EXPERT SYSTEM AS AN INTELLIGENT ASSISTANT FOR COMPUTER USERS

Riichiro MIZOGUCHI*, Yukuo ISOMOTO** and Osamu KAKUSHO*

* Institute of Scientific and Industria Research, Osaka University, Japan
** Computation Center, Osaka University, Japan

abstract

This paper is concerned with a new implementation of a production system by using a relational DBMS and FORTRAN and its application to construction of an intelligent man-machine interface of a speech database called SPEECH-DB. Our Production system can be viewed as a nondeterministic FORTRAN programming system based on content-directded invocation of subroutines. The interface is composed of the following three subsystems: Japanese Command processing Subsystem (JCS), which translates the queries stated in Japanese into commands, Problem Solving Subsystem (PSS), which diagnoses errors and answers user's questions, and MONITOR, which manages the flow of control among these subsystems and SPEECH-DB. Design philosopby and basic mechanism are presented together with some examples of the system performance.

I. INTRODUCTION

A large application *software* is apt to be difficult for nonprofessional people to use and many complicated commands sometimes cause misuse of them. Thus, a software system with high functions needs an intelligent man-machine interface[1]L2] which assists users in using it. We have been designing such a man-machine interface for a database called SPEECH-DB[3L SPEECH-DB is a speech database constructed for assisting the research of speech recognition. It stores a lot of sampled speech data together with their attributes so as to be able to retrieve them under various conditions. It has a command language called IQL. Although IQL makes it easy to retrieve data, users still need to have some knowledge about SPEECH-DB and IQL in advance. Our interface system of SPEECH-DB is designed to have the following three functions:

1) To accept queries stated in Japanese.
2) To make it easy for users to learn IQL.
3) To diagnose errors occurred while using IQL.

In order to achieve these ends, the interface has several knowledge about SPEECH-DB, IQL, Japanese sentences and so on. Our ultimate goal is to construct a so-called "manual-less" system. Such a system should have the knowledge of what it can do, what functions it has, how the user can use them, what kind of troubles the user will meet, and so on. The system can be viewed as a kind of expert systems.

One of the major characteristics of our system is that most subsystems are implemented by using a relational database called INQ. INQ is employed for the purpose of storing and retrieving procedures used for managing dialog between the system and the user and for diagnosing errors. A production system[4][5] was first implemented by using INQ and FORTRAN. Then the above interface was implemented by using the production system.

In our computation center, a number of application programs have been developed. The objective of our project is to construct a consultation system about these programs *as* well *as* the use of the computation center itself. For this purpose, we need a tool for construction such a system. Production system is well-known to be useful for consultation system construction. The problem is how to implement a PS. The tool should be able to utilize DBMS, graphics and other functions of the operating system of our host computer in order to cope with the variety of the target programs. In our computation center, FORTRAN is the only language satisfying these requirements. This is one of the main reasons why our PS has been implemented by FORTRAN.

II. OVERVIEW OF THE SYSTEM

Block diagram of our system is shown in Fig. 1. It is composed of seven database files, six of which contain speech data and one of which contains several kinds of knowledge of consultation about the usage *of* SPEECH-DB.

SPEECH-DB consistes of six files such as phoneme file, CV syllable file, VCV syllable file, word file, environment file and raw data file. The first five files are called INQ files and contain a lot of information useful for retrieving speech data. Users can retrieve speech data by using IQL interactively. IQL selects an external scheme suitable for the user's query, so that the user can get speech data he wants without knowing the logical structure of SPEECH-DB.

B. Man-machine interface

Although IQL is designed for users to retrieve data easily, they must know some syntactic and semantic restrictions of IQL. So, more sophisticated man-machine interface is desirable especially for nonprofessional users. Our interface consists of three modules such as Problem Solving Subsystem (PSS), Japanese Command processing Subsystem (JCS) and MONITOR. PSS diagnoses the errors occurred during the use of IQL and answers questions asked by the user. JCS translates queries stated in Japanese into IQL.

