

汎用 DBMS を用いた知的マン・マシン インタフェースの実現†

—音声データベース SPEECH-DB に関して—

溝口 理一郎** 芥子 育雄***
磯本 征雄**** 角 所 収**

近年、マン・マシンインタフェースの問題が盛んに論じられている。本論文では、筆者らによって作成された音声データベース「SPEECH-DB」を例にとりて実現した知的マン・マシンインタフェースについて述べている。本インタフェースは、日本語による検索要求を受けつける JCS サブシステム、エラーからの回復や利用者の疑問に答える PSS サブシステムおよびシステム全体の制御の流れを管理する MONITOR サブシステムの三つのサブシステムから成っている。知的マン・マシンインタフェースは、情報処理システムに関する専門知識を用いることにより問題解決を行って、利用者に対して知的な援助を提供するものであり、ある種のエキスパートシステムとみなすことができる。そこで本インタフェースの作成においては、汎用 DBMS、INQ とその親言語である FORTRAN によりプロダクションシステムを実現した後、それに種々の知識を埋め込むという方法を採用した。SPEECH-DB と同一の DBMS、INQ を用いることにより、柔軟性、拡張性に富んだシステム構成が実現された。われわれのプロダクションシステムは、前向き推論と後向き推論の二つの推論モードとバックトラック機能をもっている。さらに、任意の FORTRAN プログラムを呼ぶことができるため、既存のソフトウェアとの結合が容易であり、AI 的手法の現実的問題への応用に適したものとなっている。

1. ま え が き

近年、汎用 DBMS の普及によりエンドユーザによるデータベースシステムへの検索の機会が増えている。このような情報処理システムの進歩は、今後計算機の専門家以外の利用者の活動をも大いに支援するであろう。しかしながら、これらの情報処理システムは一部の専門家以外には十分に有効利用されていないのが現状である。この原因は、マン・マシンインタフェースすなわち人間と情報処理システムとのインタフェースに存在するものと思われる。一般に人間から情報処理システムへの意思表示はコマンドによってなされる。このため通常の情報処理システムを利用する場合複雑で融通の効かないコマンドを用いることを余儀なくされている。人間と情報処理システムのインタフェースにおいて、この問題を解消するためには、人間同士の会話のように、構文的な誤りあるいは意味的な曖昧さがある場合においてもうまく処理できる能力を

もち、かつ自然言語も使用可能な高度なマン・マシンインタフェース^{1),2)}の整備はきわめて重要である。そこで本研究では、筆者らにより作成された音声データベース「SPEECH-DB」³⁾を例にとり、上述のような機能をもつ知的マン・マシンインタフェースを、汎用 DBMS を利用することにより実現した。

知的マン・マシンインタフェースは、情報処理システムに関する専門知識を用いることにより問題解決を行って、利用者に対して知的な援助を提供するものであり、エキスパートシステム^{4),5)}の一種とみなすことができる。そこで本インタフェースの作成においては、まず汎用 DBMS と FORTRAN によりプロダクションシステム^{6),7)}(以後 PS と略す)を実現した後、その PS に種々の知識を埋め込むという方法を採用した。汎用 DBMS の利用目的は以下のとおりである。

- (1) SPEECH-DB と同一の汎用 DBMS を利用することによる柔軟性、拡張性に富んだシステム構成の実現。
- (2) 大量情報記憶機能を利用した、利用者との対話および問題の解決(エラー処理)という本インタフェースの基本的手続き(知識)の格納。
- (3) 高速検索機能により、データベースに格納された必要な手続きを検索しつつ処理を進めるとい

† An Implementation of Intelligent Man-Machine Interface Using DBMS—On the Speech Database SPEECH-DB— by RICHIRO MIZOGUCHI (I. S. I. R., Osaka University), IKUO KESHI (Faculty of Engineering, Osaka University), YUKUO ISOMOTO (Computation Center, Osaka University) and OSAMU KAKUSHO (I. S. I. R., Osaka University).

** 大阪大学産業科学研究所

*** 大阪大学工学部電子工学科

**** 大阪大学大型計算機センター

* 現在 シャープ(株)

‡ ソフトウェアは元来計算機の利用技術を意味するものであるが、ソフトウェアの巨大化の傾向にある現在では、ソフトウェア自身の利用技術が問題となることは興味深い。

う PS 流の新しいデータベースの利用法の実現。本論文では、汎用 DBMS による知的マン・マシンインタフェースの実現に関してその詳細を述べる。

2. 音声データベース SPEECH-DB

本章では、本インタフェースを実現する上で対象とした音声データベース SPEECH-DB⁹⁾ について概説する。

2.1 SPEECH-DB の基本構成

音声は図 1 に示すような階層構造をなしていると考えられる。ここで C, V はおのおの子音, 母音を表す。この音声の階層性を効率よく実現するために SPEECH-DB は、音韻, CV 音節, VCV 音節, 単語, 環境, 生データの六つの独立したファイルから構成されている。図 2 にその基本構成を示す。DBMS として、日本電気の INQ⁹⁾ が用いられている。生データファイルは普通のランダムファイルで、サンプリングされた種々の音声データが格納されている。他の五つのファイルは INQ ファイルと呼ばれ、音声データを検索するために必要な情報が格納されている。

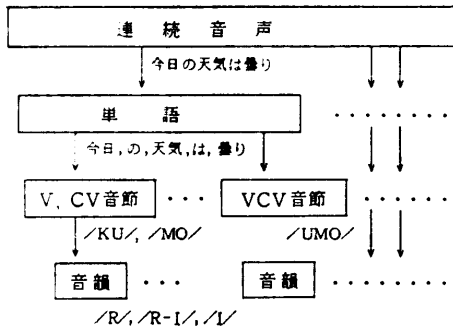


図 1 音声の階層構造
Fig. 1 Hierarchical structure of speech.

