

状況意味論に基づく談話理解システムDUALS

----- そのインプリメンテーション -----

三古秀夫, 安川秀樹^{*1)}, 平川秀樹^{*2)}, 向井国昭, 田中裕一, 横井俊夫
(財団法人 新世代コンピュータ技術開発機構)

^{*1)} 現在 松下電器産業株式会社

^{*2)} 現在 株式会社東芝 総合研究所

1. 概要

最近の自然言語処理技術の進歩は目を見張るものがあるが、談話理解のレベルになると困難な問題も多く、未だ研究段階の域を出ていない。自然言語による計算機との柔軟な対話機能を実現するためには談話理解技術は不可欠の要素であり、知的インタフェースの要となるものである。ICOTでは当初より状況意味論(Situation Semantics)[1] に基いた意味理解、談話理解の研究を行っており、今回その基本原理に基いた、談話理解のための実験システムDUALS(Discourse Understanding Aimed at Logic-based Systems)[2]の開発を行った。状況意味論はJ. Barwise とJ. Perryにより1980年頃より発表されて以来、自然言語処理あるいは言語学などの分野で注目を集めている理論である。事実の有限の集りとしての部分世界を状況としてとらえ、この状況を意味における基本的実体とする。このアイデアは言語理解の計算モデルに対しても画期的な基礎付けをもたらすものと期待されている。本稿ではこのDUALS 第1版のインプリメンテーションについて述べる。DUALS の最初の例題として取上げた文章は小学校3年生用の国語の読解力問題の中から選んだ。問題は最初にストーリーが与えられている。ストーリーは、エンジン・トラブルが発生した旅客機の中で

機長とスチュワーデスが協力して冷静に対処するという内容である。そのあとに、「機長はなぜはっとしたのですか。」のような設問が続く。

2. モジュール構成

DUALS のモジュール構成を図 2.1に示す。構文・意味解析部は、与えられた文章の各文を文法と単語辞書を参照しながら構文解析して、各文が表す状況を記述する論理式に変換する。文法記述はLFG(Lexical Functional Grammar)[3]に基いている。出力の論理式により記述されている状況はイベント・タイプと呼ばれる。出力論理式は代名詞を含んでいる。また動詞の主格や目的格が欠けていることもある。これらの確定していない情報を決定するのが次のオブジェクト同定部である。このオブジェクト同定部の出力は記述状況(described situation)と呼ばれる。記述状況はまさに文の解釈のことである。談話解析部はこの記述状況を談話構造とよぶメモリに記憶する。もし記述状況が質問文を意味しているならば、質問応答部は談話構造メモリと知識ベースを参照しながら解答を探し、応答文を生成して表示する。知識ベースは状況意味論というタイプと制約条件の形で用意する。知識ベースの内容はさまざまである。

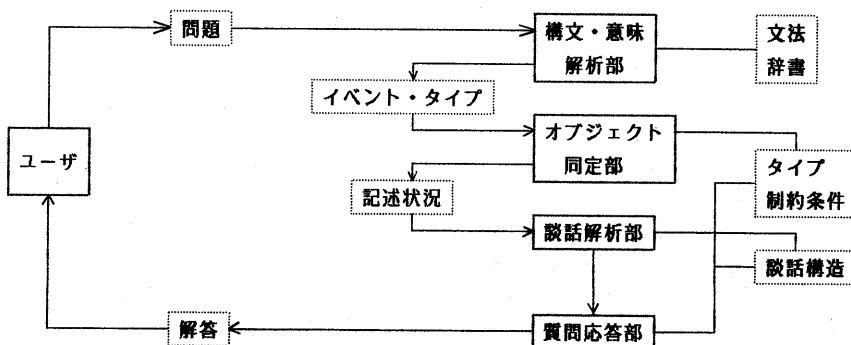


図 2.1 DUALS の構成

3. 構文・意味解析部

本章ではDUALSの構文解析、意味解析の手法について述べる。構文解析は既存のDCG[4]及びBUP[5]を利用している。またDUALSで扱う日本語の文法記述はLFGに基いている。意味表現としては状況意味論のタイプシステムを利用しており、意味解析では入力文の意味を表現する状況等を求める。意味解析は構文解析中に行われる。

3.1 構文解析

構文解析はLFGに基いて記述された文法に従って、既存のDCG及びBUPにより行われる。LFGのPrologへのインプリメントは既に報告しており[6,7],詳細はそちらを参照されたい。基本的な設計思想は、いわゆるユニフィケーション文法と同じであり、名詞句、動詞句といった文法カテゴリは単なるシンボルではなく、様々な属性(feature)とその値からなる属性の束(feature-bundle)と考え、文法的な現象をそのような属性の束のユニフィケーションによって記述しようというものである。

次にDUALSで用いている日本語文法のいくつかの特徴について述べる。主な特徴は、日本語の文をフラットな構造として扱っている点、格の属性を助詞の表層を用いて分類している点などが挙げられる。

(a) 文の構造

DUALSでは日本語文の構造をフラットのものとして扱っている。簡単な場合については、次のルールで示される。

$$(I) \quad S \rightarrow PP^* V \\ (\uparrow(\downarrow pcase)) = \downarrow \quad \uparrow = \downarrow$$

ここでSは文、Vは動詞を表す。PPは助詞を伴った名詞句、例えば「太郎が」といった構文要素を表す。このルールは、文は任意個(0を含む)のPPの並びに続いて動詞が現れるという構造からなることを示す。例えば、この文法で生成される「太郎が本を読む。」という文のc-構造とf-構造は図3.1と図3.2ようになる。

