自然言語処理 61－3
（1987. 5. 22）

多重領域をサポートした意味解析言語

松島 利幸

日立ソフトウェアエンジニアリング（株）

自然言語インターフェイスにおける、文脈情報を取り込んだ意味解析の一手法について述べる。意味解析を、入力文に対応する構文木の空間と文脈の空間との直積空間から意味表現空間への写像として捉える。この方法では、自然言語の単語をオブジェクトとして捉えており、それらには意味写像を実現する写像が付随している。これらの写像は、意味解析用のオブジェクト指向型言語で記述している。この言語は、各文脈毎の意味解析を記述するために、オブジェクト集合を各文脈に対応する多重領域に分割する機能を備えている。更に、領域の集合にラティス構造を入れることで、複雑な文脈にも対処できるようにしている。

An Language for Semantic Analysis which supports Multi-Domained Object Space

Toshiyuki Matsushima
Hitachi Software Engineering Co.,Ltd.
6-81, Onoe-Machi, Naka-ku, Yokohama 231, Japan

This paper discusses a semantic analysis method dealing with contextual information of natural language interface.

Semantic analysis is considered as the mapping (semantic mapping) from the product space of parse trees of inputted sentences and context. The words of natural language are considered to be objects attached with semantic mappings. These attached mappings are described by an object oriented language for sematic analysis. The language has a facility for contextual analysis by dividing the set of objects into closed domains which corresponds to each context. The domains can be structured as a lattice space for complex for complex contextual analysis.

１．はじめに

現在、私達は知的ＯＡシステム研究の一環として、自然言語インターフェイスにおける意味解析処理を中心にして、研究を進めてきた。その成果として、オブジェクト指向型の意味解析言語konoNUL、及び、自然言語インターフェイスシステムkonoNUSを開発した。初期の試作バージョンでは、参照語の解析や省略文の解析などのように、文脈に依存する意味解析機能は、サポートされていなかった。

そこで、昭和６１年度前半、文脈に依存する意味解析処理をサポートすると共に、対話における大域的な文脈に依存した意味解析を行うための機能を追加した。　本報告では、まず、我々がＯＡシステムをどのように捉えており、その中で自然言語インターフェイスを如何に位置づけているかを述べる。次に、文献［６］で述べたkonoNUSにおける意味解析手法について簡単に説明すると共に、若干の補足をする。そして、最後にkonoNUSでの文脈表現と文脈解析の手法について述べ、処理例を示す。

２．秘書ロボットプロジェクト

２．１秘書ロボット・システム

現在、我々は秘書ロボットプロジェクトと称して、ＯＡの統合システムを研究中である。このシステムは、オブジェクト指向型手法に基づくもので、オフィスにおける各々の業務を司る秘書ロボットが互いに自律的に通信できるようにしたものである。秘書ロボットとは、各々の業務を行う実体を抽象化したものである。従って、秘書ロボットが各々のワークステーション毎に実現される場合もあれば、マルチタスクＯＳ上のプロセスとして実現される場合もある。また、ロボット間の交信もＬＡＮを用いたり、プロセス間通信を用いたり、実現方法は様々である。

ユーザは、端末ロボットを介して、各業務ロボットに作業を依頼する。各ロボットは、各々の業務に必要なデータベースを抱えているので、このようなシステムにより、業務の分散化のみならず、データベースの分散化にもなっている。

秘書ロボットにより、オフィス業務が行われる様子を具体例を用いて説明する。図１に示すように、人や会議室のスケジュールを管理する会議予約ロボットと、人事データベースロボットがＬＡＮで結合されているとする。（ここでは、話を簡単にするために、ＬＡＮによって秘書ロボットが交信するとした。）ユーザは、ＬＡＮに接続されている端末ロボットを介して、会議予約ロボットにある会議の開催を予約したとする。この場合、会議予約ロボットは会議のスケジュール・データを変更すると共に、会議の出席者に対して、会議の開催通知を電子メールで発送する。この際、人事データベース・ロボットに問い合わせて、出席者の所属部署を調べ、該当部署宛に電子メールを送る。勿論、会議予約ロボットが人の所属部署の情報を持っていても良いのであるが、各ロボットが重複するデータを持つことは、データの一貫性を保証する上で、非常に危険である。従って、各秘書ロボットの持つデータは、できるだけ重複しないようにする。もし、重複データが存在する場合には、データの更新に際して、秘書ロボット間で連絡を取りあって、データの一貫性を保つようにする。

