識は使用する自然言語に共通な表現をもたせることができる．意味のある単語または熟語に対する概念知識を概念定数と呼ぶ．同義語とは言語知識のなかのいくつかの単語が概念知識のなかの同一の概念定数に対応する多対一の関係であり，多義語とは言語知識のなかの一語が概念知識のなかの多くの概念定数に対応する一対多の関係である.

(3) 言語知識の文法規則は汎用性をもち，自然言語の質問応答システムに共通に利用できるが，概念知識はシステムの主題や著者に固有な内容をもっている.

(4) 言語知識は自然言語の解析や生成の時点で必要となる知識であり，一方概念知識は自然言語解析の結果得られた概念の正当性を検証したり，不明な部分を推論によって明確化するときに必要となる知識である.

上の理由で，二つの知識は，われわれのモデルのなかでは，分離して扱われる．しかし，この二つの知識は相互に密接な関係をもっている．すなわち自然言語文は言語知識を用いて解析されて概念に変換される．この概念は概念知識を用いて行う推論のゴールとして働く必要があり，また推論によって得られた概念は，言語知識を用いて自然言語文を生成するときの条件として働かねばならない．したがって二つの知識における概念の表現形式は完全に整合していなければならない.

2.1.2 多重階層世界の導入

概念を階層世界で表現することは，ミンスキーの理論[6]以来知識工学の世界では多くのシステムで実現されている．われわれは，CAI システムにおいて，知識工学とは別の観点から，ページの階層とコースウェアの階層を導入することによって，traditional CAI の範囲でも，十分に柔軟なコースウェアを生成できることを示した[9].

ページとはコースウェアの著者がその教育方針に従って，学習者の介入なしに表示実行したい実験，説明，問題提示などの最小単位のことである．ページは著者にとってはコースウェア記述の単位であるが，システムにとってはコースウェアの編集の単位であり，学習者にとっては学習の単位となる．コースウェアの階層とは，その主題の意味に従って構成された，コースウェア全体の階層関係を表す．一方，おのおののコースウェアは目次部分に指定されたインデンテーションに応じたページの階層構造をもっている．階層の上位にあるコースウェアは下位のコースウェアを一つのページとして含んでいるので，そのページを実行することは下位のコースウェアを呼び出して実行することに等しい.

この階層世界は intelligent CAI の観点からみれば，階層の最下位の知識にはそのページに固有の知識を，上位にはそのレベルで一般化できる知識や下位の階層に共通の知識をもたせることになる．たとえば，高校物理Ⅰの力学の部分では重力の加速度や，力の単位の換算関係などは上位レベルに，例題中で与えられた物体の速度や質量は下位のページの知識に含まれる．このような世界の階層化は，自然言語理解に次のように有用な働きをする.

(1) 多義語処理

多義語の意味は，文中の他の語や文との関係や使用される環境によって決定されるが，階層化された世界のなかで多義語を定義すれば，環境や使用する単語の範囲が限定されるので多義語の意味範囲を縮小できる.

たとえば，清書システムのコマンド学習中に

Show me how to make the position of the header in the center.

と質問された場合は，多義語 make は

Show me how to assign the position of the header center.

と理解される．理由はこの世界の辞書の make の第一義が assign となっているからである.

(2) あいまい性の除去

階層世界のレベルに従って対話の環境が規定されるので，文中の陽に含まれない多くの条件が自動的に決定され，あいまい性を除くのに有効に働く．たとえばターミナルの使用法のコースウェア学習中に

Where is the power switch?

と質問されると，ディスプレイを説明するページであればディスプレイの電源スイッチの位置を，プリンタを説明するページであればプリンタの電源スイッチの位置を，両者を含む構成を説明するページならば両方

の電源の位置を示すことができる.

2.1.3 対話処理系のモデル

上記の考察から対話処理系のモデルを構成する.

図1は対話処理系の知識構造を表している. 丸く囲んだ部分は一つの世界を表す. 概念域の世界は言語域の中に一対一に対応する辞書世界をもち, すべての概念定数は言語域の対応する辞書世界のなかで単語または熟語と対応づけられている. 言語域は辞書世界のほかに, 文法規則を記述した文法世界を最上位にもっている. 知識はその属する世界が活性化されているときだけ使うことができる. ある世界が活性化されるときは, それより上位のすべての世界が, 上位から下位へ順に活性化される.