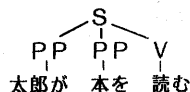


図 3.1 フラットな文の構造

フラットな構造を仮定した主な理由は、日本語の自由な語順に対応するためである。日本語では、格要素は、主動詞前の任意の位置に出現し得るため、階層的な構造とくにVPを仮定すると、複雑な解析が必要となる。フラットな構造と述語項構造を導入することにより、語順の問題は、比較的自然的に解決できる。前出の文法規則を一般化すると次のようになり、DUALSでは、文をこの規

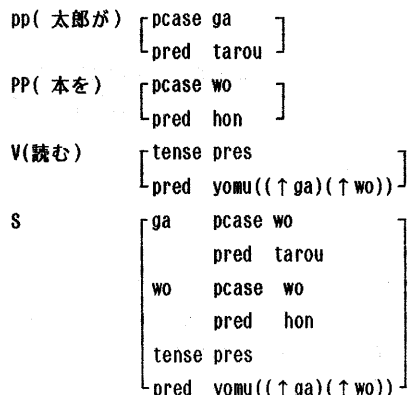


図 3.2 f-構造とユニフィケーション

則によって生成している。

$$(II) \quad S \rightarrow X^* V''$$

Xは格要素(PP), 副詞(ADV), あるいは補文(S')。すなわち任意個の格要素, 副詞, 補文が, 主動詞の前に自由な順番で現われる(V''はいわゆる述語を示す)。BUPでは任意個の並びを示す*や非終端記号に対する変項Xのような記述を許していないため、これに対する実際の文法規則は次のようになる。

$$(III) \quad \text{snuc}(\text{snuc}(\text{Stvp}), \text{Snuc}) \Rightarrow$$

$$\text{vp}(\text{Stvp}, \text{Vp}) : \text{Snuc} = \text{Vp}.$$

$$(IV) \quad \text{snuc}(\text{snuc}([\text{Stpp} \mid \text{Stsnuc1}]), \text{Snuc}) \Rightarrow$$

$$\text{pp}(\text{Stpp}, \text{Pp}) : ((\text{pcase}, \text{Pp}), \text{Snuc}) = \text{Pp} \ \& \ \text{snuc}(\text{snuc}(\text{Stsnuc1}), \text{Snuc1}) : \text{Snuc} = \text{Snuc1}.$$

$$(V) \quad \text{snuc}(\text{snuc}([\text{Stpp} \mid \text{Stsnuc1}]), \text{Snuc}) \Rightarrow$$

$$\text{pp}(\text{Stpp}, \text{Pp}) : \text{Pp} \langle \text{adjunct}, \text{Snuc} \rangle \ \& \ \text{snuc}(\text{snuc}(\text{Stsnuc1}), \text{Snuc1}) : \text{Snuc} = \text{Snuc1}.$$

$$(VI) \quad \text{snuc}(\text{snuc}([\text{Stadv} \mid \text{Stsnuc1}]), \text{Snuc}) \Rightarrow$$

$$\text{adv}(\text{Stadv}, \text{Adv}) : \text{Adv} \langle \text{adjunct}, \text{Snuc} \rangle \ \& \ \text{snuc}(\text{snuc}(\text{Stsnuc1}), \text{Snuc1}) : \text{Snuc} = \text{Snuc1}.$$

$$(VII) \quad \text{snuc}(\text{snuc}([\text{Stscomp} \mid \text{Stsnuc1}]), \text{Snuc}) \Rightarrow$$

$$\text{scomp}(\text{Stscomp}, \text{Scomp}) : (\text{xcomp}, \text{Snuc}) = \text{Scomp} \ \& \ \text{snuc}(\text{snuc}(\text{Stsnuc1}), \text{Snuc1}) : \text{Snuc} = \text{Snuc1}.$$

これらの文法規則では文に対してsnucというカテゴリを割当てているが、これは、これらの文法から生成されるのが文主要部、すなわち法(modal)の助動詞や終助詞をとらないものであることを示している。このように文と文主要部を区別した理由は、補文構造はmodalや終助詞をとらないことがあるためである。結局、文は次のルールにより生成される。

(VII) $S \rightarrow (Sadv) \& (Snuc) \& (T) \& (Modal) \& (Sp)$
 Sadvは文副詞, Tは時制を表す助動詞(る, た), Modalは法の助動詞, Spは終助詞であり, ()は省略可能であることを示す。文法規則(III)のvpは動詞句ではなく, 動詞類, いわゆる述語である。vpは動詞, 形容詞などと助動詞が結合したものなどからなる。

(b) 名詞句の構造

名詞句を構成する基本的な構文要素としては, 一般名詞, 固有名詞, 代名詞がある。すなわち,

(IX) $np(np(Stn), Np) \Rightarrow n(Stn, N) : Np = N.$

(X) $np(np(Stname), Np) \Rightarrow$
 $name(Stname, Name) : Np = Name.$

(XI) $np(np(Stpro), Np) \Rightarrow$
 $pro(Stpro, Pro) : Np = Pro.$

という規則が与えられる。これらの基本的な項目と埋込み文から複合名詞が構成される。複合名詞は, 関係節化されたもの, 同格名詞句, 名詞化構造を扱う。例えば, 「これを発見した機長のロールさん」という関係節化名詞句は次のようなルールによって生成される。