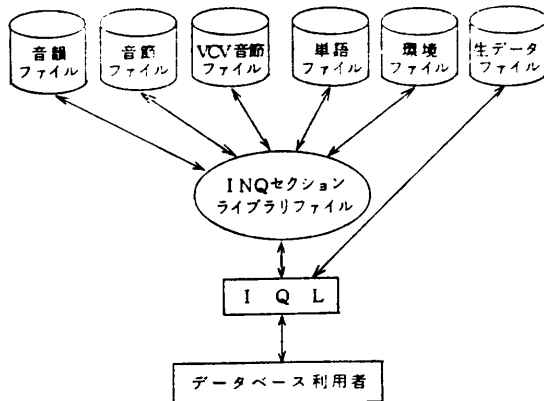


図 2 SPEECH-DB の基本構成
Fig. 2 Basic structure of SPEECH-DB

INQ では INQ セクション (外部スキーマ) を定義することにより上記五つの INQ ファイルを結合することができ、種々の論理構造を実現することができる。INQ セクションライブラリファイルには連続音声の中のさまざまな前後関係のなかでの音韻, 音節, 単語等を検索できるように種々の外部スキーマが定義されている。検索用コマンド IQL はこの外部スキーマの自動切換え能力を有している。したがって、SPEECH-DB の利用者はこの知的コマンド IQL を用いることにより、データベースの論理構造を意識することなしに検索条件を羅列するだけで検索が行える。

2.2 検索コマンド

IQL における検索は、音声の生データファイルにおける始点と終点を知るための操作と、その位置情報を用いた生データの分析・表示あるいは転送のための操作の 2 段階の操作から成っている。さらに第 1 段の操作は、検索条件を満たすレコードの検索とその項目値の表示の 2 段階に分かれている。本節では検索に用いられる最も重要な SEARCH コマンドについて述べる。以下に具体例を示す、「男の発声で無声破裂音を含む単語が欲しい」は次のように翻訳される。

CONDITION WORD; SUBJECT-SEX

(次の条件の) 単語が欲しい 男の発声で

= MALE AND

(かつ)

SYLLABLE= VOICE-LESS-PLOSIVE + \$

無声破裂音を含む

省略形を使うと次のようにも書ける。

COND WD; SS= MALE AND SY= VLP + \$

COND は次に検索条件が続くことを意味し、WD は検索対象を表す。IQL では検索対象として連続音声 (CS), 単語 (WD), VCV 音節 (VCV), CV 音節 (SY), 音韻 (PH) の 5 種類が想定されている。“;”以降に検索対象が満足すべき条件が書かれるが、順序は任意であり、一般的には「項目名 比較オペレータ 項目値」の形をしている。音声には図 1 に示すような階層構造があるが、IQL でもこの階層性を想定しているため、 $PH < (SY, VCV) < WD < CS$ (ただし、 $\alpha < \beta$ は β が α の上位にあることを表す) なる関係が暗黙に仮定されている。したがって、

SY; WD= \$BAS

と書けば /BA/ を含む単語に含まれる音節を意味し、

SY; PH= A

と書けば音韻 /A/ を含む音節を意味する。音声の専門用語を項目値として使うことも許されており、たと

例えば有声破裂音 (voiced plosive) を使うと

```
SY=VOICED-PLOSIVE+A
```

```
SY=VP+A
```

```
SY=BA OR DA OR GA
```

の三つの条件は同じ意味となる。また、“+A”は後続母音を /A/ と指定することを意味し、後続母音を問題にしない場合には“+&”と入力する。さらに“[”, “]”で囲まれた区間の条件を1回の検索条件とする多重検索も行える。

3. 汎用 DBMS によるプロダクションシステムの実現

本章では、本インタフェース実現のためのツールとして開発されたプロダクションシステム^{6),7)}について述べる。通常 PS は LISP でインプリメントされるが、処理速度が遅いことおよび他言語との結合ができないこと等の問題があり、われわれの問題には適していない。本研究では、プロダクションメモリをデータベース、ワーキングメモリの内容を検索要求とみなせば自然な対応が見つかることから、汎用 DBMS, INQ とその親言語である FORTRAN を用いてインプリメントした。したがって本 PS は、FORTRAN や OS の全機能を利用できることはもちろん、パターンマッチングによる手続き呼出しや、非決定的動きといった特徴をもつ新しい FORTRAN プログラミングシステムともなっている。また PS が本質的にもつ三つの特徴、1) modularity, 2) readability, 3) self-explanatory⁶⁾, も兼ね備えている。このような特徴をもった本 PS は大規模なソフトウェアを作る上でのプログラミング環境の改善にも有効であると考えられる。