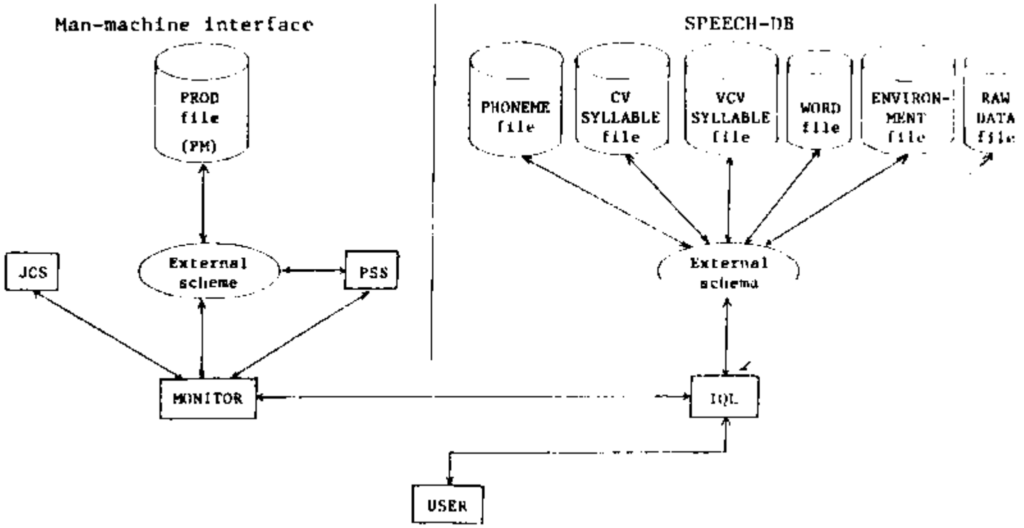

Fig. 1 Block diagram of the system.

MONITOR manages the flow of control among PSS, JCS and IQL.

III. IMPLEMENTATION OF PS USING A DBMS AND FORTRAN

Production system, which realizes a declarative representation of procedural knowledge, is well-known to be useful in knowledge engineering[6][7]. A PS consists of the following three components:

1)Working Memory(WM) which describes the problem state.
2)Production Memory(PM) which stores production rules.
3)Production System Interpreter(PSI) which retrieves appropriate rules from PM and execute one of them.

Although PS's are usually implemented by using LISP, our PS is implemented by INQ and FORTRAN which is a parent language of INQ. In our implementation PM is considered as a database and WM is considered as a query to it. By using DBMS, it is possible for our PS to cope with a large number of rules which will be required in the real world problems. PSI has two interpretation mode such as LHS-driven (forward-chaining) one and RHS-driven (backward-chaining) one. It also has an automatic backtracking mechanism. Data structure of a production rule is shown in Fig. 2.

```
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.
   04 ATTRIBUTE-NAME3    PIC X(8) DSP.
   04 ATTRIBUTE-VALUE3   PIC X(8) DSP.
```

Fig. 2 Data structure of the production rule.

IV. PSS

There are two types of errors which the user commits when using IQL. One is syntactic, in which case error messages are given. The other is semantic, in which case retrieval is done without any message but the result does not satisfy the intention of the user. PSS resolves these errors by inferring the user's intention from the sequence of IQL commands.

A. Syntactic error case

In the case of syntactic errors, PSS works in the backward-chaining mode. Since the error numbers are given in this case, backward-chaining using these error numbers as goals is suitable for the invocation mode of production rules. Control is passed to MONITOR when IQL detects an error. Then MONITOR invokes PSS. An example of a dialog is shown in the following:

```
    SYSTEM ?SPEECH-DB
    WELCOME TO SPEECH-DB
    COMMAND ?
    •SEARCH
    TYPE IN SUBCOMMAND
(1) •COND WD;EAP > EAP[ID=11] AND EAR > EAR[ID=11]
(2) ***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 MULTI RETRIEVAL***
    ********************************************

    *** CORRECTED SUBCOMMAND SEQUENCE IS AS FOLLOWS;***

    COND WD ; EAP  >  EAP [ ID = 11 ]
    SAVE
    COND WD ; EAR  >  EAR [ ID = 11 ]
    SAVE
    AND 1 2
(3) 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.

    TYPE IN SUBCOMMAND
    •
```

(1) This query means "I want some WorDs whose Averaged Pitch is greater than that of the speech of ID=11 and whose Averaged Root mean square of energy is greater than that of the speech ID=11." Although IQL permits nested retrieval, which is indicated by a paired brackets, it may not be used in parallel. Therefore, an *error* occurs, the error number is detected by IQL and then MONITOR invokes PSS.

(2) A rule which has the error number in its action part is fired.

(3) MONITOR carries out the corrected commands.