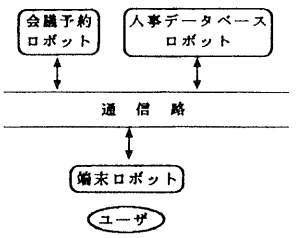


図１　秘書ロボットシステム

２．２自然言語インターフェイスロボット

ここで、自然言語インターフェイスが秘書ロボットとして、どのように使われるか述べる。まず、自然言語インターフェイスロボットを使わない場合の処理を図２に示す。ユーザが会議予約ロボットの操作コマンドを知っているときは、直接コマンドを送り、業務を依頼できる。

しかし、ユーザが会議予約ロボットの操作コマンドを知らな

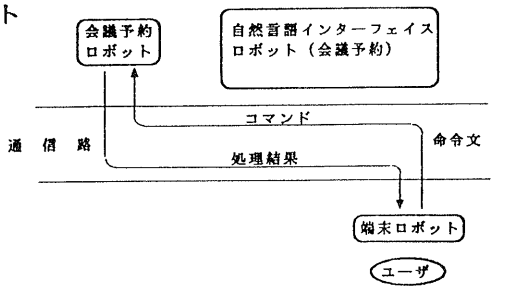


図２　コマンドによる会議予約ロボットの操作

いときには、図３に示すように、ユーザが自分のやりたいことを日常語で表現して、会議予約インターフェイスロボットに送る。会議予約インターフェイスロボットは、与えられた文章の意味を解析して、会議予約ロボットにコマンドを送る。そして、返されてきた処理結果を基に文章または表を生成して、ユーザの端末ロボットに送り返す。

更に、自然言語インターフェイスロボットのポータビリティを高める為に、図４のような構成にする。各秘書ロボット対応に、自然言語インターフェイスロボットを置くのではなく、汎用的な自然言語インターフェイスロボットを用意し、各ロボット対応に意味辞書ロボットを用意する。各ロボット対応に言葉の意味も代わるから、自然言語インターフェイスロボットは、意味を解析する際に、意味辞書ロボットに問い合わせる。こうすると、意味辞書ロボットを各々の秘書ロボット対応に作れば良い。

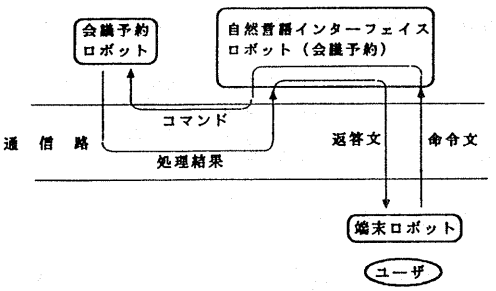


図３　自然言語による会議予約ロボットの操作

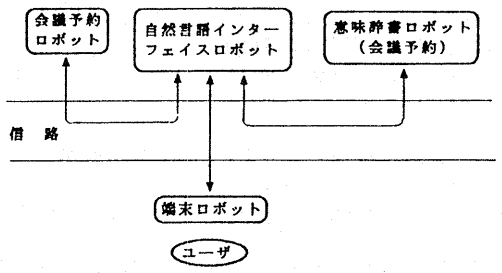


図４　各秘書ロボットごとの意味辞書ロボット

３．意味解析

ここでは、konoNUSにおける意味解析について簡単に説明する。記号などの定義は文献[６]と同じである。詳しくは[６]を参照されたい。

３．１　konoNUSにおける意味解析手法

konoNUSでは、意味解析を構文木の空間から意味表現形の空間への写像であると捉えている。構文木の空間Treeとしては、次のBNFを満たすS式の空間であるとする。

<tree> ::= (<word> {<slot>}*)
<slot> ::= (<case> <tree>)