本 PS ではワーキングメモリに現在の状態(属性名 属性値)の集合によって記述し、ワーキングメモリの書換えを行うプロダクションルールの集合プロダクションメモリをデータベース化している。プロダクションシステムインタプリタ (PSI) はワーキングメモリ内の記述を検索条件として現在の状態に適したルールを検索し、解釈・実行する。プロダクションメモリの論理構造は INQ に用意されているファイル記述言語 FDL を用いて図3のように記述されている。この INQ ファイル (SPEECH-DB の7番目のファイル) の各レコードがルールに対応し、ルールは ID コード、条件部 (LHS), 実行部 (RHS) および関数部から成る。関数部は、LHS 関数(名前が“*”で始まる)と RHS 関数から成る。

```
FDL PROD,7.
DATABASE SPEECH-DB.
02 ID-RULE PIC CP6 PKY.
02 CONDITION(N).
03 ATTRIBUTE-NAME1 PIC X(8).
03 ATTRIBUTE-VALUE1 PIC X(8) BY ATTRIBUTE-NAME1.
02 ACTION(N).
03 ATTRIBUTE-NAME2 PIC X(8).
03 ATTRIBUTE-VALUE2 PIC X(8) BY ATTRIBUTE-NAME2.
02 FUNCTION(N).
03 FUNCTION-NAME PIC X(8) DSP.
03 ARGUMENT(N).
04 ATTRIBUTE-NAME3 PIC X(8) DSP.
04 ATTRIBUTE-VALUE3 PIC X(8) DSP.
```

図3 プロダクションメモリの論理構造

Fig. 3 Logical structure of production memory

ID コード、条件部、実行部はともにキーとして登録されており、たとえば条件部に (ATR. VAL.) をもつルール等を検索することができる。一方、関数部に関してはインデックスは作られておらず、それを条件にして検索することはできない(図3における DSP は DISPLAY の略で、その項目が表示のためだけの項目であることを示す)。

関数部には手続き名が記述されており、ルールの実行はこの手続きをサブルーチンコールすることによって行われる。コンフリクトの解消はコンフリクトセットのなかから ID コードの最も大きいルールを選択することによって実現される。また、ルールの解釈実行の手順として、前向き推論と後向き推論の2種類のモードが用意されており、そのおのおののモードにおいてバックトラック機能がある。

a) 前向き推論

現在の状態にマッチする条件部をもつルールがデータベースから検索され、LHS 関数が真のとき発火して状態を書き換える。この操作が繰り返され、発火するルールがなければバックトラックを行いながら処理を進めていく。

b) 後向き推論

A→B→C のような多段階の推論を必要とする場合に、まず最終的に達成すべきゴールとして C を設定し、このゴールを満足する実行部をもつルール(たとえば B→C) がデータベースから検索される。この段階では検索されたルールを実行するためには B を達成しなければならない。そこで B をサブゴールとして設定し、それを満たす実行部をもつルールを検索する。この操作を検索されるルールが空になるまで繰り返す。最後に検索されたルールから順次実行する。

4. SPEECH-DB のための知的マン・マシンインタフェースの実現

情報処理システムは少し規模が大きくなり機能が豊富になると、マニュアルは大部なものとなり初心者には使いにくくなる。また、コマンドの数が増加すれば適当なものを選ぶのが困難となり誤った使用の可能性も増す。このように大規模処理システムほど利用に際して諸々の問題を解決し、利用者を支援するマン・マシンインタフェースが必要となる。SPEECH-DB は IQL により検索が容易に行えるように工夫されているが、初心者には事前に学ぶべき知識や制限があり、知的インタフェースとしては問題が残されていた。知的インタフェースに関しては Reddy ら¹⁾のグレースフルインタフェースなる提案があるが、筆者らは SPEECH-DB の利用に際して生じる諸々の問題を解決し、利用者を支援するマン・マシンインタフェースを実現した。

4.1 システム構成

本システム全体の基本構成を図 4 に示す。本インタフェースは次の三つのサブシステムから構成されている。

(1) 日本語コマンド処理サブシステム (JCS)

JCS は SPEECH-DB の自然言語インタフェースである。完全な日本語処理に基づくものではなく、JCS の能力が不足する部分、あるいは日本語検索文に意味的な曖昧さがある場合は利用者との対話により補っている。また、IQL に翻訳不可能な場合にはその旨を利用者に説明する機能をもっている。

(2) 問題解決サブシステム (PSS)

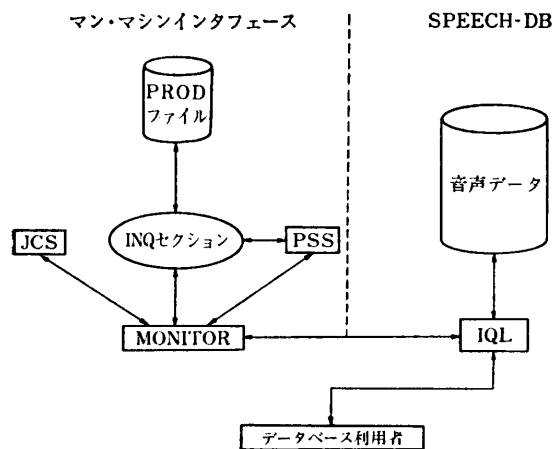


図 4 システム構成

Fig. 4 System configuration.