(XII) $nom(nom(Sts, Stnp), Nom) \Rightarrow$
 $s(Sts, S) : S < (adjunct, Nom) \&$
 $nom(Stnom, Nom1) :$
 $Nom = Nom1, Nom1 = controller(pp).$

(XIII) $nom(nom(Stn), Nom) \Rightarrow n(Stn, N) : Nom = N.$

(XIV) $np(np(Stnom), Np) \Rightarrow$
 $nom(Stnom, Nom) : Np = Nom.$

文法規則(XII)が主名詞句が埋込み文の格を支配する場合で, 格支配の関係がunification $\downarrow = \downarrow_{pp}$ で記述されている。nomは名詞類を示し(XIV)よりnpを生成する。その他, 「機長のロールさん」, 「ふちがみさんの顔」, 「50人の乗客」, 「いざという時の用意」のように限定辞(下線部)を含む複合名詞は次の文法規則で生成される。

(XV) $np(np(Stdet, Stnom), Np) \Rightarrow$
 $det(Stdet, Det) : (adjunct, Np) = \&$
 $nom(Stnom, Nom) : Np = Nom.$

(XVI) $det(det(Stnp, p(no)), Det) \Rightarrow$
 $np(Stnp, Np) : Det = Np \&$
 $[no] : (spec, Det) = no.$

すなわち, 名詞句と格助詞「の」から限定辞が構成される。spec属性は, noでマークされた名詞句が, 限定辞として機能していることを示す。

DUALSの対象とする文章中の特徴として, 「お客様にお知らせしますか。」のような会話文を含む。これを正しく理解するためには, その文の話者と聞き手を正しく

認識し, 語の意味を同定する必要がある。DUALSでは, 直接話法の文は, 構文・意味解析部では解析は行わず, 引用符に囲まれた文をそのままオブジェクト同定部に渡し, オブジェクト同定部がその環境下で, 引用された部分を解析するため, 構文・意味解析部を再帰的に呼出す。前出の会話文は図3.3のような解析をされる。

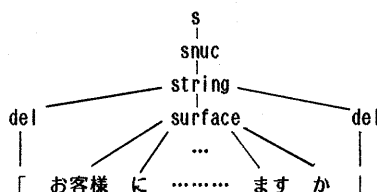


図 3.3 会話文の解析

3.2意味解析

[8]で述べたように, 状況意味論では, 文の意味を, 「その文の発話された状況と, その文の発話によって記述された状況間の関係」と定義している。すなわち単純には, 次式で表される。

$d, c \llbracket \phi \rrbracket e.$

dは談話状況(discourse situation)であり, 文 ϕ が発話された場所, 時, 話者, 聞き手などの情報を含む。cは結合状況(connective situation)であり, 話者がどのような表現で何を指しているかという情報などを含む。

eは, 文 ϕ がd, cの下に記述する状況であり, 記述状況(described situation)と呼ばれる。従って文の意味は状況eと文の発話によって決まるd, c, eに関する条件(condition)からなるタイプを考える。DUALSではこのようなタイプを表すのに構文解析で用いたf-構造を用いて次のように表す。

type	X
location	L
individual	A
condition	$in(X, at(L), (speaking, A), 1)$

すなわちタイプの不定項を属性type, 条件を属性conditionの値とし, パラメータはその名前を属性名とする属性の値として表される。各語彙項目, 文法規則中の非終端記号などの構文要素は, その意味記述をこのような形式でf-構造中に持っている。意味解析は構成的意味論(compositional semantics)に従っており, 句構造に従い, 部分の意味から全体の意味を合成する。つまり, 文法規則中の子ノードの持つタイプの表現から親ノードのタイプの表現を合成する。その際に用いる基本的なメカニズ

ムは構文解析と同様ユニフィケーションである[9]。例えば、「ロールさんがはっとする。」という文の意味を合成する際、pp“ロールさん”とv“はっとする”の意味がそれぞれ次のように与えられているとする。

```
type ind#rollsan
cond {in(cs,ld,(refers,ad,ind#rollsan,
        rollsan,name),pol#1)
      in(ds,ld,(otoko,ind#rollsan),pol#1)}
```

```
type sit*Sit
rel type hattosuru
loc type loc*Loc
pol type pol#1
ga type ind*Ind
cond {in(sit*Sit,loc*Loc,(hattosuru,ind*Ind),
        pol#1)
      in(w,lu,(t_overlap,loc*Loc,ld),pol#1)}
```

以上の2つのタイプ表現から次の文法規則を用いて意味の合成を行う。

```
s(Pp,V),Ids =>
  pp(Pp,Idpp):((pcase,Idpp),Ids) = Idpp,
              (cond,Ids) = (cond,Idpp) &
  v(V,Idv) : Ids = Idv.
```

その結果、次のような意味が得られる。

```
type sit*Sit
rel type hattosuru
loc type loc*Loc
pol type pol#1
ga type ind*Ind
cond {in(cs,ld,(refers,ad,ind#rollsan,
        rollsan,name),pol#1)
      in(ds,ld,(otoko,ind#rollsan),pol#1)}
cond {in(sit*Sit,loc*Loc,(hattosuru,ind#rollsan),
        pol#1)
      in(w,lu,(t_overlap,loc*Loc,ld),pol#1)
      in(cs,ld,(refers,ad,ind#rollsan,
        rollsan,name),pol#1)
      in(ds,ld,(otoko,ind#rollsan),pol#1)}
```