ここで、<word>とは、名詞、動詞等に対応するシンボルであり、<case>は助詞等の表層格を表わすマーカである。また、意味表現形の空間Repは、構文木と同様に次のBNFを満たすS式の空間である。

<meaning> ::= (<primitive> {<attribute>}*) | (<concept>)
<attribute> ::= (<attribute name> <meaning>)

<primitive>は、ターゲットシステムの処理やデータを表わすための基本的な概念を表わす。また、<attribute>は、それらの概念の属性を表わす。<concept>は、意味ネットのノードに対応していて、ターミナルな属性値として捉えられる。

次に、意味解析に対応する写像（意味写像）を考える。基本となるのは、

「ある単語への修飾は、その単語が表わす概念の属性を与えることである」

という考え方である。従って、意味写像は各単語毎に、修飾とそれに対応する属性の表現を与えることにより記述される。意味解析を表わす写像をΦとすると、

Φ：　Tree ---> Rep

Φを各単語ωの上で与える。つまり、下のようなωに対する修飾の集まりSlot(ω)を考え、

Slot(ω) = { $(slot_1, \ldots, slot_n)$: $(\omega\ slot_1, \ldots, slot_n) \in$ Tree }

Slot(ω)からRepへの写像Φ_ω を次のように定義する。

$\Phi_\omega(s) = \Phi((\omega\ slot_1, \ldots, slot_n))$　$s = (slot_1, \ldots, slot_n) \in Slot(\omega)$

ここで、Φ_ω がどのようにして記述されるか具体例を挙げて説明する。

ω ＝「会議」の場合を考える。Repにおいて、「会議」にはeventという<primitive>が対応し、eventの<attribute name>は、event-name，event-time，event-placeであるとする。event-nameはeventの名前、event-timeはeventの開催時刻、event-placeはeventの開催場所を表わす属性である。「会議」を修飾する語としては、「来週の」、「明日の」といった時間を表わす語や、「第1会議室での」、「講堂での」といった場所を表わす語が考えられる。そして、「<時間を表わす語>の」といった修飾により、event-timeの属性値が決まり、「<場所を表わす語>での」といった修飾により、event-placeの属性値が決まる。これをIF-THEN型のルー

ルで表わすと次のようになる。

$RULE_1$: IF「<時間を表わす言葉X>の」THEN event-timeの属性値がX

$RULE_2$: IF「<場所を表わす言葉X>での」THEN event-placeの属性値がX

$RULE_1$、$RULE_2$に対応する写像を各々ϕ_1、ϕ_2とすると、

ϕ_1((の (明日))) = (event (event-time <明日>))

ϕ_2((での (会議室))) = (event (event-place <会議室>))

ここで、<明日>、<会議室>は、各々「明日」、「会議室」の意味表現形を表わす。また、修飾語の如何にかかわらず、event-nameは<会議>である。これは、

$RULE_0$: IF ANY THEN event-nameが<自分自身>

で、表わされる。$RULE_0$に対応する写像をϕ_0とする。

ϕ_0( ) = (event (event-name (会議)))

「明日の会議室での会議」というように、2つ以上の語で修飾される場合には、event-time、event-placeの属性値が同時にセットされる。

$\Phi_{会議}$ ((の (明日)) (での (会議室))) = (event (event-name (会議))
(event-time <明日>)
(event-place <会議室>))

このような写像をϕ_0、ϕ_1、ϕ_2から合成するのが積オペレータである。

$\Phi_{会議} = \phi_0 \otimes \phi_1 \otimes \phi_2$

積オペレータ⊗は、大雑把に言って、属性を組み合わせる働きをする。

一方、2つ以上の異なった意味を持つ単語がある。例えば、「開く」は、「イベントを開催する」という意味の他に「(扉などを) 開ける」という意味がある。このような意味の切り分けを実現するのが、和オペレータ⊕である。このオペレータの定義は［6］では詳しく述べてないので、ここで与えることにする。