PSS は、エラー処理と問題解決機能をもつ。MAC-SYMA ADVISOR²⁾と同様、利用者が入力したコマンド系列をもとに利用者の検索コマンドに対する誤解を発見する。プラン認識は行わないが、あらかじめゴールまでの条件分岐をプログラムしたアドホックなシステムではない。PSS では SPEECH-DB の初心者の検索例を分析することにより、誤りの原因、結果に関する知識をデータベース化している。また、エラー処理、利用者との対話といった手続きもデータベースに格納されている。これらのデータベースに格納された知識によって利用者が入力したコマンド系列からゴールまでのパスを決定し、コマンドの診断・修正を行う。エラー処理ではエラー番号*というゴールが与えられるのでそのパスは容易に決定される。問題解決機能では、利用者の誤解がゴールとなりゴールを設定することができないので、利用者が入力したコマンド系列をもとに試行錯誤的な探索を行いながらゴールへのパスを探索する。

(3) MONITOR

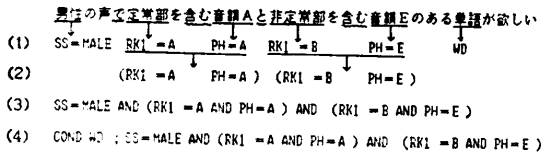
MONITOR は、本システムにおける制御の流れを管理する。具体的には PSS が修正したコマンドを IQL へ送って検索を行ったり、利用者との簡単な対話機能をもつ。

4.2 日本語コマンド処理サブシステム (JCS)

日本語での SPEECH-DB への検索要求を受け付け、IQL に翻訳する。また、翻訳結果を提示し、利用者が自然に IQL を学習できるように配慮されている。JCS はあくまで初心者用であり「使い慣れさえすればコマンドのほうが自然言語より使いやすい」というのが筆者らの基本的立場である。

IQL は当初より自然言語インタフェースを作成することを想定して設計されたため、検索条件を羅列するだけのコマンド言語であること、さらに対象とする日本語文が SPEECH-DB への検索要求というきわめて強い制約を受けていることから、JCS では構文解析はほとんど行われておらずキーワードの探索と置換えを中心とした処理が採用されている。情報が不足する場合や曖昧な部分がある場合には利用者にお問い合わせにより補っている。例 1 に日本語検索文から IQL への翻訳過程の例を示す。実際の検索文は単語単位に分ち書きされたローマ字で入力される。翻訳過程は次の 4 段階の処理から成る。

* コマンド処理におけるエラーのチェックは INQ がもつエラー検出機能に依存している。



例1 翻訳過程の例
Example 1 Translation process.

```

PLEASE SPEAK IN JAPANESE !
=KAZE O HIITA SEIJIN DANSEI NO BION O FUKUMU
=ONSETU GA HOSHI
*****
*** WARNING ***
*****
'KAZE HIITA' CANNOT BE RETRIEVAL CONDITIONS
      IN 'SPEECH-DB'.

*****
TRANSLATION INTO IQL
*****

COND SY=NS+* ; SAG>=20 AND SS=MALE
    
```

例2 JCS が翻訳不可能なことを説明する例
Example 2 Indication of the difficulty of translation.

- (1) 辞書を用いて IQL の検索条件 (項目名 比較オペレータ 項目値) の形に変換する。
- (2) 同一 INQ ファイル内の項目名が三つ以上あるときは、「で」、「と」、「含む」等をキーワードとして修飾関係を見つける。
- (3) 「あるいは」等のキーワードがないときは、すでに変換された検索条件を「AND」で結合する。
- (4) (1) で生成されたものの内最後の項目が属する INQ ファイルによって検索対象を決定し、完全な IQL に変換する。

意味的に曖昧な場合、たとえば「8以下の音節」という検索文では「8」が何を意味するかわからないため、そのことを問い合わせ、もしそれが「ピッチ」(音声の基本周期)を意味することが教えられたとすれば、さらにそれが最大値、最小値、平均値のいずれであるかを問い合わせることにより曖昧さを解消することができる。現在 JCS への入力文は、代名詞を含まない一つの文章で、原則として IQL に変換可能な内容をもつものであることが仮定されてはいるが、JCS は IQL に翻訳不可能な場合にはそのことを利用者に説明する能力をもっている。この機能は図5に示すような2種類の単語辞書をもつことにより、それ以外の単語が検索文中に表れた場合に IQL に翻訳できないことを知らせることによって実現されている。その例を例2に

示す。IQL へ変換できる部分については翻訳を行っている。

4.3 問題解決サブシステム (PSS)

IQL の使用に際して起こりうる誤りには、IQL の構文上の誤り (エラー番号が出力され検索は行われない) と意味上の誤り (一応検索は行われるが結果は意図したものとは異なる) の2種類がある。PSSはこのような誤りを、利用者が入力したコマンド系列をもとに利用者との対話により解決する。本節では、PSSの使用例とともに、その概略について述べる。

4.3.1 IQL の構文上の誤り

構文上の誤りに対する推論規則の例を図6*に示す。PSSの処理モードは、この場合エラー番号というゴール状態が与えられるため、後向き推論モードが用いられる。IQLの構文はそれ自体単純であるので、利用者が入力したコマンド系列と IQL が出力したエラー番号をもとに2段の推論、すなわちサブゴールを一つ設定することにより誤りの原因を判別することが可能である。後向き推論の方法は、ゴール状態を満足するす

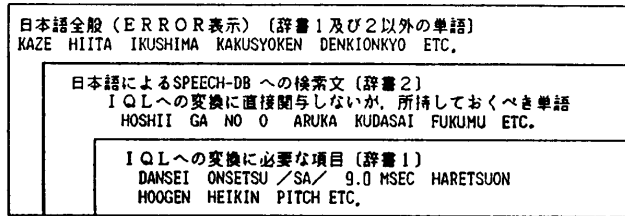


図5 JCS における2種類の単語辞書
Fig. 5 Two kinds of dictionaries of JCS.