図2に対話処理の概略を示す. 言語域と概念域の二つの長方形はその時点において活性化されている世界を表す. 入力した文章は言語域において構文解析が行われ, 得られた結果は推論のゴールとして概念域に渡される. 概念域では推論を行い, 結果を自然言語で表現するために言語域に渡す. 言語域はこの結果を満たす自然言語を生成し出力する.

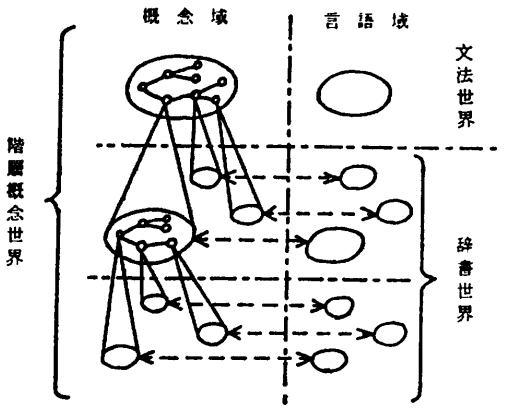

図1 自然言語対話処理のための知識構造
Fig. 1 Knowledge structure for natural language dialogue.

2.2 対話処理系の記述

知識は Prolog-s と呼ばれる言語を用いて記述される. Prolog-s はよく知られている多くの Prolog[4),8),10)] と同様な基本機能をもっているが, 前述のモデルを実現するために, 下記の二つの機能とそれに伴う組込み関数が追加されている. 現在用いている Prolog-s は Lisp の上に実現されており, Lisp の関数として Prolog-s を呼び出すことや, Prolog-s のなかから Lisp の関数を使うことができる. 本文中, 変数は第一字を大文字で, 定数は小文字で表す.

2.2.1 多重世界の実現

世界はそれぞれ固有の名前をもつ. 知識は Prolog の節で表現される. 節の述語の属性として世界名を付加することによって, 多重世界での知識の帰属を表す. 世界の階層関係自身も知識である. 多重世界に関して5種類の組込み関数がある. すなわち, 新しい世界を階層世界のなかに定義するための new__world, 現在活性化している世界を不活性化して新しく活性化する世界を指定するための init__world, 現在活性化している世界に追加して新しい世界を活性化するための open__world, 現在活性化している世界の一部を非活性化する close__world, 現在活性化している言語域(または概念域)の世界を対応する概念域(または言語域)の世界に入れ変えて活性化する change__world である.

Prolog のマッチングは, 活性化された世界のなかだけで行われる. マッチングは, 活性化された順番と逆順に行われる. 理由はいちばん最近に活性化された世界が現在の話題の中心環境と考えられるからである.

2.2.2 状態変数の導入

多重世界を効果的に運用するためには, 多重世界の間の情報の受け渡しを容易にする必要がある.

一般に, Prolog の変数は, ユニフィケーションのスコープの範囲内だけで意味をもつもので, マッチングのトップレベルを越えて値を保持する方法は, assert

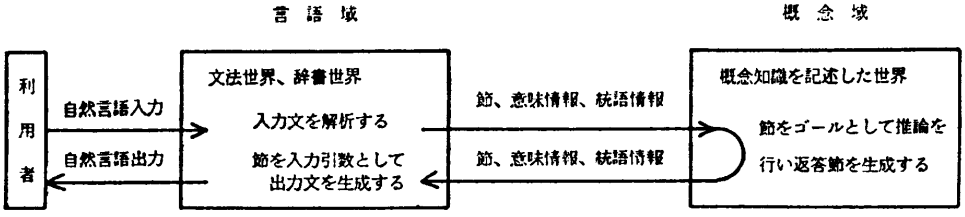

図2 対話処理の概略図
Fig. 2 Outline of the dialogue processing.