ωを単語として、ω上の表現写像をϕ_ωとする。このとき、ϕ_ωの有効定義域$\Delta(\phi_\omega)$を次のように定義する。

$\Delta(\phi_\omega) = \{\sigma \in Slot(\omega): \phi_\omega(\sigma) \neq 空値\}$

ここで、空値はある表現写像では無意味である修飾として扱われるものである。

さて、ω上の表現写像ϕ_ω、ϕ_ωが与えられ、次の条件を満たすとする。

$\phi_\omega(\sigma) \sim \phi_\omega(\sigma) \quad (\forall \sigma \in \Delta(\phi_\omega) \cap \Delta(\phi_\omega))$

このとき、次式で⊕を定義する。

$\phi_\omega \oplus \phi_\omega(\sigma) = \phi_\omega(\sigma) \ (\sigma \in \Delta(\phi_\omega))$
$= \phi_\omega(\sigma) \ (\sigma \in \Delta(\phi_\omega))$
$=$ 空値

このオペレータが有効なのは、

$\Delta(\phi_\omega) \cap \Delta(\phi_\omega) = \{()\}$

の場合である。ここで、'()'は空リストであって、修飾語がない場合に対応している。これは、ϕ_ωとϕ_ωがωの直交する意味を表わしていることを示す。

3．2　多重継承

konoNUSでの意味解析は、各単語(オブジェクト)毎に対応する意味写像を記述することで実現される。しかし、全ての単語に対して意味写像を記述することは大変であるし、その必要もない。実際、次のようにすればよい。

（1）単語をノードとして意味ネットの中に埋込み、ネットワークのノードをオブジェクトとして捉える。そして、意味的上下関係でオブジェクト間の上下関係を指定する。

（2）各オブジェクト(単語)には、TreeからRepへの表現写像をセットしておく。この表現写像は各単語に対応する概念の属性値を、単語への修飾に応じて指定するものである。上位の概念から下位の概念に属性が継承されることに対応して、上位オブジェクトの表現写像を下位オブジェクトに継承させる。

この継承により、一部のオブジェクトに対して、表現写像を書くだけで済む。特に、名詞については、シソーラスの上下関係を用いると、極少数のオブジェクトに対して表現写像を記述するだけで良い。

表現写像の継承は⊗オペレータによって実現される。つまり、単語ω_1とその上位の単語ω_2があったとして、付随する表現写像を各々ϕ_{ω_1}、ϕ_{ω_2}とする。このとき、ω_2のϕ_{ω_2}がω_1に継承して、ω_1上の表現写像

$\phi_{\omega_1} \otimes \phi_{\omega_2}$

が得られる。では、2つ以上の上位の単語があったときはどうなるであろうか。意味的に考えて次の2つの場合がある。ω_1の上位の単語として、ω_2、ω_3があったとし、付随す

る表現写像をϕ_{ω_1}、ϕ_{ω_2}、ϕ_{ω_3}とする。

（1）上位の単語の意味が論理積として捉えられる場合。
ω_1がω_2とω_3の両方の意味であるとき、ω上の意味写像は、

$$\phi_{\omega_1} \otimes (\phi_{\omega_2} \otimes \phi_{\omega_3})$$

で与えられる。例えば、「田中」という人がいて、横浜市の「市民」でもあり、日立ＳＫの「社員」でもあるとする。「市民」としての「田中」さんには、<所属行政区>という属性があり、「社員」としては、<所属部署>という属性がある。「中区の田中」と言ったとき、「<所属行政区>が中区である。」という意味であり、「研究部の田中」とすれば、「<所属部署>が研究部である。」という意味になる。<所属行政区>も<所属部署>も「田中」さんの属性としては両立するものである。従って、「市民」に付随する表現写像ϕ　　と「社員」に付随する表現写像ϕ　　は、⊗オペレータを用いて合成する。

（2）上位の単語の意味が排他的論理和として捉えられる場合。
ω_1がω_2、ω_3のどちらか一方の意味であるとき、ω上の意味写像は

$$\phi_{\omega_1} \otimes (\phi_{\omega_2} \oplus \phi_{\omega_3})$$

で表わされる。例えば、「開く」は、イベントを「開催する」という意味の場合もあれば、ドアを「開放する」という意味の場合もある。そして、「開催する」という意味と「開放する」という意味は直交している。ここで注意すべきは、どちらの意味になるか、修飾だけでは分からないときもあるということである。例えば、「いつ開くか。」と言った場合、「開催する」なのか「開放する」なのか、文脈に依存する。