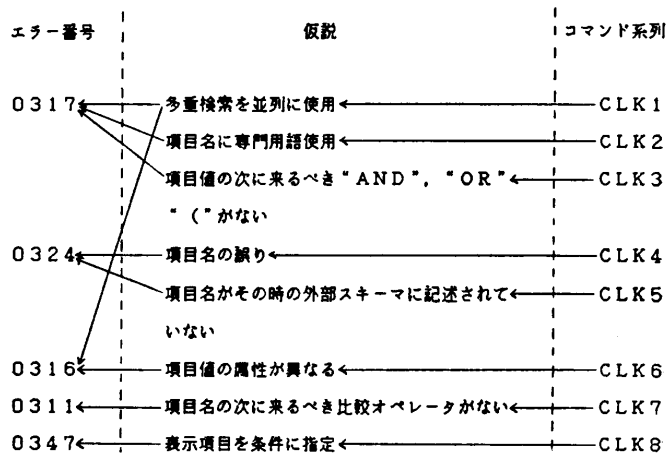


図6 構文上の誤りに対する推論規則の例
Fig. 6 Some examples of inference rules for syntactic errors.

* 多重検索の並列使用の場合、そのときの外部スキーマに依存して 0316 と 0317 の二つのエラー番号が出力される。

IR=1	IF	多重検索を並列に使用	THEN	エラー-0317が起る
IR=2	IF	項目名に専門用語使用	THEN	エラー-0317が起る
IR=3	IF	項目値の次に来るべき "AND", "OR", "(" が無い	THEN	エラー-0317が起る
IR=4	IF	項目名の誤り	THEN	エラー-0324が起る
IR=5	IF	項目名がその時の外部スキーマに記述されていない	THEN	エラー-0324が起る
IR=6	IF	多重検索を並列に使用	THEN	エラー-0316が起る
IR=7	IF	項目値の属性が異なる	THEN	エラー-0316が起る
IR=14	IF	L-func (コマンド系列に "[" "] " が2組以上並列にある)	THEN	多重検索を並列に使用 R-func (" ["] " が 1組のみの複数のコマ ンド系列に分解する)
IR=15	IF	L-func (コマンド系列において比較オペレータの前に専門用語がある)	THEN	項目名に専門用語使用 R-func (その部分の検索条件をJCSで入力させる)
IR=16	IF	L-func (コマンド系列に項目値の次に項目名がきているものがある)	THEN	項目値の次に来るべき "AND", "OR", "(" が無い R-func (項目値の次に "AND", "OR" を 入力させる)
IR=17	IF	L-func (コマンド系列にIQLの辞書にない項目名がある)	THEN	項目名の誤り R-func (項目名をJCS で入力させる)

図 7 構文上の誤りに対する規則の例

Fig. 7 Some examples of production rules for syntactic errors.

すべてのパスを求めた後、コマンド系列→仮説のルールから実行する。このルールのLHS関数がコマンド系列に対してその仮説が正しいかどうかを評価し、正しいときはRHS関数によって構文上の誤りの診断および治療を行う。用いられた規則の例を図7に自然言語表現を用いて示す。

IQLはエラー番号を検出すると制御をMONITORに移し、MONITORがエラー番号をゴールとして

```

1 SYSTEM SPEECH-DB
  COMMAND ?
2 =SEARCH
  TYPE IN SUBCOMMAND
3 =COND WD<EAP>EAP [ID=11] AND EAR>EAR [ID=11]
4 ***SYNTAX ERROR***
  COND WD EAP > EAP [ ID = 11 ] AND EAR > EAR [ ID = 11 ]
  DO YOU UNDERSTAND WHAT IS WRONG? (YE OR NO)
  =NO
  ***YOU CAN NOT USE PARALLEL MULT: RETRIEVAL***
  ***CORRECTED SUBCOMMAND SEQUENCE IS AS FOLLOWS:***
  COND WD ; EAP > EAP [ ID = 11 ]
  SAVE
  COND WD ; EAR > EAR [ ID = 11 ]
  SAVE
  AND 1 2
5 SHALL I EXECUTE THE CORRECTED SUBCOMMAND SEQUENCE?
  (YE OR NO)
  =YE
  1. COND WD ; EAP > EAP [ ID = 11 ]
    8 RECORDS FOUND.
  2. SAVE
    SAVED FILE ID NUMBER = 7
  3. COND WD ; EAR > EAR [ ID = 11 ]
    6 RECORDS FOUND.
  4. SAVE
    SAVED FILE ID NUMBER = 8
    5. AND 7 8
    4 RECORDS FOUND.
6 TYPE IN SUBCOMMAND

```

例 3 多重検索を並列に使用したときの PSS の使用例
Example 3 Dialog in the case of misuse of multiple retrieval