述語による節の追加だけである．これに加えて，Prolog-s には，多重世界の間で共通に情報を受け渡すための状態変数が用意されている．

状態変数は，$ で始まる名前をもち，スタックによって実現されている短期記憶（セッション継続中だけ保存される）である．Prolog-s の実行中に引用される節中に状態変数が含まれる場合は，マッチングの直前にスタックの先頭値で置き換える．ただし，quote の引数として用いられた場合と，状態変数に関する組込み関数の引数として用いられた場合は置換えは起こらない．状態変数に関する組込み関数は次の4通りある．スタックを初期化する init_state，スタックに値を入れる push，スタックから値を取り出す pop およびスタックを初期化して値を入れる set である．Init_state と set とはバックトラックによって元の値に戻ることはないが，pop と push はバックトラックが起こると，元の値に復元される．

この状態変数は，われわれの対話システムのなかで次のように使われている．

（1） 自然言語文の意味を表す．

たとえばCAIコースウェア “How to use RUNOFF system” の概念知識のなかでは

assign (top_margin 10)

はあいまいな表現であって，多くの解釈が可能である．実際この節を含む世界に対応する言語域では次のように多様な文が対応する．

Assign the top margin 10.
Was the top margin assigned 10?
Can the top margin be assigned 10?
The top margin has been assigned 10.

など．したがって，われわれの対話システムでは自然言語を解析して得られる意味とは，節や話題等の意味情報と，時制，態，法等の統語情報の集合であると定める．自然語文の解析で得た意味は状態変数によって保存され，自然言語理解と応答文生成の過程に引き継がれる．

（2） 学習経過の記録

学習者の模擬実験や問題解決の手順，利用した法則，学習者の質問の意味，コースウェアからの質問に対する回答，計算の過程や結果等は，状態変数に保存できる．この場合は，コースウェアの著者が該当する節の本体に状態変数への代入を記述しなければならない．

（3） 学習者の理解度の記録

CAIでは，教育方針に従って具体的な教授目標を設定し，構造化した到達目標と，それに対する何段階かの到達度を設定して，学習者の理解度の判定や，次に与える教育内容の選定を行う場合がある．このときの到達目標を状態変数に割り当てて置けば，（2）の学習経過の記録を用いて到達度を決定する式または手続きを与えることによって理解度を計算し，状態変数に記録することができる．

3. 自然言語対話

3.1 自然言語対話の概要

1章で述べたように自然言語対話は解析・理解・生成の3段階に分けることができる．理解に関しては，2章で述べたモデルの概念域で推論を行って結論を得ることと同義であると考える．このように考えると，対話における自然言語の解析と生成は，自然言語理解に用いるゴールや各種の状態変数の値と，この値を意味としてもつ自然語文との間の変換となる．

DCG[11] は Prolog によって表現される記述の容易な文法であり，自然言語をゴールや状態変数の集合に変換するという考え方に最も適している．したがって言語域の文法世界と辞書世界は DCG を基礎にした拡張表現を用いて記述されている．

自然言語による対話は次の3段階を繰り返して行う．

（1） 第一ステップ：学習者が自然言語文を入力すると，その時点での学習位置を表す概念の世界に対応する言語域の辞書世界と文法世界が活性化される．入力文はゴール，意味情報，統語情報の3種の状態変数に変換される．ゴールは3.3節で示す方法で生成され，状態変数 $goal に保存される．構文解析で得られた統語情報すなわち態，基本時制，完了進行時制，法助動詞，法，文の種類および副詞句または副詞節は，$voice, $tense 1, $tense 2, $modality, $mood, $statement および $adverb の値となる．

主語，目的語，名詞補語およびそれを構成する名詞と $goal の先頭値は $topics のなかに保存され，代名詞の実体を推論するときの基礎データとなる（3.2節参照）．前置詞の種類，動詞と不定詞（または動名詞）の組合せ，副詞節の接続詞の種類等によって，$adverb の値が，条件，時，場所，原因，結果，程度，目的，手段，付随，譲歩のどれを表すかが決定できる場合は，$adverb の値を $condition, $time, $place, $cause, $effect, $extent, $purpose, $means,

$belongs, $concession の該当する変数に代入する．この10個の変数と，$topics を意味情報と呼ぶ．付録1は第一ステップの例である．

（2） 第二ステップ：辞書世界に対応する概念域の世界が活性化され，言語域は不活性となる．第一ステップで生成した $goal の先頭の値をゴールとしてPrologが実行され，成功または失敗の結果を得る．成功した場合，ゴールに含まれる変数はユニフィケーションの結果に置き換えられて，$output に保存される．失敗した場合は次の処理を行う．

a）ゴールに対応する述語がない場合

自然言語の意味が全然わからないわけであるから

I can't understand what you mean.

