

知的コマンドインタープリタICIにおける問い合わせ文の理解

美馬 秀樹, 青江 順一
徳島大学

自然言語入力文より、情報システムのコマンドとコンサルティングを自動実行する知的コマンドインタープリタ (ICI, Intelligent Command Interpreter) の実現について述べる。ICI は情報システムと協調することで、情報システムの実状況に適合したユーザーへの応答を可能にする。ICI は、各コマンドに対するコマンド知識を知識ベースとして用意し、ユーザーの要求が不十分な場合はコマンド知識の制約条件を検査することで段階的にコマンド実行に導くので、コマンドの学習と、安全な利用環境が実現できる。また、入力文の意図を理解するために、コマンド知識に意図リンクを定義し、ユーザーの意図が陽に示されていない発話においても、真に意図するコマンドを推論するアルゴリズムを提案する。

QUERY UNDERSTANDING IN THE ICI INTELLIGENT COMMAND INTERPRETER

Hideki Mima, Jun-ichi Aoe

The University of Tokushima
Minami Josanjima Cho 2-1, Tokushima-Shi 770, JAPAN

Abstract

The ICI (Intelligent Command Interpreter) system is an intelligent natural language interface which allows users to execute and learn commands of a given system—e.g., OS, information systems, etc. Users are assisted and instructed by ICI, which acts as an intelligent consultant, in issues concerning commands. Knowledge about commands is contained within a knowledge base. When the demand for the execution of commands is incomplete, users are assisted by ICI through a gradual guidance until the user's demand has been satisfied or the goal has been achieved. Two different tools allow ICI to understand a user's intention: a) intention links, which are defined into the command knowledge base; and b) an algorithm which permits ICI to infer, from queries that do not express the user's real intention explicitly, the command intended by the user.

1. まえがき

近年の情報システムの開発の活発化と共に、システムの機能はますます複雑化・高度化してきたが、利用環境が向上しているとは言いがたい。特に、O A、F Aの進展とも相まって誰もが容易に使える情報システムが望まれているにもかかわらず、分厚いマニュアルを学習し、コマンド機能を覚えなくてはならないのがユーザの大きな障害になっている。

この計算機ユーザインターフェイスを改善する研究は、視覚的なコマンドインタプリタを実現するものと自然言語対話を導入した知的ヘルプ機能に基づくコマンドインタプリタを実現するものに大まかに分類できるが、双方を取り入れる場合もちろん可能である^[8]。前者はユーザに分かりやすいが、コマンド言語の表現力を制限しており、また後者は表現力は大きいもののユーザの意図を抽出するのが難しい^[21]。しかし、自然言語でやりたい動作を要求することは、ユーザが特別な準備なしに入り込める非常にスムーズなアプローチであり、データベースの問い合わせ^[13-16, 17-19]；マニュアル情報やコマンドのコンサルティング^[12, 21, 22]を目的とした自然言語マンマシンインタフェースの研究として進められている。本報告の目的は、冒頭で述べたようにコマンド機能に関するユーザインターフェイスを改善することであり、自然言語入力文に対してユーザの意図を理解し、情報システムの実状況に適合したコマンドを自動実行させる知的コマンドインタプリタ (I C I, Intelligent Command Interpreter) の実現を目標とする^[6, 6, 7, 10]。

しかし、上述のように、すべての発話文において、ユーザの意図がその表層に表れているとは言えず、ユーザの真の意図を抽出するのは非常に難しい課題である。本報告では、これを解決するために、各コマンドに対してコマンド実行に対する制約条件と、表層から得られるコマンドから意図するコマンドを関係づける意図リンクを定義したコマンド知識を構築する。そして、この意図リンクを利用したコマンドの推論のアルゴリズムを提案する。これによって、ユーザの意図が陽に示されていない発話においても、真に意図するコマンドを推論することが可能になった。