PSS を起動する。例3と例4に PSS と利用者との対話例を示す。下線部が利用者によって入力された部分を示している。また、IR は規則の ID 番号であるが、{ } 内は、おのおのの部分で発火した規則の ID 番号を示す。

【例 3】

① SPEECH-DB を起動して、IQL コマンドの入力待ちの状態である。

② SEARCH コマンドを入力し、サブコマンドの入力待ちの状態である。

③ COND サブコマンドを入力し検索を行っている。検索条件は「平均ピッチ (EAP) と平均エネルギー (EAR) が、音声の ID コード (ID) が 11 のものより大きい単語 (WD)」である。「[', ']' で囲まれた区間は 2 重検索を表している。IQL で

は 2 重検索のネスティングは許しているが、それを並列に使用することを認めていないので、エラー番号 "0316" が検出され制御が MONITOR に移され、PSS が起動される。

④ {IR=6, 14} IR=6 のルールが検索され、WM

```

TYPE IN SUBCOMMAND
(1) =COND WD<SS>MALE AND VP=A
  PREVIOUS VOWEL ?
  =
  FOLLOWING VOWEL ?
  =A
(2) ***SYNTAX ERROR***
  COND WD ; SS = MALE AND VP = A
  DO YOU UNDERSTAND WHAT IS WRONG?(YE OR NO)
  =NO
  ***YOU CAN NOT USE A TECHNICAL TERM FOR ATTRIBUTE NAME***
  CAN YOU INPUT THE CORRECTED ATTRIBUTE NAME?(YE OR NO)
  =NO
  I WILL DIAGNOSE THE PART:
  VP = A
(3) DO YOU THINK THE FOLLOWING CONDITION MATCHES
  WITH YOUR INTENTION?(YE OR NO)
  ** YOU WANT PHONEME /A/. **
  =NO
  THEN WHAT DO YOU MEAN BY THE ABOVE RETRIEVAL CONDITION?
(4) PLEASE SPEAK IN JAPANESE !
  =KJQKUBDIN /A/ NO YUSEIHARETUON O FUKUNU ONSETU
  TRANSLATION INTO IQL
  *****
  SY=VP+A
(5) ***CORRECTED SUBCOMMAND SEQUENCE IS AS FOLLOWS:***
  COND WD ; SS = MALE AND SY = VP + A
  SHALL I EXECUTE THE CORRECTED SUBCOMMAND SEQUENCE?
  (YE OR NO)
  =YE
  9 RECORDS FOUND.

```

例 4 項目名に専門用語を使用したときの PSS の使用例
Example 4 Dialog in the case of misuse of a technical term.

の状態を「多重検索を並列に使用」にする。次に IR=14 のルールが検索される。このルールが実行され、LHS 関数によりコマンド系列に “[”, “]” が2組以上並列にあるかを調べ、あれば RHS 関数を実行する。RHS 関数では、誤りの個所を“へ”で示し、誤りの原因をわかっているかを聞く。わかっているならば利用者に正しいコマンドを入力させ、わかっているなければエラーメッセージを表示し、訂正されたサブコマンドの系列を示す。

⑤ 制御が再び MONITOR に移される。ここでは MONITOR が、PSS で訂正された IQL コマンドを利用者に代わって実際に SPEECH-DB に対して検索を行っている。

⑥ 制御が IQL に戻り、サブコマンドの入力待ちの状態である。

【例4】

① 利用者が意図した検索条件は「後続母音が /A/ の有声破裂音 (VP) を含み、男性が発声した単語」である。専門用語を項目名に使用することができないために、エラー番号“0317”が検出され、PSS が起動される。

② {IR=2, 15} の誤りの所を示し、エラーメッセージを表示した後、誤りの検索条件について診断を行っている。

③ {IR=15} 「音韻 /A/ が欲しい」の意図で記述したのかを聞いている。“NO”が入力されたのでその検索条件をどのような意図で記述したのかを聞いている。

④ 制御が JCS に渡された。日本語で入力された検索条件「後続母音 /A/ の有声破裂音を含む音節」の IQL への翻訳を行っている。

⑤ {IR=15} 制御が再び PSS に戻り、訂正されたコマンドを示した後、MONITOR が検索を行っている。

4.3.2 IQL の意味上の誤り

意味上の誤りでは、利用者の誤解がゴール状態となり、構文上の誤りのようにゴールを設定することができない。そのため、利用者が入力した IQL コマンドをデータとしながら、利用者との対話により前向き推論を行って誤りの原因を解明していく。

IQL における意味上の誤りには大きく分けて次の三つのものがある。

- (1) あらかじめ用意されている外部スキーマ (INQ セクションの論理構造) に対する誤解。

(2) 検索条件同士の修飾関係の制限における“(” “)”の使用誤り。

(3) 検索条件を連結する論理演算記号 AND と OR の使用における“(”, “)”の使用誤り。

(1)の誤りは三つのなかで最も多く起こるものであるが、その典型的な例を示す。

【誤1】 COND WD; PH=A AND RKI=A

利用者の意図は「定常状態 (RKI=A) を含む音韻 /A/ を含む単語」であるが、正しくは

【正1】 COND WD; (PH=A AND RKI=A)