意味解析部は以上のようなタイプ表現の他に、オブジェクト同定部が必要とする次のような情報を抽出する。「文番号、文、文の種類（平叙文か疑問文か）、不定項、条件、主文(matrix sentence)のタイプ不定項、オブジェクトリスト、質問の焦点（疑問文の場合）」

4. オブジェクト同定部

本章ではDUALSのオブジェクト同定処理[10]について述べる。オブジェクト同定とは文中に現れるオブジェクト（通常の名詞だけでなく省略や照応も含む）を文脈中のオブジェクトと関連付けることである。これは状況意味論の用語で言えば、「文と文脈から新たな資源状況(resource situation)と話者のコネクション(speaker's connection)を形成する」となる。以下、DUALSに於けるオブジェクト同定処理の概要と各種ヒューリスティクスについて述べる。

4.1 モデル

オブジェクト同定部は、現在の文脈、文脈上のオブジェクト、知識及び入力された文を対象として動作して、新たな文脈とオブジェクトを出力する(図4.1)。これらは、次の入力文と共に次のオブジェクト同定処理の入力となる。オブジェクト同定処理は、文中の表現が示すものに関する仮説の生成と、その仮説が一貫性があるかどうかの検証からなり、基本的に縦型探索で行われる。

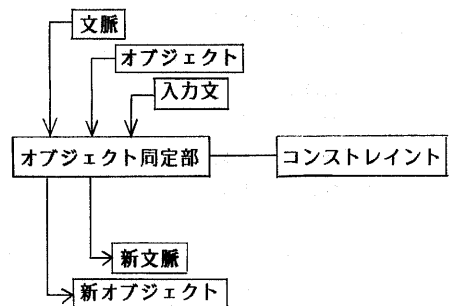


図 4.1 オブジェクト同定サイクル

4.2 仮説生成

本モジュールは、まず統語情報に関するフォーカシングメカニズム[11]に基いて、名詞句と関連付けられるべき候補を選び出す（仮説生成）。以下では、本モジュールに導入された仮説生成ヒューリスティクス等について述べる。ここでは日本語の名詞句の用法を次のように分類している[12]。

総称的(generic)

非総称的(non-generic)

不定(indefinite)

定(definite)

文脈参照的(referential)

属性参照的(attributive)

機能的(functional)

本モジュールでは名詞句の表層的な形態情報によって、その名詞句が示すオブジェクトの仮説の生成方法を定めている。現在、名詞句表現は次の4つに分類される。

- (a) 普通名詞
- (b) 名前
- (c) 0 代名詞
- (d) 代名詞

以下、各名詞形態に関する仮説生成のヒューリスティクスについて述べる。

(a) 普通名詞

普通名詞は一般的に総称的な用法、不定の用法、定の用法のいずれかをとりうる。しかし構文レベルではどの用法であるかを判定するのは困難なことが多い。そのため次のような強いヒューリスティクスを導入している。

- H1 物語に於いては、総称的な用法は使用されない。
- H2 文脈中に普通名詞の示すオブジェクトとして、真性のあるものがあればそれを仮説として優先する。
- H3 H2が適用されない場合についてのみ不定の用法であるとする。

(b) 名前

現実世界では、おなじ名前で異なったオブジェクトを示す場合、逆に複数の名前で同じオブジェクトを示す場合の両方があるが、DUALS では次のヒューリスティクスを導入している。

- H4 物語において、名前はユニークな個体を示し、また名前が異なれば個体も異なる。

普通名詞に関するヒューリスティクスH2, H3 は名前に関しても成立する。

(c) 0 代名詞

0 代名詞は「省略された名詞句」のことであり、0 代名詞が頻繁に使用されるのは日本語の特徴の 1 つである。0 代名詞の用法は次のように不定、文脈参照の 2 つの用法が考えられる。

文脈参照用法：太郎は花子を見た。(0は) 本を読んでいた。

不定用法：(0が) サンマを焼くにおいがした。

ここでは、0 代名詞に関して、次のヒューリスティクスを導入している。

- H5 0 代名詞は基本的に文脈参照用法を取る。文脈参照の可能性が無い時に限り不定の解釈とする。

また 0 代名詞の指示する内容に関しては、次のヒューリスティクスを導入している。

- H6 0 代名詞は、個体 {individual} またはその集合 {set} を示す。

0 代名詞については、統語構造に基くフォーカス機能を

導入して処理を行っている。ここでは Kameyama の Zero-Pronoun Interpretation Rule[13] を拡張したメカニズムを用いている。Kameyama はフォーカスに類似した CENTER という概念を導入しており、0 代名詞の解釈は次のアルゴリズムで進む。

- (i) ある文 S1 に対して、Expected Center Order 規則を適用して、次の文 S2 に対する CENTER の候補を文 S1 から順序付けして抽出する。

- (ii) 文 S2 を入力し、CENTER Establishment 規則により、

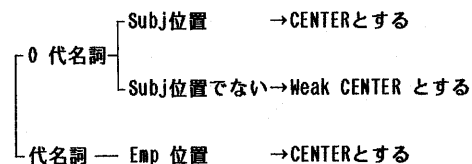
- (i) で作られた候補と S2 から文 S2 の CENTER を決定する。