B. Semantic error case

When semantic errors occur, there is no goal to set up, since such errors are found only by users after they checked the property of the retrieved data. Therefore, PSS performs inference in the forward-chaining mode. The following is an example of a dialog in the case of semantic errors:

```
   TYPE IN SUBCOMMAND
(1) *COND WD;(SY*KU AND RK1*A) AND (SY*RI AND RK1*A)
               2 RECORDS FOUND.
   TYPE IN SUBCOMMAND
   *LIST IW WD SY PH RK1

   - - - - - - - - - - - - - - - - - - - - - - - -

   TYPE IN SUBCOMMAND
(2) *?
   WELCOME TO CONSULTANT SYSTEM FOR SPEECH-DB
   WHAT CAN I DO FOR YOU?
   PLEASE SPEAK UP
(3) *SUBCOMMAND O SINDAN SITEKUDASAI
   DO YOU REQUIRE DIAGNOSIS OF SUBCOMMAND WHICH YOU TYPED IN?
   *YES
   ***********************************
   *WELCOME TO PROBLEM SOLVER SUBSYSTEM*
   ***********************************

   SHALL I DIAGNOSE THE FOLLOWING SUBCOMMAND? (YE OR NO)
   COND WD ; ( SY * KU AND RK1 = A ) AND ( SY = RI AND RK1 * A )
   *YES

   I WILL DIAGNOSE THE PART:
   SY = KU AND RK1 * A

(4) DO YOU THINK THE FOLLOWING RETRIEVAL CONDITION MATCHES
   WITH YOUR INTENTION? (YE OR NO)
   PH=U OF SY*KU IS RK1=A
   *YES

   I WILL CURE THIS PART AS FOLLOWS:
   COND WD ; [ SY = KU ; PH = U AND RK1 = A ] AND ( SY * RI AND
                                                         RK1 =A )
   SHALL I CONTINUE TO DIAGNOSE? ( YE OR NO)
```

(1) This query means "I want some WorDs which contain SYllable /ku/ with RanK=A and SYllable /ri/ with RanK=A."

(2) The commands "?" is input by the user, since the retrieved data is slightly different from those he expected to have.

(3) Users who do not know the module name which answers his questions can say what they want to do in Japanese.

(4) The condition RK1=A, which means the corresponding part of speech is stationary, is the property not of syllable but of phoneme in SPEECH-DB. So, PSS asks the user his intention.

During the dialog in both cases, PSS calls JCS when necessary, so that users can input their requirements in Japanese. This makes the interaction very friendly.

V. JCS

JCS translates queries stated in Japanese into IQL and assists users in learning IQL by showing the translated IQL commands. JCS is designed just for beginners. The authors believe that artificial command language is more efficient than natural language once one gets familiar with it.

Semantics-based processing of queries is mainly performed instead of syntactic one. This style of processing is based on the fact that queries to SPEECH-DB is much restricted to the semantics of SPEECH-DB. IQL is designed to be used as a target language of natural language queries, so that it was very easy to implement JCS. For example, IQL can select an external scheme appropriate for the query. Furthermore, it does not care about the order of condition elements. These characterisetics of IQL makes the translation easy drastically. In the current implementation, JCS can accept almost queries realizable in IQL and can reject queries which are beyond the ability of IQL with some messages indicating the inappropriate part of them. *JCS* has two kinds of dictionaries. One contains some keywords corresponding to the attribute names and their values used in SPEECH-DB. The other contains some words having syntactic information which are used for examining the relationship between conditions. *JCS* do not have to know any other words stored in the above two dictionaries, since a query composed of such words can not be realizable by IQL at all.

VI. CONCLUSIONS

Although the implementation of our interface is based on a new production system, the whole model is rather ad hoc. In order to realize the "manual-less" system mentioned earlier, several problems should be resolved.

1) To establish a psychological model of a user.
2) To establish a theory of proceeding with conversation based on the user's model.
3) To incorporate the results of intelligent CAI into our system.

These are the subjects to be attacked at the next step in our study.

[REFERENCES]

[1]P. Hayes et al.:"Breaking the man-machine communication barrier," IEEE Computer, Vol. 14 p. 19 (1981).

[2]M.R. Genesereth:"The role of plans in automatic consultation," Proc. of the 6th IJCAI, p. 311 (1979).

[3]R. Mizoguchi et al.:"Speech database with *an* intelligent access mechanism——SPEECH-DB," Trans, of Information Processing of Japan, Vol. 24 (1983).

[4]R. Davis et al.:"An overview of production systems," Machine Intelligence, Vol. 8, p.300 (1977).

[5]R. Mizoguchi et al.:"Hierarchical production system," Proc. of the 6th IJCAI, p.586 (1979).

[6]E. Shortliffe: "Computer-based medical consultations: MYCIN," American Elsevier, New York (1976).

[7] W. Clancey: "Tutoring rules for guiding a case method dialogue," International Journal Man-Machine Studies, Vol. 11, p. 25 (1979).