2. システムの概要

2. 1 システムの構成

I C Iは、図1のように構成され、次の解析部と知識ベースを使用する。

- 1) 形態素解析部は、入力文の分類、語彙の属性構造、形態素の局所的合成、語彙とコマンドの対応を決定し、語彙知識辞書^[3, 4]を利用する。
- 2) 実状況知識獲得部は、入力文の名詞句の実体が情報システムに存在するの可否かをチェックし、存在すればその属性と属性値を決定して語彙知識から得られた静的知識にこの動的知識を融合する。
- 3) 構文・意味解析部は、ユニフィケーション文法^[8-10, 16, 21]に基づく拡張LRパーサで実現され、1)と2)から送られる属性構造に基づいて解析を行う。
- 4) 文脈知識は、入力文が不完全で付加情報を必要とする場合、属性構造を格納してインクリメンタルな増強を行うための一時的な知識である^[1, 2]。
- 5) コマンド理解部は、構文・意味解析結果とコマンド知識を参照して、正しいコマンドの自動生成と実行が可能かどうかを決定するものであり、それが不可能な場合は、ユーザに付加情報の入力を催促する応答文を生成する。
- 6) コマンド知識は、コマンドが実行できる実状況の制約条件を保有し、コマンド生成と応答文の情報を提供する。さらにコマンド推論における、発話意図に関するコマンドの関連情報を与える。

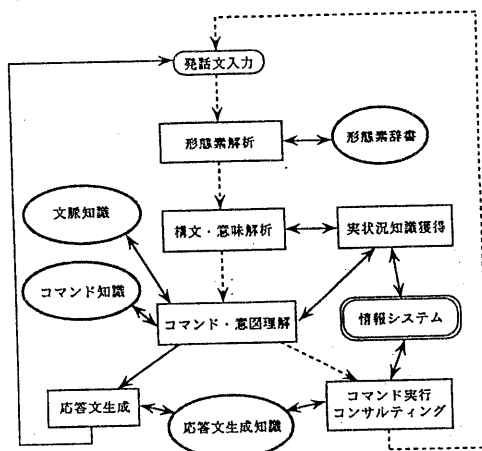


図1 I C Iの構成

7) 文生成知識は、5)の応答文を生成する文生成のパターンであり、属性構造の不足情報や制約外の属性値について回答し、新しい情報の入力进行要求する^[23]。

2. 2 指示実体の分類

ユーザが要求するコマンド機能を理解・実行するためには、入力文の名詞句の対象世界（情報システム）での指示実体（referent）を同定・理解し、システムの状態に依存したコマンドを実行する必要がある。

例えば、UNIXシステムで入力文「a.cをb.cにコピーして下さい」に対して、「cp a.c b.c」を実行するためには「a.c」と「b.c」に対して次のことを実状況知識として理解する必要がある。

- (1) a.cの実体が（現在のディレクトリに）存在する。
- (2) a.cが存在すれば、読みだし可能である。
- (3) b.cの実体が存在しない。

このような名詞句の実体を理解するために、情報システムに現れるすべての事物の集合（UNIVERSE）を対象として、名詞句と対象世界における指示照応関係（reference）を吉浦ら^[20]の分類を利用し、I, E, AT, Nを考慮した対話構造を考える。さらに、UNIXなどでは、一般的にユーザはその対象世界を暗黙的に決定していると思われる場合があり、例えば(1)では、ユーザはa.cの存在する対象世界がカレント・ディレクトリだと考えている。よって、UNIVERSEのサブセットとして個々の対象世界をDOMAIN(i, j)（但し、iを対象世界の实体名、jは対象世界の属性とする）と定義し、その対象世界に対する指示照応関係を表1のように定義する。

表1 指示照応関係の分類

分類	内容	例
Identifiable(I) in DOMAIN(i, j)	聞き手がDOMAIN(i, j)に実体を同定しうる	a.cを削除したい a.cがDOMAIN(\$current, directory) カレント・ディレクトリに唯一つある
Anaphoric Total(AT) in DOMAIN(i, j)	候補が複数個あり、実体はDOMAIN(i, j)内の全てである	ファイルを全て削除する
Erroneous(E) in DOMAIN(i, j)	話し手は、聞き手が実体を同定できると考えているが聞き手は同定できない	a.cを消去したい (a.cがDOMAIN(\$current, directory) に見つからない)
Nonspecific(N) in UNIVERSE	実体は、UNIVERSE（全対象世界）に存在しない。 実体が不特定である。	ファイルを削除する（ファイルがない） ファイルを消す方法は （対象名が示されていない）

3. 構文意味解析部の概要

構文意味解析部は、ユニフィケーション文法に対する拡張LRパーサーである。出力される属性構造は必ずしもコマンド自動実行のための完全な知識を含んでいないので、何回かの対話の結果、条件が満足される場合も多々ある。このインクリメンタルな知識構築が文脈知識上で行われ、情報修正や情報付加に対しても属性構造は再度ユニファイされる。また、構文意味解析部で作り上げられたf-構造が、応答文生成部に中間表現として送られる。