文 S2 に対して (i) を適用する。

Expected Center Order 規則とは、「文の中の Cf (forward-looking center) に関して次の文の Cb (backward-looking center) の候補としての暗黙の優先付けが存在する。」という仮説に基いた規則で次のように定義される。ただし、Cf は通常その文に現れる名詞句の示す entity, Cb はその文の CENTER である。

- 1) TOPIC
- 2) SUBJECT
- 3) OBJECT (2)
- 4) others

CENTER Establishment 規則は、文の Cb を決定する規則であり、次のように定義されている。



この読み方は、例えば上方向だけだとすると、「0 代名詞が Subj 位置にあれば、その 0 代名詞が示す Cf (前の文に対して Expected CENTER Order 規則を適用して得られる) をその文の CENTER とする。」である。Weak CENTER とは、Cf のオーダリングとしてその文の Cb の次にあげられるものである。

その他、1 つの文に 2 つ以上の 0 代名詞が使用される場合に関する Multi Pronoun 規則がある。

- ・ 1 つの文に 2 つ以上の 0 代名詞が存在する場合に、Emp 位置が文の CENTER となる。ただし、Emp 位置とは、Emphathy position であり、通常の動詞では Subj 位置になる。Emp 位置が明確に現れるのは、次のような場合である。

- あげる、もらう → Subj 位置
- くれる → Obj2 位置

またDUALS ではMultiple Pronounに関して、次のヒューリスティクスを導入する。

H7 // Multiple Pronounは異なるオブジェクトを指示する。

なお、Kameyamaのアルゴリズムは動詞を含む句単位に実行されるので次のような文は扱うことができない。

太郎は学校は行って／(0が) 友達と会った。
DUALS が実際に扱う文章には複文や埋込み文が含まれている。そこで次のヒューリスティクスを導入している。

H8 // 1つの文におけるCfは、主文の主述語(main pred-icate)の格要素、及びそのPOSS(所有格)位置にあるもののみに限定する。

このようなヒューリスティクスにより、上記の例では、次のようにCfが作られる。

太郎は学校へ行って友達と会った。
[Cf1=太郎, Cf2=友達]

(d) 代名詞

代名詞には、性質上総称的な用法はない。また、不定の用法は非常に特殊な場合に限られ、通常は文脈参照的に用いられる。このため、DUALS では代名詞については次のヒューリスティクスを導入している。

H9 代名詞は文脈参照用法としてのみ使用される。
ここで文脈参照用法というのは、その代名詞の発話時における文脈という意味であり、次例のように代名詞が現われている文自体も文脈の一部となり得る。

太郎が愛した女性は彼に話しかけた。
代名詞の同定におけるフォーカシングについては、0代名詞と同様のメカニズムを用いて仮説生成のオーダリングを行っている。0代名詞の処理では、H7により、主文の要素のみを対象としていたが、代名詞の場合は、文脈に登場するオブジェクト全体を候補とする。例えば、次の文で、「彼」は埋込み文中のオブジェクトを示す。

太郎が音楽を学んだ学校は赤羽橋にあった。

彼はそこでピアノを習った。

代名詞と0代名詞は、それらが指示できる要素において差がある。0代名詞は、文脈中の個体(individual)のみを示すが、代名詞は次の例のように、ある出来事(event)又は状況意味論の用語ではsituation)を示すことができる。

どうした事が急にエンジンから白い煙が吹出しました。
これを発見した機長のロールさんははっとしました。

(これ=エンジンから白い煙が吹出したこと)
このため、代名詞の仮説の生成については、オブジェクト及び出来事の両方を仮説の対象としている。

4.3 仮説検証

仮説検証部は、仮説生成部で作られたオブジェクトの同定仮説が、与えられたコンストレイントに関して一貫性があるか否かを検証する。現在の版では、コンストレイントは関係(relation)のアーギュメントのタイプ制限のみに限定している。これは言語処理に於ける意味素性のチェックに対応している。次は例題に現れる「吹き出す」という関係のタイプの定義である。

type-definition(吹き出す, Self, (流体?, 具象物?), Env, Cond)

この記述は、「吹き出す」のタイプは「流体」と「具象物」をパラメータとして持ち、その定義はCondの部分に書かれることを表す。Selfにはis-aの階層を記述する部分、Envには実行時の計算環境を記述する。例えば、「エンジンから白い煙が吹き出す」という表現に対しては、次の式が仮説として生成される。

in(sit#S1, loc#L1, (吹き出す, 煙# 煙1, エンジン# エンジン1), pol#1)

in(sit#S1, loc#L1, (白い, 煙# 煙1), pol#1)

この場合、「流体」タイプと「煙」タイプ、及び「具象物」タイプと「エンジン」タイプの整合性チェックが行われる。このチェックはいわゆるシソーラスを用いたconsistency checkに相当する。

4.4 会話文の処理

DUALS のオブジェクト同定処理の特徴の1つとして会話文の処理がある。文章中に現れる会話のオブジェクト同定処理では、その発話者が誰であるかに応じて同定処理を変える必要がある。例えば、「私」という表現が会話文中に現れた場合には、その文の発話者と「私」とを同定する必要がある。このような処理のために、オブジェクト同定部は構文/意味解析部と自分自身を再帰的に呼出す機能を有している。図4.2は会話処理の様子を示す。DUALSでは先生が生徒に問題を与えるモデルを想定しているため、トップレベルの発話は全て先生の発話としている。トップレベルの文の中に会話が現れると、オブジェクト同定部は話者を先生から登場人物へとスイッチする。この操作をdiscourse situationのスイッチと呼ぶ。ここで問題になるのは、オブジェクトの認識に関するパースペクティブである。現在、これに関しては次のヒューリスティクスを導入している。