の返答を行う．

b）ゴールに対応する述語があり，ゴールが変数を含まない場合

ゴールを否定した返答を行い，その後主導権を学習者に任せたままにするか，システム側がもつかは，著者の教育方針に従う．たとえば学習者が

Can the position of the header be assigned at the bottom?

と尋ねた場合

No, it can't be.

と返答したあと

It must be a member of (center right left).

と返答することも

Can you tell me all the valid values for the position of the header?

等と問い返して学習の主導権をコースウェアに戻すこともできる．

c） ゴールに対応する述語があって，ゴールが変数を含んでいる場合

入力疑問文を従属節に変換して I don't know に続ける．

実行の結果，概念域の新しい世界を活性化したり出力文の生成に関する統語情報（時制，態など）を書き変える必要がある場合は，引用する節の本体に記述しなければならない．話題が変わる場合は，第三ステップを終了した後に活性化すべき世界の名前を $next_world に指定する．

（3） 第三ステップ：第一ステップと同じ世界が活性化され，$output の先頭値を引数としたゴールすなわち

: —output (String [ ] $output), push ($string String)

とのマッチングが行われ，得られた自然言語文は $string に保存された後出力される．付録2は第三ステップの例である．

3.2 辞　書

辞書は言語域に属する世界として実現されており，辞書に対応する概念域の世界名に dictionary という文字列を続けた世界名をもつ．たとえば "runoff" という名前の概念域の世界に含まれる概念定数は "runoff_dictionary" という名前の辞書の中に引数として含まれている．辞書は下記の品詞をファンクタとする節の集りである．

aux_verb trans_verb intrans_verb have be do ques_word noun determinant adjective adverb conjunction preposition

名詞はさらに次の種類に分けられている．

common_noun collective_noun material_noun abstract_noun person_pronoun demo_pronoun indef_pronoun relative_pronoun propernoun person_propernoun

および概念域のテーマに応じて追加した任意個の種類．たとえば清書システムのなかでは command, variable, value が名詞の種類として追加される．

原則として辞書中の節は本体をもたないが，必要に応じて制限事項などを本体に記述できる．頭部は次の形をしている．

functor (String Word Tail Root Case (for noun)/Tense (for verb) Person Preposition Argument Protogoal)

最初の四つの引数，String, Word, Tail および Root は DCG に従う．残りの引数は3.3節に示す目的で導入されたもので，Case は格を，Tense は動詞の活用を，Person は人称と数を表す．最後の引数 Protogoal は，索引語 Word に対する概念定数が概念域のなかでファンクタとして使われているときはそのファンクタと仮引数をもつゴールの形をしており，仮引数は Argument の値となる．一方，概念定数がゴールの引数として使われているときはその概念定数が Protogoal の値となり，Argument の値は空となる．Protogoal の引数とファンクタの関係を示す前置詞は Preposition の値となる．

例1 trans_verb ([assigns|X] assigns X assign present [s3] to [Subject Object_d Object_i] assign (Subject Object_d

Object_i)) : —
n_kind (Object_d variable),
n_kind (Object_i value).

例 2 noun ([father|X] father X father [subjective objective] [s3] of [Y Z] father (Y Z)) : —n_kind (Y animal), n_kind (Z animal).

本体の名詞の種類は例1のように多義語の意味を確定したり，代名詞の値を $topics のなかから選定する場合に有効である．たとえば

What is the default value of the indentation number?
It is Zero.

という会話の際の $topics の先頭値は

[[default value of indentation number][default value] [indentation number] default_value (indentation_number What)]

となっている．この状態で

Can it be assigned 5?