ここでf-構造の例を示す。例えば「ファイルa.cをb.cにコピーして下さい」の入力に対して、f-構造は図2のようになる。図の属性"command"はユーザーの要求する概念を示しており、また属性"d_class"は要求に応じてコマンド実行、コンサルティングなどを行うためのユーザーの要求クラス^[23, 25]を示している。

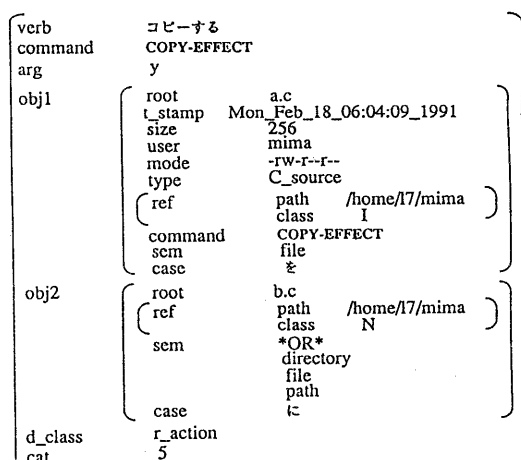


図2 f-構造例1

また、「ファイルa.cをコピーして下さい」の入力に対してはobj2が図3のように変わる。

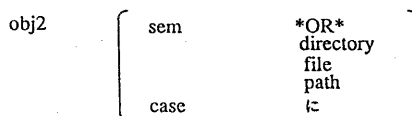


図3 f-構造例2

この場合、obj2のreferentの指定が入力文にないことが分かるので、再入力による情報補充のために、上記

の属性構造は文脈知識に一時的に格納される。また、
 “ファイル”が“すべて”と共に起きたときには、reference_classの分類はATとして設定される。

4. コマンド知識と意図理解

4. 1 コマンド知識

本システムでは、知識の表現形式として、UC^[9]のKODIAK(Keystone to Overall Design for Integration and Application of Knowledge)システムの考え方を利用する。KODIAKとは、関係理解の知識表現システムで、概念は最小の要素(absolute: グラフにおいて四角のノードで表現される)と考えられ、知識は概念が関連する関係(aspectual: 丸のノードで表現)としてとらえられている。そしてそれらは、リンクによる制約で関係づけられる。以下にそのリンクの概要を示す。

・dominate (D) link

概念間の上位・下位の関係を表現する。下位の概念は、上位概念の役割を継承する。

・constrain (C) link

アスペクトを定型的なものに束縛する(型を表わす)。

・instantiate (I) link

概念と、その実体を表現する。

・value (V) link

アスペクトを概念に結び付ける値を表現する。

・equate (=) link

アスペクトが等しいことを表現する。

ICIで用いるコマンド知識とは、正しいコマンドを生成するために、コマンド理解部から参照される知識ベースである。それは、すべてのコマンドについて用意されており、コマンド理解部から送られるコマンド、及び引き数の情報を用いて、正確なコマンドを生成する。また、引き数に不足情報や誤った情報があれば、それをフラグにしてコマンド理解部へ送る。

ICIの目的の一つがソフトウェアの保全にあるので、「a. cをb. cにコピーして下さい」の入力に対して、単に“cp a.c b.c”を生成・実行すれば良いものではない。この場合には、ファイル a.c の実体

が存在することやファイル b.c の実体が存在しない制約条件を満たしないと、コマンドは実行できない。

また、不完全入力に対してもICIは正しいコマンド実行を導く補足入力を催促しなくてはならないので、コマンド実行のためのreferenceも含めた制約条件を保有するコマンド知識が必要となる。UCではコンサルティングが目的であるので、コマンド知識にはreference情報は不要であるが、本研究では実状況知識も考慮したコマンドの知識表現を構築する。なおコピーコマンドに関する知識表現例を図4に示す。コピーコマンドにはコピーを含めて4種類あり、それぞれの引き数の制約条件が記述されている。この知識ではファイルからファイルへのコピーのフォーマットについて定義しており、引き数1と引き数2のパス名が違うこと、また、引き数2のファイル名の実体は同定できないものであるなどの制約が定義されている。