H10 登場人物の持つ文脈(あるいはresource/situation)は、物語の話者(先生)の持つ文脈と等しい。

また、連続した会話文の話者、聞き手の同定に関しては、次のヒューリスティクスを導入している。

H11 会話文の連続は、明示的な話者/聞き手の表現がなければ、会話文

毎に話者と聞き手の立場の交替が起る。

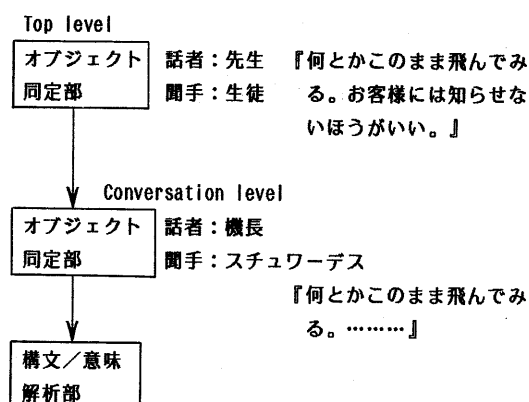


図 4.2 会話文の処理

5. 談話解析部

談話解析部はオブジェクト同定部の出力である記述状況を談話構造という形で記憶する。そもそも談話構造として何をどのような形式で保存するかというのは非常に研究的なテーマであり、今後の課題でもある。現在のDUALSではオブジェクト同定部から出力される、状況に関するコンディション及びオブジェクト同定処理の結果得られるオブジェクト間の関係をもって談話構造と呼んでいる。

6. 質問応答部

質問応答部[14]は、物語に関する質問文が入力されると、談話構造と知識ベースを参照して解答を探し、応答文を表示するモジュールであり、問題解決部と文生成部からなる(図 6.1)。以下に各サブモジュールについて述べる。

6.1 問題解決

物語文の内容に関する質問文が入力・解析され、その意味構造が送られると、問題解決部では、物語全体の意味を表現する談話構造及び知識ベースを用いて質問に対する解答を求め、それを次節で述べる文生成部に送る。状況意味論の立場からは、問題解決は質問の内容を示す不要要素へのアンカリングに対応する。談話構造は5.で述べたようにオブジェクト同定部からの出力である。一方、知識ベースはいわゆる一般知識・常識であり、問題を解くための推論規則である。DUALSで用いている一般知識はタイプ定義とコンストレイントに大別される。タイプ定義は世界におけるオブジェクトの性質やふるま

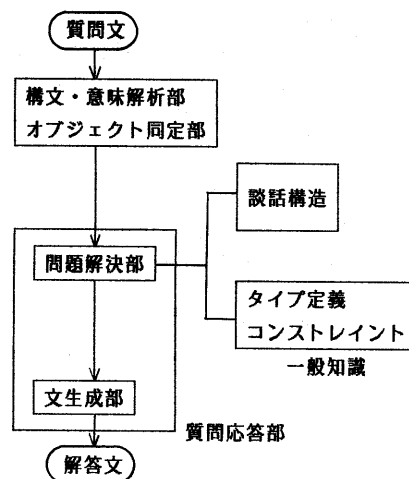


図 6.1 質問応答部の処理

い、あるいは事象のふるまいを記述するものであり、あらゆる事柄の原型(プロトタイプ)である。従ってタイプ定義はMinskyのフレーム[15]あるいはSchankのスクリプト[16]に類似したものであると考えられる。現版のDUALSのタイプ定義とコンストレイントはCIL[17]の初期の版により記述されている。図 6.2は「知らせる」という動作(事象)のタイプ定義の例である。

```
siraseru indexed __type
self: sit?1,
loc: loc?1,
pol: pol?1,
agent: ind?agent,
receiver: set?receiver,
information: sit?information,
cond: in(sit?1, loc?1,
(siraseru, ind?agent,
ind?receiver, sit?information),
pol?1).
```

図 6.2 「知らせる」という動作(事象)のタイプ定義

図 6.2は次のような意味を持つ。

「『知らせる』という動作はself, loc, pol, agent, receiver, informationという 6個のパラメータを持つ。その内容は 7番目のcondというスロットに

記述されており、ある状況sit?1 のロケーション（時間・場所）において、動作主ind?agent が受け手ind?receiverに情報sit?information を知らせるという条件が成立するか否か(pol?1の値が 1または 0かで区別される）という形式に表現される。」
同様に図 6.3は「知る」という動作のタイプ定義である。これについても同様に、

「『知る』という動作は、ある状況sit?1 のロケーションloc?1 において、動作主set?agent がある知識sit?knowledge を知る。」
ということを表している。

```

siru    indexed _type
self:   sit?1,
loc:    loc?1,
agent:  set?agent,
knowledge: sit?knowledge,
cond:   in(sit?1, loc?1,
           (siru, ind?agent,
            sit?knowledge), pol?).

```

図 6.3 「知る」という動作のタイプ定義

次にコンストレイントについて述べる。コンストレイントは事象の性質や事象間の関係を記述するためのルールであって、Prologと同様のHorn節の形式で記述される。図 6.4は図 6.2と図 6.3の 2つのタイプ定義を用いて、「ある動作主Aが他の人物Bに情報Pを知らせれば、BはPを知る。」という因果関係を記述した例である。

```

cause(Ins1, Ins2) <-
  look_into(Ins1, siraseru, Id1, Slots1) &
  create(siru, Id2, Ins2) &
  receiver!Ins1 = agent!Ins2 &
  information!Ins1 = knowledge!Ins2.