４．文脈解析
４．１　文脈情報を取り込んだ意味写像
　さて、これまで構文木の形のみを基にして、意味解析することを述べてきた。つまり、意味解析をTreeからRepへの写像であるとした。しかし、上でも少し触れたように、実際には、文脈情報無しでの意味解析は不可能である。自然言語インターフェイスでは、各秘書ロボットの操作と言う、かなり限られた文脈で意味解析をすればよい。それでも、文脈情報は必要である。例えば、「会議」という言葉を考えてみる。

対話例１
ユーザ　：「会議を検索したい。」
システム：「どのような時間範囲の会議を知りたいのですか。」
ユーザ　：「３時から７時までの会議です。」

対話例２
ユーザ　：「会議を予約したい。」
システム：「どんな会議を予約したのですか。」
ユーザ　：「３時から７時までの会議です。」

「３時から７時までの会議」の意味は、上記２つの対話では、次のようになる。

対話例１：「開催時間が、３時から７時までの範囲に含まれる会議」

対話例２：「開催時間が、３時から７時までの会議」

このように、「会議」に対する修飾語が全く同じでも、意味が違ってくる。
そこで、意味写像を単に構文木の空間Tree上で定義するのではなく、構文木を文脈の対にの空間上で定義すべきである。文脈を表わす空間Contextが与えられたとして、意味写像は、

$$\Phi : \mathrm{Tree} \times \mathrm{Context} \longrightarrow \mathrm{Rep}$$

というように、TreeとContextの直積空間Tree×ContextからRepへの写像として表わされる。ある文脈 c （c ∈ Context ）が固定された場合は、勿論Φは TreeからRepへの写像になる。

$$\forall c \in \mathrm{Context},$$
$$\Phi^c : \mathrm{Tree} \longrightarrow \mathrm{Rep}$$
$$\Phi^c(t) = \Phi(t, c) \quad (t \in \mathrm{Tree})$$

Contextをどのように構成し、記述するかは非常に難しい問題である。
konoNUSでは、過去の対話の記録である文脈スタックΣと大域的文脈を記述する文脈状態

Γの対でもって、文脈を表現する。つまり、文脈は文脈スタックと文脈状態の直積で表わされる。

Context = Σ×Γ

ここで言う大域的文脈とは、自然言語インター-フェイス・ロボットが交信している秘書ロボットでの処理の各フェイズに対応するものである。例えば、我々が自然言語インターフェイス・ロボットを試作した会議予約ロボットでは、次の３つの文脈状態がある。

Γ = {検索、追加イベント特定、追加イベント整合性チェック}

また、文脈スタックは、入力文に対応する構文木、意味表現形、秘書ロボットの処理結果から成る。Σは、「それ」、「その」などの指示語に対応する単語や省略文での足りない述語をサーチして求めるのに用いる。

４．２　オブジェクトの多重領域化による文脈解析の実現

では、文脈に依存した変換をどのようにして記述したら良いであろうか。つまり、文脈ｃ（ｃ∈ Context）に対して、上のΦ^cを記述する訳である。ひとつの方法は、オブジェクトにおけるIF-THEN型の処理規則で、文脈を陽に指定するものである。つまり、IF-THEN型ルールの条件部に、ある文脈であるかどうかをチェックする条件パターンを書けるようにする。しかし、次の理由からこの方法は好ましくない。

（１）各オブジェクトにすべて文脈に対応する処理規則を書くと、オブジェクトの処理記述が複雑で錯綜したものになってしまう。

（２）複数の人により開発作業をする場合、各々の文脈毎に開発者が異なることもある。このようなとき、各人のインタ-フェイスが取りにくい。

そこで、オップジェクト集合を各文脈状態毎の部分集合（オブジェクト領域）に分けて、多重領域化する。そして、各オブジェクト領域の中だけで、メッセージのやりとりが行われるようにする（図５）。