と入力しなければ意図に合わない単語もいっしょに検索されてしまう。すなわち、音韻 /A/ に定常状態を含むものがなくてもその単語に含まれる他の音韻に定常状態を含むものがあれば検索されてしまう。このときの外部スキーマは単語ファイルをベースとし、音韻ファイルは単語ファイルの下位ファイルとして展開されている。したがって、誤1の条件では、音韻 /A/ を含む単語ファイルのレコードと定常な音韻 (RKI=A) を含む単語ファイルのレコードとの論理積がとられるが、これに対して正1の条件では、音韻が /A/ の音韻ファイルのレコードと定常な音韻のレコードとの論理積がとられた後、そのレコードを含む単語ファイルのレコードが検索されるからである。

(1)の誤りのもう一つの例を示す。

【誤2】 COND WD; (SY=SA AND RKI=A)

利用者の意図は「定常状態の音韻 /A/ を含む音節 /SA/ を含む単語」であるが、正しくは、

【正2】 COND WD; [SY=SA; PH=A AND RKI=A]

と2重検索を用いた表現で入力しなければ意図に合わない単語もいっしょに検索されてしまう。すなわち、音節 /SA/ の音韻 /A/ に定常状態を含むものがなくても、その単語の他の音韻に定常状態を含むものがあれば検索される。これは、このときの論理構造が、単語、音節、音韻ファイルの順で階層構造に実現されていれば誤2の条件で利用者が意図した検索は可能であるが、実際に IQL に用意された外部スキーマでは、単語ファイルをベースとしてその下位に音節、音韻ファイルが同レベルに並列に展開されているため、1回の検索では1レベルの階層構造を含む検索しかできないからである。IQL では、2レベル以上の階層構造を含む検索を可能にするため、正2の条件で用いた多重検索が用意されている。

次に(2)の誤りの例を示す。

```

TYPE IN SUBCOMMAND
(1) COND WD: (SY=KU AND RK1=A) AND (SY=RI AND RK1=A)
      2 RECORDS FOUND.
TYPE IN SUBCOMMAND
(2) LIST IW WD SY PH RK1
      ...
TYPE IN SUBCOMMAND
(3) ?
WELCOME TO CONSULTANT SYSTEM FOR SPEECH-DB
WHAT CAN I DO FOR YOU?
PLEASE SPEAK UP
(4) *SUBCOMMAND 0 BINDAN SITEKUDASAI
(5) DO YOU REQUIRE DIAGNOSIS OF SUBCOMMAND WHICH YOU TYPED IN?
    =YES
*****
*WELCOME TO PROBLEM SOLVER SUBSYSTEM*
*****
(6) SHALL I DIAGNOSE THE FOLLOWING SUBCOMMAND? (YE OR NO)
COND WD: ( SY = KU AND RK1 = A ) AND ( SY = RI AND RK1 = A )
    =YES
(7) I WILL DIAGNOSE THE PART:
SY = KU AND RK1 = A
(8) DO YOU THINK THE FOLLOWING RETRIEVAL CONDITION MATCHES
WITH YOUR INTENTION? (YE OR NO)
PH=U OF SY=KU IS RK1=A
    =YES
(9) I WILL CURE THIS PART AS FOLLOWS:
COND WD: ( SY = KU ; PH = U AND RK1 = A ) AND ( SY = RI AND
      RK1 = A )
(10) SHALL I CONTINUE TO DIAGNOSE? ( YE OR NO )
    =YES
    
```

例 5 外部スキーマの論理構造に対する誤解がある場合の PSS の使用例
 Example 5 Dialog in the case of misunderstanding of the external schema.

```

IR = 41 IF PSSを使用する THEN ・コマンド系列に ( ) がある
      L-func ( " " ) ・コマンド系列に ( ) がない
      R-func ( PSS が診断するコマンド系列を決定する)
IR = 42 IF コマンド系列に ( ) がない THEN 同一ファイルの項目がある
      L-func ( " " )
IR = 43 IF コマンド系列に ( ) がある THEN 診断していない ( ) がある
      L-func ( " " ) R-func ( 各 ( ) の位置を示すリストを作成する)
IR = 44 IF 診断していない ( ) がある THEN R-func ( その部分の検索条件の各項目が属するファイルによりWMの状態を決定する)
      L-func ( " " )
      あれば1つ取り出す)
IR = 45 IF 音節、音韻ファイルの項目 THEN ・満足している
      L-func ( SY, RK1がある) ・診断していない ( ) がある
      R-func ( 多重検索により訂正する。多重検索を並列に使用をWMの状態に記述する)
    
```

図 8 意味上の誤りを処理するルール例
 Fig. 8 Some examples of production rules for semantic errors.