```

図 6.4 コンストレイントの例

図 6.4のコンストレイントは実行時には次のような動作をする。

もしワールド中に「知らせる」という動作のインスタンス（タイプ定義をオブジェクト指向プログラミングにおけるクラスと見做す）があれば「知る」という動作のインスタンスを作り、「知らせる」の与格と「知る」の主格の人物を同一化する。また、知

らせる情報と知る知識を同一化する。

このコンストレイントはAllen のアクションに関する理論[18]を単純化したものである。

図 6.5は「風が吹けば桶屋が儲かる。」のような長い距離の因果関係（因果関係連鎖）を辿るためのコンストレイントである。

```

causal_chain(Ins1, Ins2) <-
  cause(Ins1, Ins2).
causal_chain(Ins1, Ins3) <-
  cause(Ins1, Ins2) &
  causal_chain(Ins2, Ins3).

```

図 6.5 因果関係連鎖を辿るためのコンストレイント

図 6.5のコンストレイントは次のように解釈される。

「2つの事象Ins1とIns2が直接の因果関係にあるならばそれらの間には因果関係連鎖がある（1番目のルール）。2つの事象Ins1とIns2が直接の因果関係にあり、Ins2とIns3の間に因果関係連鎖があるならば、Ins1とIns3の間には因果関係連鎖がある（2番目のルール）。」

図 6.6のコンストレイントは物の上下関係を記述するコンストレイントであり、「AがBの上にあるならば、BはAの下にある。」という知識を表現している。

```

in(sit?, loc?, (ue, obj?, doko?), 1) <-
  in(sit?, loc?, (sita, obj?, doko?), 1).

```

図 6.6 物の上下関係を記述するコンストレイント

その他、「ある人が好ましくない状況を発見すれば、その人ははっとする。」というようなmental stateに関するヒューリスティクスもコンストレイントの形で持っている。

6.2文生成

問題解決部で解答が求まると、文生成部はその意味構造から自然言語風な文を生成する。ここでは次のような簡単な文生成アルゴリズムを用いている。

- (1) 解答を表す意味構造がsituation type以外の場合は、答となるオブジェクトに「です」という助動詞をつける。
- (2) 解答を表す意味構造がsituation typeの場合は、

そのsituation type中のリレーションの構造の部分
が表す情報を分解し、格パターンを用いて文にする。
格パターンは現在基本的な述部に対してのみ持っている。
例えば「吹出す」という述部に関してはargument
を2個とり、「AがBから吹出す」という表層
文を作るという情報を持っている。

〔例〕

「乗客は何人ですか?」という質問が出されると、
構文・意味解析部は、条件として、
in(sit*sit48, loc*loc45, (howmany,
zyoukyaku*zyoukyaku2, suu*suu1), pol*1) &
set __of(human*human2, sit*sit49,
hitobito*hitobito9) &
zyoukyaku*zyoukyaku2 = zyoukyaku*zyoukyaku1 &
hitobito*hitobito9 = zyoukyaku*zyoukyaku2

を満たすようなsuu*suu1を求める疑問文であると認識する。
問題解決部は談話構造の中から条件を満たす(マッチする)記述を見つけだし、suu*suu1に50という値をinstantiateする。
これは上で述べた方法(1)に当てはまるので文生成部は

「50 desu (50 です)」

という解答文を生成、出力する。

〔例〕

次は解答がsituation typeになる例である。「ロールさんは何故はっとしたのですか?」という質問に対して、
構文・意味解析部は、

in(sit*sit53, loc*loc49, (hattosuru,
ind#rollsan), pol*1) &
in(sit*sit54, loc*loc50, (why, sit*sit53,
sit*sit55), pol*1) &
in(ds*sent17, loc*sent17, (otoko, ind#rollsan), pol*1)

を満たすsit*sit53を求める疑問文であると解釈する。
つまりがロールさんがはっとした原因となる状況を求める。
問題解決部はコンストレイントを適用することにより、
談話構造の中から解答となるsituation sit*sit4を得る。
sit*sit4の構造は、

in(sit*sit4, loc*loc3, (hukidasu,
kemuri*kemuri1, engine*engine1), pol*1)

であり、その中のリレーション、

(hukidasu, kemuri*kemuri1, engine*engine1)

を分解、格パターンの情報を用いて、

「kemuri ga engine kara hukidasu (煙がエンジンから吹き出す)」

という解答文を得る。

7. 今後の課題

DUALS は構文、意味、語用解析さらに問題解決や文生成まで含むシステムである。現版のDUALS は状況意味論という一本の柱を軸にこれらの各処理の骨組を示したものである。各処理部とも今後の肉付けに待つところが多い。しかし、状況という実体の導入により自然言語の意味処理の見通しが大きく開けたということは示すことができたと思われる。深い談話理解のための条件として次のことが明らかにされていなければならない。

- 1) 日常言語における時間・空間概念のモデル
- 2) 人間のさまざまな行為の意味の定義
- 3) 行為の定義の基礎となる意図・知識・信念のモデル

これらに対する決定的な理論は存在していない。一方理論がどこかで出来上がってくるのを持っていることも許されない。従って応用領域を限定した上で、上述の問題を問いてゆく積極的な姿勢が必要とされる。状況意味論が提供している多くの新鮮なアイデアはその場合大いに役立つと期待される。