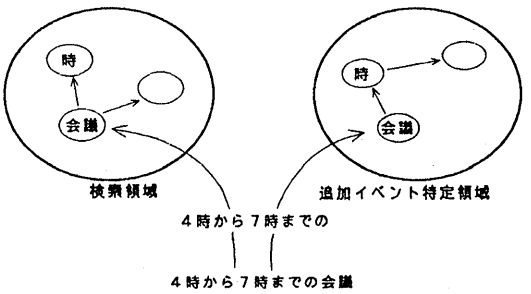


図５　各文脈対応のオブジェクト領域

つまり、文脈状態γに対応するオブジェクト領域Ｄ(γ)とすれば、Ｄ(γ)には

$\Phi^{(\gamma)}$: Tree×Σ ---> Rep

$\Phi^{(\gamma)}(t, \sigma) = \Phi(t, (\sigma, \gamma))$

を記述する。

例えば、上の例において「３時から７時までの会議」を解析することを考える。構文木は

（会議　（から　（時　（val（３）））
　　（まで　（時　（val（７））））

となる。対話例１では、検索領域の「会議」に、メッセージ

{（から　（時　（val（３））），（まで　（時　（val（７））））}

が送られる。検索領域の「会議」には、

IF（から <時間を表わす言葉Ｘ>）（まで <時間を表わす言葉Ｙ>）
THEN（event（event-time <ＸからＹまでの範囲に含まれる>））

というルールが書いてある。このルールにより、「会議の開催時間が３時から７時までの時間範囲に含まれると解釈される。一方、対話例２では、同じメッセージが<追加イベント特定>領域の「会議」に送られ、

IF（から <時間を表わす言葉Ｘ>）（まで <時間を表わす言葉Ｙ>）
THEN（event（event-time <ＸからＹまでに等しい>））

というルールによって解析され、「会議の開催時間が３時から７時までである」と解釈される。

４．３　メッセージ・センディング・パスとしての文脈

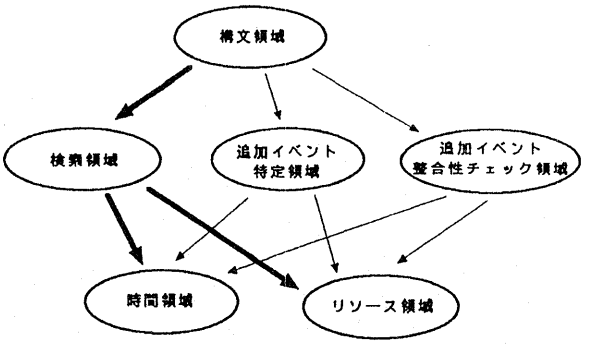
上で述べたオジェクト領域の考え方は、あるひとつの文脈における意味解析処理を閉じた世界に閉じ込めることにより、モジュール性を高めるものである。しかし、複数の文脈で共通する意味解析処理も存在する。共通処理を重複して記述せずに済ませるために、オブジェクト領域の集合にラティス構造（半順序構造）を導入する。

［６］で述べたように、初期バージョンのkonoNUSでは、構文に由来する意味を解析するオブジェクト領域(構文層)と単語固有の意味を解析するオブジェクト領域(意味層)を設

定し、ある単語ωの意味を解析するときは、まず、構文層のオブジェクトωにメッセージを送り、そこから意味層のオブジェクトωにメッセージを送るようにしていた。構文層に記述された処理は、文脈に依存しないものであった。そこで、複数の文脈を扱う場合も同様に、〈構文〉オブジェクト領域(構文層をこう呼ぶことにする)にメッセージを送ってから、検索オブジェクト領域等に送る。従って、文脈状態には、メッセージ・センディングのパスが対応する。

Γ = {〈構文〉 --> 〈検索〉,
〈構文〉 --> 〈追加イベント特定〉,
〈構文〉 --> 〈追加イベント整合性チェック〉}

更に、もう少し複雑な構造を入れることもできる。時間を表わす言葉の処理を扱う〈時間〉オブジェクト領域、人や会議室などのリソースを扱う〈リソース〉オブジェクト領域は、全ての文脈において共通に使われる。しかし、それらは各単語レベルの意味解析において使われるので、〈検索〉、〈追加イベント特定〉等の次のレベルにくるべきものである。従って、オブジェクト領域間の関係は図６のようになる。この場合、各文脈状態には、木構造のメッセージ・パスが対応している。