と質問された場合は assign の辞書と

n_kind (indentation_number variable)

とのマッチングから it は (indentation number) を指すことがわかる．

3.3 ゴールの生成と出力文の生成

DCG によれば，入力文の解析と出力文の生成は同じ節の入力引数と出力引数を逆にした関係にある．したがって原則的には同一の規則で解析と生成が可能である．しかし，実用上入力文と出力文の間の文法レベルを違えたり，マッチングの効率を上げたり，さらに，解析の出力変数や生成の入力変数の値を概念域の節の形と一致させるために次のような拡張を行っている．

（1） 入力文と出力文の文法レベルの相違

入力文に文法間違いがあっても，解釈できる限り受け入れるだけの柔軟性をもたせることが望ましい．一方出力の場合は，文法にかなった文を生成する必要がある．すなわち，入力文法は出力文法に比べて文脈依存度を減少する必要がある．DCG では，文脈依存度は，節の引数と，その引数に関する条件節の挿入によって表されるので，たとえば

sentence : —noun_phrase (Person1), verb_phrase (Person2),
{if ($in_out=out) then (Person1*Person2 ≠0) else true}.

とすれば文章中の人称と数の一致は出力文だけに要求されることになる．ただし*は集合の積を，0 は空集合を，$in_out は解析と生成を区別する状態変数を表す．簡単のために他の引数は省略されている．

（2） マッチングの効率

DCG の本体に条件節を挿入することによって無駄なマッチングを省略して効率を上げることができる．付録2の(2)式の最初の条件節がこの例である．

（3） 概念域と言語域の整合性

言語域で得られた解析結果と概念域で用いるゴールの形，および概念域で得た出力変数と言語域で用いる文章生成の引数の形の整合性は，辞書の中の三つの引数 Preposition, Argument, および Protogoal によって保たれる．すなわちこれらの引数は原則的に次の約束によって決まる．ゴールのファンクタには動詞および名詞，形容詞に対する概念定数が用いられる．Argument には，ファンクタが完全自動詞の場合は主語と空に，不完全自動詞の場合は主語と補語に，完全他動詞の場合は主語と目的語と空に，授与動詞の場合は主語と直接目的語と間接目的語に，不完全他動詞の場合は主語と目的語と補語に，形容詞と名詞の場合は零個以上の名詞にそれぞれ対応する概念定数をこの順序で用いる．Protogoal の引数は Argument の値の部分集合であり，概念域の節の頭部の形と一致するように指定する．すなわち Argument は DCG による解析結果を辞書に引き継いで，Protogoal の引数にユニファイする役目をもつ．付録1の(4)，(8)式はこの例である．受動態の文章の場合は能動態に変換したときの文法上の働きによって上の約束が適用される．これらの引数が文中に現れる位置や働きに応じて必要となる前置詞が Preposition の値となる．これらの引数の使用例は付録2の(2)，(7)，(8)式に示されている．

4. おわりに

自然言語を使って学習者と対話をするためのモデルと，それを実現するための記述言語およびそれを用いた対話の方法について報告した．このような対話機能をもつコースウェアをつくるための作成者の負担は，traditional CAI に比べて知識を作成する部分だけ増加する．知識のなかの言語域は，概念域ができればほとんどの部分が機械的に生成できるが，概念域はコースウェアの著者が漏れなく，しかも矛盾なく記述しなければならない．したがって著者の負担を軽減するた

めには，traditional CAI の画面エディタやテキストエディタと同様に知識ベースエディタも authoring system のなかで重要な役割を果たすようになるであろう．このエディタは，自然言語文，式，表，節およびプログラムで入力した知識を Prolog-s に変換して概念域のなかの世界に記録すると同時に，言語域のなかに対応する辞書世界を生成する機能をもつ．われわれは，現在，このような働きをする知識エディタを CAI authoring system の中に実現し，研究中である[13]．