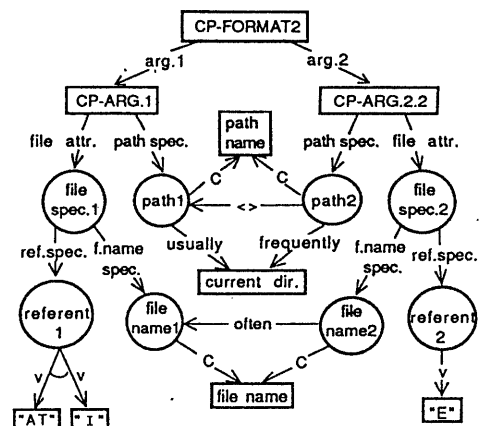


図4 コマンド知識の例1(cp-format)

4. 2 意図理解

ユーザーとシステムとの対話において、ユーザーの意図する内容が直接表層文中に表れている場合、その発話文よりコマンドを同定することは簡単である。しかし、そのようなコマンドに関する表現が発話内に隔に表現されていない場合、もはや表層的な意味解析だけではユーザーの意図の解釈は不可能となる。例えば次のような文、

「ファイルのモードを変更したい」

では、ユーザーの表現する意図とシステム内のコマンドが一意に対応するが、

「他人に読まれたくないファイルがある」

「newfile.cは勝手に消されたくない」

などの表現では、その意図するコマンドは表層文中には示されてはならず、さらに次のような文、

「プロセスが消えている」

では、表層に現れる主動詞「消す」とは全く逆の表現である「消したくない」がユーザーの真の意図であると思われる。このように、発話に表れる言語表現に発話者の意図がすべて明示的に示されているとは決して言い難い。

そのような問題に対処するためには、文中に表れるテンス・アスペクトおよび、ムード表現^[2,6]に注目し、発話文中にかくされた意図を正確に抽出する必要がある。テンス・アスペクト・ムードとはそれぞれ、動作や状態の発話者・認定者に対する時間的關係（テンス）、動作・作用の過程の焦点（アスペクト）、および、助動詞・終助詞によって表される断定・推量・命令・勧誘等の話し手の心的態度（ムード）を表す動詞の形態論的、意味論的あり方である。それらの分類をもとに、質問応答における既存のコマンドに対する一般的な発話文350例（内訳は表2参照）について、意図抽出に特に関係があると思われるテンス・アスペクトおよびムード表現を分類し、その意図として表3のような関係情報を考えた。

本章では、発話文を解析することにより得られるこれらの情報を、意図リンクとして定義し、その意図リンクにより関連づけられたコマンド知識表現を提案する。さらに、その知識表現から、発話者の真に意図するコマンドを推論する手法を述べる。

表2 発話文の内訳とその例

分類	発話例	数量
表層より意図が直接推論できる	ファイルをコピーしたい プロセスを消したい	231 (11%)
実状況によるコマンドの推論が必要	lang?を消したい (lang?: file or dir etc.)	51 (15%)
意図が暗に表現されていない	読まれたくないファイルがある ファイルを勝手に消されたくない	53 (15%)
意図が示されておらず、かつ実状況による推論が必要	文書が印刷できない (実状況により環境設定、デモン起動などが考えられる)	14 (4%)

表3 意図表現の分類と意図リンク

テンス・アスペクト・ムード	表層表現	意図リンク
結果残存	ている、でいる	contrast
動作の実現	てしまう、でしまう	
経緯・完了	てしまった	impossible
不可能	できない	
否定	ない	deny
完了（否定）	なかった	
禁止	な	refuse
希望（否定）	たくない、ほしくない	
必要・義務	ならない、いけない	must

4. 3 意図リンクによるコマンド推論

KODIAK^[1,2]に基づく知識表現に対して、意図リンクにより拡張を行ったコマンド知識の例を図5に示す。この例の知識では、EXECUTE, WRITE, READのEFFECT概念があり、消去することがWRITE概念の下位に当たること（DELETEは何もない情報を書くの意味）、コマンドを実行することによりそれぞれの効果が表れること、さらにEXECUTE, WRITE, READのEFFECT概念から意図リンクによるPERMISSION-EFFECTへの関連が定義されている。