[誤 3] COND WD; SY=BA AND SAP>7.0 AND SY=KU
 この条件では、「平均ピッチ (SAP) が7.0 msec 以上の音節を含み、音節 /BA/ と /KU/ を含む単語」が検索される。平均ピッチ 7.0 msec 以上が音節 /BA/ か /KU/ のどちらかにかかるとなれば、その部分を“(, “)”によって修飾しなければならない。
 (3)の誤りは、論理演算記号の優先順位 ((,)>

NOT>AND>OR) の誤解によるもので、最も単純な誤りであるが、誤りの原因が IQL に対する理解と無関係であるため、IQL コマンドのみからその誤りを同定することは不可能である。したがって、PSS は(1), (2)の誤りをまず調べて、そうでない場合に(3)の誤りについて考え、利用者との対話により解決する。

例 5 に意味上の誤りに対する PSS の使用例を示す。また、おもなルールの日本語表現を図 8 に示す。

[例 5]

- ① COND サブコマンドを入力し検索を行っている。条件は「/KU/ という音節でそのランクが定常のものを含み、かつ音節 /RI/ でそのランクが定常のものを含む単語」である。
- ② LIST サブコマンドにより検索されたレコードのデータ項目値を表示させる。
- ③ 検索された結果が期待していたものと異なるため、コマンド“?”を入力し制御を MONITOR に移している。

- ④ 「サブコマンドを診断してください」と入力されると、MONITOR は辞書を用いて“診断”をキーワード“PSS”に変換し WM に記述する。
- ⑤ {IR=41} PSS が起動された。以下の部分は、利用者が入力したコマンドをデータとして前向き推論により誤りの原因を解明していく過程を示している。
- ⑥ {IR=41} “?”を入力する直前に利用者が入力した COND サブコマンドについて診断を行うかどうかを聞いている。ここで“NO”を入力すると診断して欲しい IQL コマンドを入力することができる。
- ⑦ {IR=43, 44} まず最初の“(, “)”で囲まれた部分について診断を行う。
- ⑧ {IR=45} その部分が、「音節 /KU/ の音韻 /U/ が定常である」の意図で記述したのかを聞いている。このように曖昧な部分は利用者との対話によって解決され、より正しい IQL コマンドへと導かれる。ここで“NO”と入力すると JCS に制御が渡され、利用者は日本語で検索要求を入力できる。
- ⑨ {IR=45} 最初の“(, “)”で囲まれた部分はどのように訂正された。
- ⑩ {IR=46} 診断を続けるかどうかを聞いている。

この例においては、最終的に延べ9個のルールが発火しており、構文上の誤りと比較すると誤りを訂正する過程は複雑なものとなっていることがわかる。

5. む す び

本論文では、汎用 DBMS を用いたプロダクションシステムの実現と、本 PS を用いた、SPEECH-DB との知的マン・マシンインタフェースの実現について述べた。現在、会話用および問題解決用として52個のルールからなるプロダクションシステムが作成されている。JCS では、最初に検索文の例をいくつか提示しているが、音声に関する知識をもつ協力的な利用者がこれらの例を参考にして作成した検索要求文であれば十分実用に耐えることが確かめられている。PSS に関しては、事前に想定された誤りに対しては完全に訂正可能であるが、実際の使用に際しては思いがけない誤りが生じ、対処できないことがある。新しい誤りの訂正用のルールを追加して能力の向上に努めているが、このような段階的なシステムの構築には本システムの基本構成は適したものとなっている。

ソフトウェア工学においてはソフトウェアの生産性と信頼性の向上が重要な課題であることは論を待たないが、ソフトウェアというものは元来計算機の利用技術、すなわち計算機を使いやすくするためのものであることも忘れてはならない。情報処理システム自体の有用性もさることながら、その使いやすさに欠陥があれば、真の意味での計算機の一般社会への浸透は望めないであろう。計算機の非専門家あるいは初心者にとって、大部で難解なマニュアルを読まなければならないということが大きな負担となっている事実は否めない。ここに「マニュアルレスシステム」の有用性が見いだされる。「マニュアルレスシステム」はたんなるHELP コマンドの拡張ではない、構造化されたマニ

ュアルの計算機による管理はもちろんのこと、使用中に発生する種々の誤りからの回復、諸々の疑問に対する解答等の機能をもたなければならない。もちろん、自然言語インタフェースがあることが望ましい。「マニュアルレスシステム」は別の見方をすれば、そのシステムの利用全般に係わる知識をもつある種のエキスパートシステムとも考えることができ、計算機にインテリジェンスをもたせるという情報科学の課題の一つとしてその重要性は増すものと思われる。

参 考 文 献

- 1) Hayes, P., Ball, E. and Reddy, R.: Breaking the Man-Machine Communication Barrier, *IEEE Comput.*, Vol. 14, No. 3, pp. 19-30 (1981).
- 2) Genesereth, M.R.: The Role of Plans in Intelligent Teaching Systems, Stanford University, HPP-80-4 (1980).
- 3) 溝口, 前田, 浜口, 芥子, 柳田, 角所: 知的アクセス機能をもつ音声データベース「SPEECH-DB」, 情報処理学会論文誌, Vol. 24, No. 3, pp. 271-280 (1983).
- 4) Shortliffe, E.H.: MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection, Stanford University, CS-74-465 (1974).
- 5) Stefik, M. et al.: The Organization of Expert Systems, A Tutorial, *Artif. Intell.*, Vol. 18, No. 2, pp. 135-173 (1982).
- 6) Davis, R. and King, J.: An Overview of Production Systems, *Mach. Intell.*, Vol. 8, pp. 300-332 (1977).
- 7) 溝口, 大上, 富田, 西山, 角所: 階層的プロダクションシステム—HIPS, 情報処理学会論文誌, Vol. 23, No. 2, pp. 177-186 (1982).
- 8) 日本電気(株): INQ 概説書, INQ 文法説明書, INQ 運用説明書(1981).

(昭和58年6月30日受付)

(昭和58年10月11日採録)