注) ⟶ は、検索処理の文脈に対応するメッセージパスである。

図６　ラティス構造が入った多重オブジェクト領域

Γ = {〈構文〉 --> 〈検索〉 --> 〈時間〉
--> 〈リソース〉,
〈構文〉 --> 〈追加イベント特定〉 --> 〈時間〉
--> 〈リソース〉,・・・・ }

４．４　処理例

図７と図８に、konoNUSでの処理結果を示す。応答時間は、２秒から５秒である。

ユーザ：　来週会議はいつどこで開かれますか。

システム：５月２２日の１４時から１５時まで第１会議室で開かれます。

ユーザ　：それには誰が出席しますか。

システム：鈴木と佐藤と田中と加藤が出席する部長会議が開かれます。

ユーザ：　鈴木部長の明日の予定を知りたい。

システム：該当するイベントは、次の通りです。

会議名　：　部長会議
開催場所：　特別会議室
開催日時：　５月２３日　９時から１０時まで

ユーザ　：明後日は？

システム：該当するイベントは次の通りです。

会議名　：　予算審議会
開催場所：　第２会議室
開催日時：　５月２４日　１０時から１２時まで

ユーザ　：田中課長は？
システム：該当するイベントは次の通りです。

図７　検索処理の例

ユーザ　：来週の月曜の午後３時から４時まで会議を開きたい。

システム：どんな会議ですか。

ユーザ　：課長会議です。

システム：次のようで宜しいですか。
会議名　：課長会議
開催場所：第１会議室
開催日時：５月２５日　１５時から１６時まで
主席者　：斎藤、中田、河村、加藤
議題　　：定例

ユーザ　：議題を製品開発に変えたい。

システム：次のようで宜しいですか。
会議名　：課長会議
開催場所：第１会議室
開催日時：５月２５日　１５時から１６時まで
出席者　：斎藤、中田、河村、加藤
議題　　：製品開発

ユーザ　：はい。

ユーザ　：明日の１１時から１２時まで第１会議室で、賞与査定について部長会議を開きたい。

システム：次のようで宜しいですか。
会議名　：部長会議
開催場所：特別会議室
開催時間：５月２４日　１１時から１２時まで
出席者　：鈴木、佐藤、田中、斎藤
議題　　：賞与査定

ユーザ　：はい。

図８　追加処理の例

５．おわりに

今回、文脈を考慮した意味解析を実現するために、オブジェクトの集合を多重領域化し、大域的文脈をオブジェクト領域と対応させることを試みた。この結果かなりの有効性は確認できたが、文脈解析の十分な定式化を与えるまでには到っていない。特に、文脈スタックの情報を意味解析に取り込む処理に関しては、指示語や、省略文が現われるいろいろなケース毎に、文脈スタックをサーチする方法を変えて対応しているに過ぎない。今後、この処理の定式化を与えたいと考えている。また、意味表現空間の構造の意味を定式化する予定である。

参考文献

[１] Goldberg A. , Robson D. "Small-talk 80 The Language and Its Implementation" Addison Wedsley 1983.

[２] Griffith P. L., Robson D. "Three Principle of Representation for Semantic Networks" ACM TODS Vol.7 No.3 1983

[３] Simmons R. F., Chester D. "Relatingf Sentences and Semantic Networks with Procedual Logic" CACM Vol.25 No.8 1982

[４] 奥村　他、「意味解析言語ＳＲＬ／０の設計とＤＣＫＲ」
自然言語処理研究会報告　No.54 5 1986

[５] 大沢　他、「オブジェクト指向方式による対話理解システム」
自然言語処理研究会報告　No.44 7 1984

[６] 松島　他、「ポータビリティを目指した日本語インターフェイスの試作」
自然言語処理研究会報告　No.54 4 1986