まず、ユーザーの発話文を形態素解析し、テンス・アスペクト・ムード表現を抽出する。そして、表3より決定した意図情報を付加する。さらに、それを構文・意味解析し、ユニフィケーション文法による属性構造を得る。その時点で、概念EFFECTを獲得することになる（図2参照）。

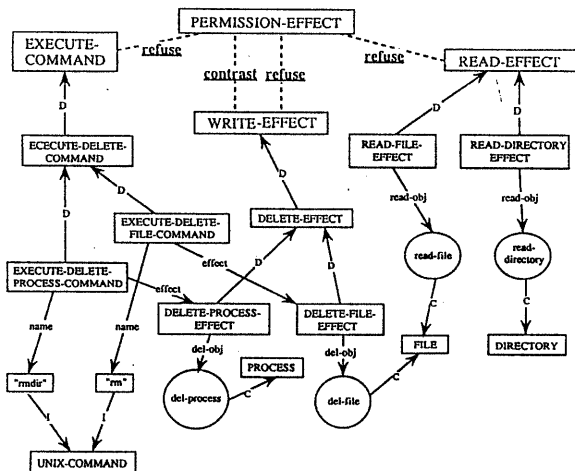
そして、次のようなアルゴリズムによりコマンドの推論を行う。

アルゴリズム

ステップ1: ユーザーの発話文 s に対して、EFFECT_VALUE(s)の値が一つに同定されるまで、ユーザーとの対話によって絞り込む。ここで、EFFECT_VALUE(s)の値を x とする。

ステップ2: 入力文 s に対する意図を表3より決定し、意図リンク INTENSION(s)を y とする。

ステップ3: y が null の場合（表3で決定できない）は、ステップ6へ進みコマンドの決定を行う。そうでない場合は次のステップへ進む。



ステップ4: x より上位概念リンク (Dリンク) を辿り、
 y の値を持つリンク (意図リンク) があれば遷移する。 y のリンク先の EFFECT を z とする。

ステップ5: z よりさらにDリンクを逆に辿り(下位概念),
 入力文 s に対する f -構造の情報を使って新
 しいEFFECTに移移する. その値を w とする.

ステップ6: w より effectリンクを逆遷移し、概念EXECUTE-COMMANDへと遷移する。この時点でコマンドの同定は一応完了するが、EXECUTE-COMMANDに関係づけられている format等のコマンド知識と実状況との関連により ステップ5 へのバックトラックまたは、ユーザーとの対話処理が起こり得る。

以下に発話文「ファイルを消されたくない」を例にし、そのコマンド推論のプロセスを上記のアルゴリズムに基づいて詳述する。

入力: s = " ファイルを消されたくない "

ステップ 1: $x = \text{EFFECT_VALUE}(s)$ より, $x = \text{DELETE-FILE-EFFECT}$ を得る.

ステップ2: $y = \text{INTENSION}(s)$ より意図リンク `refuse` を得る.

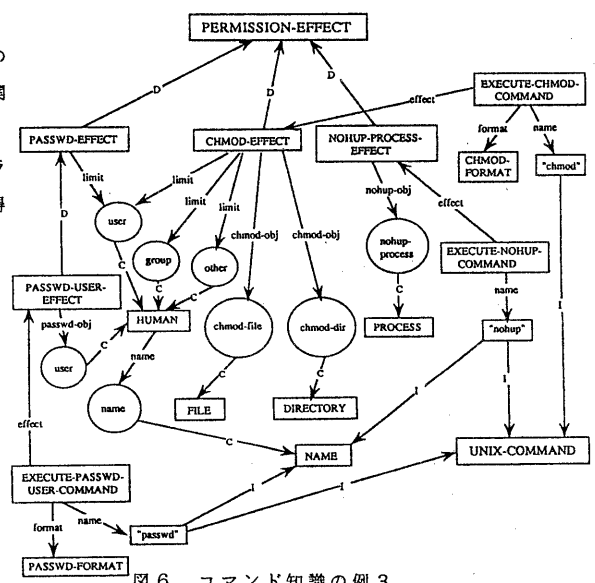
ステップ3: $y \neq \text{null}$ より次のステップへ.

ステップ 4: **x** より **D** リンクを送り, **WRITE-EFFECT** で意図リンク **refuse** と **y** がマッチングする。よって, その遷移を送り **z** = **PERMISSION-EFFECT** を得る。

ステップ5: z より再びDリンクを逆に辿り、入力文 s に対する f-構造の obj-FILEより CHMOD-EFFECT に遷移する (図6参照).

