データベース用自然言語インタフェース「ｄＢｍａｔｅ」

5Q-1

伊藤　篤　　高橋　清一

㈱ＣＳＫ総合研究所

１．はじめに

データベースの使用は初心者にとっては困難であり、そのため、自然言語インタフェースを持ったＤＢＭＳがいくつか考案されている。しかし、自然言語から派生する諸々の問題により実用レベルのものはないというのが現状であろう。

そこで、我々は、パソコンベースで普及しているリレーショナル・ＤＢＭＳを対象とし、実用に耐えるシステムを目指して、自然言語インタフェース「ｄＢｍａｔｅ」を開発した。このシステムは、自然言語からＤＢＭＳのコマンドを生成・実行し、自然言語による情報検索，情報操作を可能にしている。

２．システム概要

ｄＢｍａｔｅでは、自然言語によって生じるデメリットを最小限にして、自然言語のメリットを生かし、実用レベルにもっていくために、以下の点を工夫した。

・自然言語システムにありがちな問題点である「まず最初に目的に合わせて辞書を作成しなければならない」という面倒をなくす。これは、後述のＤＢ依存辞書の自動作成機能によって達成した。
・Ｃ言語によるパーザを使い解析を高速にする。
・分かち書きが要らない。
・漢字かな混じり文でユーザの直感に近い文を入力できる。
・未定義語の処理によって、全ての使用する単語を辞書に登録する必要はない。
・「営業の男は？」の例のようにかなり省略した文も解析・実行する。
・解析が失敗すると間違ったと思われる場所に↑を表示して再入力を促す。
・ヒストリー機能により過去に入力した文はすぐに呼び出してエディト・実行ができる。
・例文メニューによりよく使う文は簡単に入力できる。
・データベースを自動的に選択・結合する。

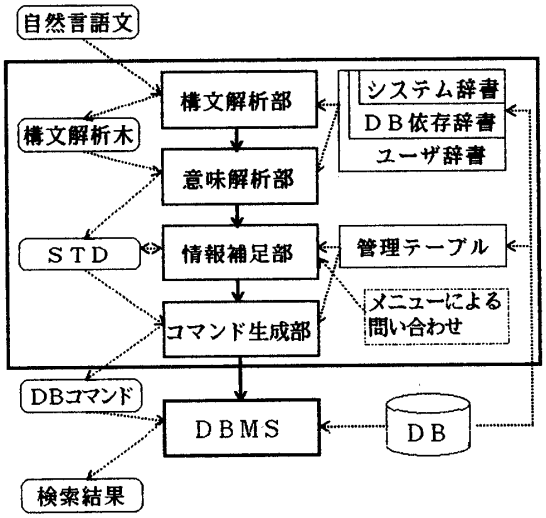


図１　システム構成

「dBmate」
　Natural Language Interface for DBMS
Atsushi ITOH, Seiichi TAKAHASHI
CSK Research Institute

１）システム構成

システム構成を図１に示す。ｄＢｍａｔｅは３層の構造を持つ辞書（文法）とデータベースの状態を保持する管理テーブルを持つ。

ユーザがかな漢字変換ＦＥＰによって入力した自然言語文から、辞書とパーザの構文解析によって解析木を作成する。さらに解析木は意味解析によってＳＴＤ（Semantic Tree for DBMS）と呼ぶ中間表現に変換される。ユーザの入力した自然言語文は省略が多かったり、データベースの指定がなかったりするので、このままではＳＴＤは完全ではない。そこで、足りない情報を補ってＳＴＤを完全にする。この情報とは主にフィールドとファイル（データベース名）である。

ＳＴＤが完全になるとコマンド生成部によってＤＢＭＳプログラムに変換しＤＢＭＳに引き渡して自動的に実行される。結果はＤＢＭＳによって画面・プリンタ等に出力される。

３．データ構造

１）辞書の構造

辞書はシステム辞書，ＤＢ依存辞書，ユーザ辞書からなる。

システム辞書はシステムに組み込まれている辞書で、文法に関する情報と必要最低限の単語（助詞等）を含む。これは固定であり一切変更されない。

ＤＢ依存辞書はシステム立ち上げ時に指定したディレクトリに存在する（複数の）データベースから自動的に作成する。ファイル名，フィールド名，よく使われそうなデータ等を登録する。ＤＢ依存辞書の自動作成と同時に管理テーブルも作成する。

システム辞書とＤＢ依存辞書によって一通りの操作が可能であるが、さらにユーザ独特の表現を可能にするためにユーザ辞書がある。ユーザ辞書はユーザが必要に応じて付け足すもので「ＸＸＸを辞書に登録する」といった文を入力することやメニューによって登録する。

２）ＳＴＤ

パーザによって作られた解析木から意味解析によってＳＴＤを作成する。ＳＴＤの構造を図２に示す。ＳＴＤのルートノードは[FRAME]と呼ばれ６つの要素からなる。最初の<機能名>にはCREATE/USE/APPEND/EDIT/DELETE/DISPLAY/INDEX/REPLACE/AVERAGE/COUNT/SUM/RECALL/QUIT等が入る。これはＤＢＭＳのコマンドに１対１に対応するのではなく、データベースを作りたい（CREATE）とか情報を得たい（DISPLAY）といった機能に対応する。実際にどのコマンドを生成するかはコマンド生成部の仕事である。