DUALS の今後の課題として意味理解処理のさまざまな問題点を掘り起す場を提供するということがある。そのためにはモデルのスクラップ・アンド・ビルドが何回か必要となるであろう。

今明らかになっている課題は次のとうりである。

- 1) DUALSの記述言語の整理を行うこと

現版のDUALS はDEC2060 上のDEC-10 Prolog でインプリメントされているが、大きく分けて 3つの言語によって記述されている。一つは裸のPrologでありオブジェクト同定部はこれで記述されている。二つ目はLFG である。LFG は構文解析と意味解析に用いられている。3番目はCIL である。正確にはCIL の初期の版である。問題解決部の知識はこれで記述されている。ここで指摘しておきたいことはLFG もCIL もともに単一化(Unification) をベースとする言語である。機能的に重複が見られる。LFG で書かれた文法記述をCIL に変換すれば機能重複のオーバーヘッドが解消され、システムが軽くなるであろう。CIL の効率の良い実装が鍵となる。PSI マシン[19]はバインド・フックのプリミティブを備えており、ESP [20]というシステム記述言語を既に持っているのでCIL のターゲット・マシンとして適当であると思われる。今後はPSI に移植してゆく予定である。

- 2) 次の課題は制御に関するものである。

- a) He saw her with a telescope.
- b) She bought it at the store.

c) He borrowed it from her.

この3つの文を順番に読む。最初の文には曖昧性がある。彼が望遠鏡を持っている彼女を見たのか、あるいは彼が望遠鏡で彼女を見たのかは最初の文だけでは分らない。次に文b)を読むと望遠鏡を持っているのは彼女であると思われる。しかし文c)を読むと逆に「彼が望遠鏡で見た」が妥当な解釈になる。この3文のあとにさらに文が続けば別な解釈も可能である。DUALSは単純なバックトラッキング制御なのでこの例のような人間らしい制御はできない。いくつかの可能性を同時に保持し、それらの間の優先順位を情報の増加にともなって更新していく方式が必要である。これらは非単調論理として現在盛んに研究されている問題と深く関連している。

3) 扱える文章のレパートリーを増やすこと。

現在、DUALSは固定された文章を理解できるのみである。文法、辞書、常識を増やして一般性を示すことである。

4) モデルを展開すること。

最初に述べたように時間・空間概念、行為、そして意図・知識・信念のモデルを作ることは柔軟な自然言語インターフェイスの鍵を握っている。これらのモデルと計算規則あるいはヒューリスティクスを開発することである。

[参考文献]

- [1] J. Barwise and J. Perry, Situation and Attitudes, MIT Press, 1983.
- [2] 新世代コンピュータ技術開発機構：電子計算機基礎技術開発成果報告書、基礎ソフトウェアシステム編、1985.
- [3] R. M. Kaplan and J. Bresnan, Lexical-Functional Grammar: A Formal System for Grammatical Representation, in Bresnan(ed.) The Mental Representation for Grammatical Relations, MIT Press, 1982.
- [4] F. Pereira, and D. Warren, Definite Clause Grammar for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks, Artificial Intelligence, 13, 1980.
- [5] Y. Matsumoto, H. Tanaka, et al, BUP: A Bottom-Up Parser Embedded in Prolog, New Generation Computing, Vol.1, No.2, 1983.
- [6] 安川秀樹, 古川康一：文法関係の形式的な記述について-LFG in Prolog -, 自然言語処理研究会39-5, 1983.
- [7] H. Yasukawa, LFG in Prolog, Proc. of COLING'84, 1984.
- [8] 田中裕一, その他：状況意味論に基づく談話理解システムDUALS -その基本原理, 自然言語処理研究会50-6, 1985.
- [9] 安川秀樹：ユニフィケーションによる意味解析, ICOT TR-110, 1985.
- [10] 平川秀樹：談話理解システムDUALSにおけるオブジェクト同定処理, ICOT TR-117, 1985.
- [11] C. L. Sidner, 'Focusing in the Comprehension of Definite Anaphora', in Berwick(ed.) Computational Models of Discourse, MIT Press, 1984.
- [12] 白井賢一郎：日本語における指示(Reference)の問題について-モンタギュー文法の観点から-, 京都大学文学部, 修士論文, 1979.
- [13] M. Kameyama, Centering in Japanese: Basic Observation, Draft of Ph.D thesis, Dept. of Linguistics, Stanford Univ., 1984.
- [14] 三吉秀夫：談話理解システムDUALSにおける質問応答処理, ICOT TM-0110, 1985.
- [15] M. Minsky, A Framework for Representing Knowledge, in Winston(ed.) The Psychology of Computer Vision, McGraw-Hill, 1975.
- [16] R. C. Schank and R. Abelson, Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates, 1977.
- [17] K. Mukai, Horn Clause Logic with Parameterized Types for Situation Semantics Programming, ICOT TR-101, 1985.
- [18] J. F. Allen, Recognizing Intentions from Natural Language Utterances, in Berwick(ed.) Computational Models of Discourse, MIT Press, 1984.
- [19] H. Nishikawa, H. Yokota, et al, The Personal Inference Machine(PSI): Its Design Philosophy and Machine Architecture, ICOT TR-13, 1983.
- [20] T. Chikayama, ESP Reference Manual, ICOT TR-44, 1984.