よって $\Psi = \text{CHMOD-EFFECT}$ となる.

ステップ6: ■ より effectリンクを逆遷移、意図するコマンドを同定する。この時点より FORMAT情報などを用い、コマンド"chmod"の実行に満足すべくユーザーとの対話とコマンド作成処理を行う。



5. 応答文生成部

以下、応答文生成部の流れを例文に従って説明する。例として「ファイル a. c を消去して下さい」の入力に対する理解プロセスを説明する。応答文生成部では、構文意味解析部からの中間表現である f-構造に対して、まずコマンド理解部でコマンド実行が可能かどうかをチェックする。その際、rm-commandの引き数 "a.c" の属性情報を格納する。次にそれぞれの変数をコマンド知識に送り、EXECUTE-DELETE-FILE-COMMAND のコマンド知識における format 情報の引き数に対応させ、それぞれの引き数の属性情報をチェックする。その結果、実行可能ならば "rm a.c" を生成し、それをコマンド実行部で実行する。また「消去したい」などの入力に対しては、引き数の実体名が存在しないのでコマンドは実行不可能である。この場合、コマンド知識において DELETE-EFFECT までは同定されるが、その EFFECT の対象がファイルかディレクトリかプロセスかなどによる最下位の EFFECT 概念に曖昧性が残る。よって文生成知識から適切な文を生成し、ユーザーとの対話による決定を行う。そして、その決定後 EXECUTE-DELETE-FILE-COMMAND の format 情報における第 1 引き数の要求が処理される。

6. 現在のシステムと今後の課題

現在、UNIX システムとワードプロセッサを対象として約 50 個のコマンドに対する実験を行っている。システム全体は、辞書管理モジュール等を含めると約 2 万行の C 言語で記述されている。稼働マシンは、Sun/4 である。今後の課題としては、

- (1) ユーザーレベル（初心者、熟練者）に対するユーザーモデル^[29]とコマンド知識の関連づけ。
 - (2) コマンド間の因果関係による連続コマンド実行への拡張。例えば、モードが原因でコピーできない場合は、モード変更コマンドを実行した後コピーの処理に移る。
 - (3) コマンド知識の効率的獲得手法の確立
- などきりが無いが、一つの大きな課題は、本研究の ICI をどのような情報システムにも組み込めるように、

解析・理解モジュールと関連する三つの知識ベース（語彙知識、実状況知識、コマンド知識）に対して、情報システムとの独立性と従属性を明確にし、形式化する点である。そして、独立モジュールは専用化（固定化）し、従属モジュールや知識ベースは独立モジュールとのインタフェースを定義したうえで、構築するための支援ルーチンを開発する必要がある。

また、日本語以外の入力に対する各モジュールの独立性と従属性についても考慮し、多言語に対する一般化についても考慮することは興味深い研究である。

7. むすび

自然言語による入力手法はコマンド学習を目的とする場合でも、段階的な学習状況に対する柔軟な入力処理が可能とあるので、極めて有望である。特に、将来制限付きでも音声認識が実用化されるようになれば、自然言語による実行要求は相乗の効果を発揮するはずである。

今後は、現在運用している UNIX やワードプロセッサのように多くのコマンドを持つ情報システムへの段階的な試験使用を重ねて実用化に進める計画である。

謝辞：本研究は民間企業との共同研究、文部省科学研究費一般研究（C）、電気通信財団、マツダ財団、人工知能財団の援助を受けた。

文 献

- [1] J.Aoe, J.S.Lui, Y.S.Alam, R.Cohen and T.Sato, "Coherent analysis and intelligent machine translation", In Proc. of COMPSAC'88, Chicago USA, pp.463-471, Oct.1988.
- [2] J.Aoe, "Intelligent machine translation using a contextual knowledge representation," In Proc. of Workshop on Tools for AI, Fair Fax USA, pp.145-152, Oct. 1989.
- [3] J.Aoe, *Computer Algorithms-Key Search Strategies*-, IEEE Computer Society Press., 1991.
- [4] J. Aoe, "An efficient digital search algo-