<機能名>以外は子ノードとして[SUBFRAME]を持つ。[SUBFRAME]は<operation><left><right>からなるセルで

<left><right>にはさらに[SUBFRAME]が入る。<operation>に 0 が入る場合は特別にリーフセルを意味し、このときは<left>にそのセルのタイプが入り<right>にはデータ等が入る。

例えば「営業の男は？」という文が入力された場合のＳＴＤを図３(ａ)に示す。

３）管理テーブル

管理テーブルにはデータベースの状態が記述される。データベースの状態とは、データベース名，そのデータベースの作業領域番号とそのデータベースの持つフィールドの状態である。フィールドの状態はフィールド名，フィールドの型（文字型／数値型／論理型／日付型），インデックスの状態からなる。インデックスの状態はさらにインデックスファイルを管理する番号，状態（有り／無し／不整合），マスタキーかどうか，データの頻度（異なったデータが多い／普通／同じデータが多い）からなる。システムはこれらの情報からＳＴＤの足りない部分を補足する。

４．意味解析後の処理概要

１）フィールド情報の補足

ＳＴＤの中でフィールド指定のないデータを見つけると、フィールド情報を補う。管理テーブルからフィールド名を抽出してユーザにメニュー形式で問い合わせる。ユーザが選択するとフィールドが決まる。

２）ファイル情報の補足

フィールド情報が充足すると、次はどのデータベースから検索するかを決める。フィールドの指定によっては２つのデータベースを結合して情報を得ることも有り得る。場合によってはファイルの選択が一意に決まらない場合がある。このようなときはユーザにメニュー形式で問い合わせて確定する。

フィールド情報，ファイル情報を補足したＳＴＤの例を図３(ｂ)に示す。

３）コマンド生成

ＳＴＤが完全になると、<機能名>によって各コマンド生成を行なう。コマンドが生成されるとＤＢＭＳに引き渡され実行される。ここでコマンドといっても一つのコマンドではなくコマンド列のことである。プログラムといっても良い。「営業の男は？」の場合に生成されるプログラムは図３(ｃ)を参照されたい。実行が終了すると再び自然言語文の入力待ちになる。

５．おわりに

ｄＢｍａｔｅはユーザが使いやすいシステムとなるために、ＤＢＭＳに合わせてｄＢｍａｔｅ側でいろいろと工夫した。しかし、真に使いやすいシステムを考えるならば、ＤＢＭＳの側から歩みよるべきではないだろうか? 例えば、ファイル（データベース）の自動選択・連結等は自然言語から派生した要求とはいえ、ＤＢＭＳ側でやるべきことであると思う。

我々は単なるプロトタイプではなく、実用的なシステムを目指してｄＢｍａｔｅを開発した。その目的はほぼ達したと考える。

謝辞

本研究開発の機会を与えて下さり、討論に加わっていただいた当社関係各位に深く感謝の意を表します。

参考文献

[１]伊藤，高橋：データベース用自然言語インタフェースにおけるデータベースの自動連結，第38回全国大会講演論文集

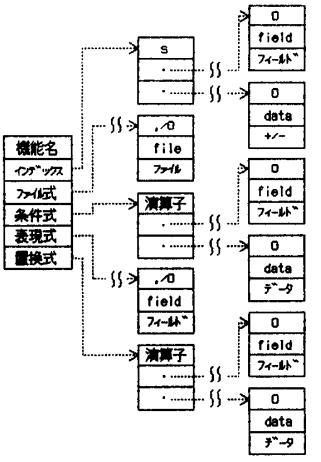
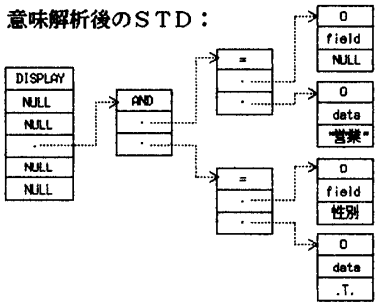


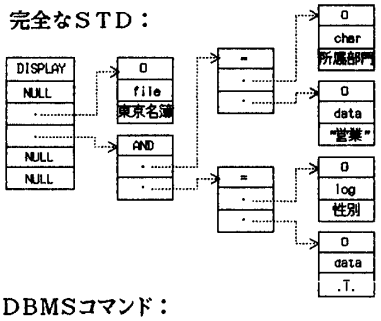
図２　ＳＴＤの構造

入力文：「営業の男は？」

(ａ) 意味解析後のＳＴＤ：



(ｂ) 完全なＳＴＤ：



(ｃ) DBMSコマンド：

```
SET TALK OFF
SET DATE ANSI
SET EXACT ON
SET DELETED ON
?"---------- 東京名簿 ----------"
SELECT 1
USE A:¥MATE¥東京名簿
LOCATE FOR (所属部門 = "営業") .AND. 性別)
IF FOUND()
DISPLAY FOR (所属部門 = "営業") .AND. 性別) OFF
ELSE
        ?"《条件に合うデータはありません。》"
ENDIF
SET INDEX TO A:¥MATE¥東京名07
SET EXACT OFF
SET TALK ON
```

図３　ＳＴＤの例