参考文献

1) Brown, J. S., Burton, R. R. and Bell, A. G.: SOPHIE, a Step Toward Creating a Reactive Learning Environment, *Int. J. Man-Mach. Stud.*, Vol. 7, No. 5, pp. 675-696 (1975).
2) Carbonell, J. R.: AI in CAI, *IEEE Trans. Man-Mach. Syst.*, Vol. MMS-11, No. 4, pp. 190-202 (1970).
3) Clancey, W. J.: Tutoring Rules for Guiding a Case Method Dialogue, *Int. J. Man-Mach. Stud.*, Vol. 11, No. 1, pp. 25-49 (1979).
4) Kowalsky, R.: Predicate Logic as Programming language, Proc. IFIP '74 (1974).
5) Melle, W. V.: Mycin, a Knowledge-Based Consultation Program for Infectious Disease Diagnosis, *Int. J. Man-Mach. Stud.*, Vol. 10, No. 3, pp. 313-322 (1978).
6) Minsky, M.: A Framework for Representing Knowledge, in Winston, P. H. (ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York (1975).
7) 永野和男：学習診断に対する教師の判断基準の抽出とプロダクションシステムの適用，*J. CAI*, Vol. 2, No. 2, pp. 3-10 (1982).
8) Nakasima, H.: *Prolog/KR User's Manual*, Dept. of Mathematical Eng. and Instr. Phys., University of Tokyo, Tokyo (1981)
9) Otsuki, S. and Takeuchi, A.: A Unified CAL System for Authoring, Learning and Managing Aids, Proc. IFIP WCCE '81 (1981).
10) Pereira, L. M., Pereira, F. C. N. and Warren, D. H. D.: User's Guide to DEC-10 Prolog, (1979).
11) Pereira, F. C. N. and Warren, D. H. D: Definite Clause Grammars for Language Analys is, *Artif. Intell.*, Vol. 13, No. 3, pp. 231-278 (1980).
12) 佐藤隆博：ISM 法による学習要素の階層的構造の決定，日本教育工学雑誌，Vol. 4, No. 1, pp. 9-40 (1979).
13) Takeuchi, A. and Otsuki, S.: CAI System with Natural Language Dialogue, Proc. Compcon Fall '83 (1983).

(昭和58年10月11日受付)
(昭和59年1月17日採録)

付録 1 "Show me how to assign it 13." を解析して，assign ([ ] indentation_number 13) を生成する例
<=> の右は規則を，左は規則にマッチングした節を示す．

```
                                                    (1)
input([show me how to assign it 13] [] C)           <=> input(C)->
                                                        sentence(Tense1 Tense2 Voice Mood Modality Negation
                                                                 Adverbs C),
                                                        {put_allstate(Tense1 Tense2 Voice Mood Modality
                                                                      Negation Adverbs)}.
                                                    (2)
sentence([show me how to assign it 13] R present    <=> sentence(present [] V imperative [] N [] C)->
        [] V imperative [] N [] C)                      verb_phrase(present [] V [] N Person [] C).
                                                    (3)
verb_phrase([show me how to assign it 13] R         <=> verb_phrase(T [] active [] affirmative Person Subject C)->
            present [] active [] affirmative            trans_verb(W Root T Person Prep
            Person [] C)                                           [Subject Object_d Object_i] C),
                                                        noun_phrase(Person1 Case1 Object_d),
                                                        noun_phrase(Person2 Case2 Object_i),
                                                        {if($in_out=out)then member(objective Case1),
                                                                             member(objective Case2)
                                                                        else true}.
                                                    (4)
trans_verb([show me how to assign it 13] show       <=> trans_verb(show show present [s1 p1 s2 p2 p3] to [S X Y]
           [me how to assign it 13] show present            Y).
           [s1 p1 s2 p2 p3] to [[] X Y] Y)
                                                    (5)
noun_phrase([me how to assign it 13] R Person       <=> noun_phrase(Person Case C)->
             Case C)                                    noun(W Root Case Person [] [] C).
                                                    (6)
noun([me how to assign it 13] me                    <=> noun(me i [objective] [s1] [] [] []).
       [how to assign it 13] i [objective] [s1]
       [] [] [])
                                                    (7)
noun_phrase([how to assign it 13] R [s3]            <=> noun_phrase([s3] [subjective objective] how(C Ans))->
            [subjective objective] how(C Ans))          [how to ],
                                                        verb_phrase(present [] Voice [] affirmative Person [] C).
                                                    (3)
verb_phrase([assign it 13] R present [] active      <=>
            [] affirmative Person [] C)
                                                    (8)
trans_verb([assign it 13] assign [it 13] assign     <=> trans_verb(assign assign present [s1 p1 s2 p2 p3] to
           present [s1 p1 s2 p2 p3] to [[] X Y]                 [S X Y] assign(S X Y))->
           assign([] X Y))                              {n_kind(X variable),n_kind(Y value)}.
                                                    (5)
noun_phrase([it 13] R Person Case C)                <=>
                                                    (9)
noun([it 13] it [13] it [subjective objective]      <=> noun(it it [subjective objective] [s3] [] [] X)->
       [s3] [] [] [indentation_number])                 {topics(it X)}.
                                                   (10)
noun_phrase([13] [] [s3] [subjective objective]     <=> noun_phrase([s3] [subjective objective] X)->
            13)                                         {number(X)}.
```