- rithm by using a doublearray structure,"
IEEE Trans. Software Eng., SE-15, 9, pp.1066-1077, Sept. 1989.
- [5] 青江, Alfred Maeda, 久次米, 森本, "自然言語入力によるコマンドの理解," 電子情報通信学会言語理解とコミュニケーション研究会, 1990年12月
- [6] 青江, Alfred Maeda, 久次米, 森本, "自然言語入力に対する知的コマンドインターフェイスにおける統合処理," 自然言語処理における統合処理シンポジウム, 1991年1月
- [7] Alfredo Maeda, J.Aoe, K.Morimoto, "An Intelligent Command Interpreter for Consulting and Use of Information Systems," International Symposium on Artificial Intelligence, Nov., Mexico Cancun pp.533-539(1991)
- [8] S. J. Kaplan, "Designing a portable natural language database query system," ACM Trans., Database Syst., 9, 1, pp.1-19, Jan. 1984.
- [9] K. Knight, "Unification: A multidisciplinary survey," ACM Computing Surveys, 21, 1, pp.93-124, Jan. 1989.
- [10] S.Yasutome, A.Masanao and J.Aoe, "Parallel semantic disambiguation for unification-based grammars", In Proc. of PRICAI'90, Nagoya Japan, pp.292-297, Nov. 1989.
- [11] K. Takeda, D.N. Chin and I.Miyamoto, "MOANA: A software engineering tool with natural language user interface, In Proc. of PRICAI '90, Nagoya, pp.897-902, Nov. 1989.
- [12] R. Wilensky and D. N. Chin, "The berkeley UNIX consaltant project", Computational Linguistics, 14, 4, pp.35-84, Dec. 1988.
- [13] 河村一樹, "形式化された自然言語によるリレーショナルデータベースの問い合わせについて," 情処研資データベースシステム, 64-7, 1988.
- [14] 木下, 加納, 高橋, 小林, "文献データベースの知的インターフェイスとその性能評価," 情処研資データベースシステム, 72-20, 1989.
- [15] 絹川博之, "表階層モデルに基づく自然言語インタフェイス処理方式," 情処論, 27, 5, pp.499-506, 1986.
- [16] 佐野, 福本, "単一化に基づく文脈・意味処理," 情処研資自然言語処理, 78-7, 1990.
- [17] 佐藤, 斉藤, 菊池, "特許情報検索のための日本語質問文解析," 情処論, 25, 3, pp.365-371, 1984.
- [18] 竹本, 田口, 中川, "質疑応答システムにおける日本語文の理解と生成," 情処研資自然言語処理, 74-13, 1989.
- [19] 中川, 加藤, "日本語データベース検索システムにおける意味理解方式," 情処論, 27, 11, pp.1069-1076, 1986.
- [20] 永田, 久米, "単一化に基づく枠組みにおける日本語対話文解析用文法の記述とその計算的側面," 情処研資自然言語処理, 76-1, 1990.
- [21] 樋口, 伊藤, "対話型UNIX支援システムの機能と構成," 情処平成2年後期全国大会, 5S-2, pp.3-171~3-172, 1990.
- [22] 福本, 佐野, "情報伝達に基づく文脈処理へのアプローチ," 情処研資自然言語処理, 78-8, 1990.
- [23] 横本, 垣内, 上原, 豊田, "対話型システムにおける文脈情報を利用した文章生成について," 情処研資自然言語処理, 74-13, 1989.
- [24] 山岡, 飯田, "文脈を考慮した音声認識結果絞り込み法," 情処研資自然言語処理, 78-16, 1990.
- [25] 吉浦, 片山, 中西, 平沢, "日本語質問応答システムにおける質問の曖昧性を解消する意味解析方式," 情処論, 27, 3, pp.321-329, 1986.
- [26] 情報処理進行事業協会技術センター, "計算機用日本語基本動詞辞書 I P A L (Basic Verbs) - 解説編 -, 昭和62年2月
- [27] 酒井, 池田, 藤田, "ガイドンス機能を備えた自然言語インターフェースシステム(2)-発話意図抽出処理-, 情報処理学会第42回全国大会, 1991
- [28] 安西, 神岡, "自然言語理解の構造-理解のメカニズム: 対話と文脈," 情処論 Vol.30, No.10, pp.1150-1160, Oct.1989
- [29] 加納, 岸野, "ユーザーモデルを用いた知的文献検索インターフェース," 電子情報通信学会論文誌 '91, 8, Vol. J74-D-I No.8