付録 2 "assign ([ ] top_margin 5) から "5 was assigned to top margin" を生成する例

```
$tense1=past $tense2=[] $voice=passive $mood=statement $modality=[] $negation=affirmative $in_out=out
$subject=5 $adverb=[]

output(String [] assign([] top_margin 5))

sentence(String [] past [] passive statement
        [] affirmative []
        assign([] top_margin 5))

trans_verb([assign|R0] assign R0 assign
          present [s1 p1 s2 p2 p3] to
          [[] top_margin 5]
          assign([] top_margin 5))

noun_phrase([5|R1] R1 [s3] [subjective
           objective] 5)

be_part(R1 R2 past affirmative Person)

be([was|R2] was R2 be past [s1 s3] [] [] [])

trans_verb([assigned|R3] assigned R3 assign
          pp [s1 p1 s2 p2 s3 p3] to
          [[] top_margin 5]
          assign([] top_margin 5))

preposition([to|R4] to R4 to [] [] [] [] [])

noun_phrase(R4 [] Person Case C)

noun([top margin|[]] [top margin] []
     [top margin] [subjective objectibe]
     [s3] [] [] top_margin)
```

```
(1)
  <=> output(C)->sentence($tense1 $tense2 $voice $mood
                          $modality $negation $adverb C).
(2)
<=> sentence(Tense1 [] passive statement [] Negation [] C)->
    {if ($in_out=out) then
        trans_verb(St W Re Root present Ps Pr [S X Y] C)
                   else true},
    {if ($in_out=out),($subject~=[]) then (Y=$subject)
                                     else true},
    noun_phrase(Person1 Case1 Y),
    {if ($in_out=out) then member(subjective Case1)
                   else true},
    be_part(Tense1 Negation Person),
    {if ($in_out=out) then (Person1*Person~=0) else true},
    trans_verb(W3 Root pp P1 Pr [S X Y] C),
    preposition(Pr Pr [] [] [] [] []),
    noun_phrase(Person2 Case2 X),
    {if ($in_out=out) then member(objective Case2)
                   else true}.
(3)
<=> transe_verb(assign assign present [s1 p1 s2 p2 p3]
                to [S X Y] assign(S X Y))->
    {n_kind(X variable),n_kind(Y value)}.
(4)
<=> noun_phrase([s3] [subjective objective] X)->
    {number(X)}.
(5)
<=> be_part(Tense affirmative Person)->
    be(W be Tense Person [] [] []).
(6)
<=> be(was be past [s1 s3] [] [] []).
(7)
<=> transe_verb(assigned assign pp [s1 p1 s2 p2 s3 p3] to
                [S X Y] assign(S X Y))->
    {n_kind(X variable),n_kind(Y value)}.
(8)
<=> preposition(to to [] [] [] [] []).
(9)
<=> noun_phrase(Person Case C)->
    noun(W Root Case Person Pr A C).
(10)
<=> noun([top margin] [top margin]
         [subjective objective] [s3] [] []
         top_margin).
```

付録 3 自然言語対話の例

```
> Show me how to assign the header position center.
  Use command .HEADER with the value of the 1st parameter c.
> Why is the top margin 6?
  The value of the top margin was not defined,
  so the default value of 6 is valid.
> What is the value of the indentation number before the phrase "In a sense"
> in the file "F1289.CAI57.TEXT"?
  The value of the indentation number is 10.
> Why isn't it 13?
  The following sequence of values were assigned in the order printed below.
  The value was changed to 13 by the command ".INDENT 13" before the phrase
  "2. CHARACTERISTICS".
  The value was changed to 10 by the command ".LEFT 3" before the phrase
  "To fulfil the second purpose".
> Show me how to assign it 13.
  Use command .INDENT with the value of the 1st parameter 3.
> Show me how to assign the header position 5.
  5 dosen't satisfy the following conditions; .
  value of the header position must be a member of (left center right
  alternative).
```