Comparative analysis of proposed novel Multimodal user interface and other user interfaces to relational databases – A review

Mrs. D. MABUNI
Research Scholar in Computer Science
Rayalaseema University
Kurnool
Andhra Pradesh, India
mabuni.d@gmail.com

Dr. M. Hanumanthappa
Professor & Chairman
Dept. of Computer Science and Applications
Bangalore University
Bangalore, Karnataka, India
hanu6572@hotmail.com

Dr. S. Aquter Babu
Associate Professor
Dept. of Computer Science
Dravidian University
Kuppam
Andhra Pradesh, India
aquter@gmail.com

***Abstract*—** A user interface part of a software system plays an important role in the success of software system. It controls the way how user and computer system interact each other. Nowadays, databases are being involved in many ways to provide information to computer users. As the number of computer users who require data from databases is increasing day-by-day, there is a need to develop user interface which enables the users to access data from databases easily and effectively. Many different types of user interfaces to databases were developed during the past five decades. One of the authors of this paper has proposed a novel architecture for the development of Multimodal user interface to relational databases in the paper published in the journal IJESI [1]. This present paper makes a comparative analysis of the proposed multimodal user interface and other user interfaces to relational databases.

Keywords: comparative analysis, user interface, relational database, multimodal, usability, graphics, menus, natural language, forms.

I. INTRODUCTION

A user interface of a software system enables the users to use all the functionality provided by the software system. It behaves like an intermediary between the user and computer. A user interface plays a vital role in the success of a software system. Nowadays, every organization maintains its data in the form of a database so that any user who requires data of an organization can be given data quickly. But, there is a need of a user friendly user interface to access data easily and effectively from a database.

A query language interface such as Structured Query Language (SQL) is provided by almost all Database Management Systems (DBMSs) to retrieve data from a database. The main disadvantages of this formal query language interface is that the users should learn the syntax of a query language and should know the database

structure details such as tables, attributes, relationships among tables etc. to formulate a query that to be submitted to the DBMS to retrieve data from a database. Typical users generally do not possess this knowledge about query language and database structure details.

Many alternative user interfaces to databases were developed to retrieve data from databases such as Form-based user interfaces, Menu-based user interfaces, Graphics-based user interfaces and Natural Language user interfaces. In each of these user interfaces, there are some advantages and disadvantages. None of these user interfaces satisfy all the usability principles such as reduction of users' memory load, use of users' language, prevention of errors etc. Users need a user interface to database that satisfy most of the usability principles. One of the authors of this paper has proposed a novel architecture for the development of Multimodal user interface to relational databases in the paper published in the journal IJESI [1]. This present paper makes a comparative analysis of the proposed multimodal user interface and other user interfaces to relational databases.

II. LITERATURE REVIEW

The following types of user interfaces to databases are found in the literature review.

a. Command-based user interfaces
b. Graphics-based user interfaces
c. Menu-based user interfaces
d. Form-based user interfaces
e. Natural language based user interfaces
f. Proposed Multimodal based user interfaces

III. PROPOSED NOVEL MULTIMODAL USER INTERFACE TO RELATIONAL DATABASES

One of the authors of this paper has proposed a novel multimodal user interface to relational databases in the paper published in the journal IJESI [1]. The same has been reproduced here for highlighting its features. It has been observed that a multimodal based approach i.e., combining different modalities such as graphics, menus, forms and natural language to merge the advantages of different types of user interfaces, can be used to develop a very effective user interface to database which will satisfy the following usability principles.

i) A simple and natural dialogue between the user and system.
ii) Informs in the users' language
iii) Users' memory load must be reduced
iv) Maintain consistency

v) Feedback must be provided to the user

vi) Exits should be marked clearly

vii) Good error messages must be provided

viii) Prevent most of the errors

The user interface that is developed using the multimodal approach is called Multimodal Interface to a Relational Database (MIRDB). The MIRDB uses the architecture mentioned in the following figure 1.

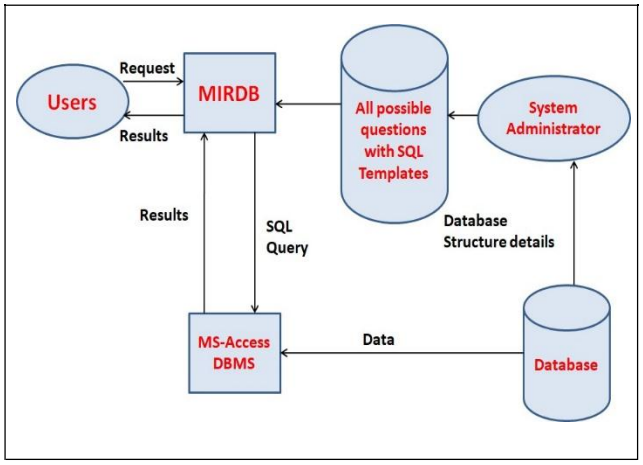


Figure 1 : Proposed Architecture for MIRDB

The above architecture proposes the system administrator to perform the following activities.

a. The System Administrator will analyze the structure of the Relational Database i.e., Tables and their relationships in the Database.

b. Identifies all the possible questions for each table and relationship between or among tables.

c. For each possible question, a unique question number is attached.

d. For each possible question, a SQL (Structured Query Language) template and other details are attached.

e. Question number, SQL template and other implementation details are stored in the Database itself as system configuration tables.

f. Four Standard Forms are developed to process questions selected by the user.

g. A Multimodal Interface to a Relational database (MIRDB) is developed utilizing all the modalities such as graphics, menus, forms and natural language

The following are the activities of users proposed by this architecture to retrieve data from a database.

a. Multimodal interface to a Relational Database (MIRDB) displays all the table names of all the tables in the database as graphic objects.

b. The MIRDB also displays each relationship between or among the tables as graphic object.

c. The user can submit his/ her request for data from the relational database by selecting the Table or Related Tables and clicking on the corresponding Graphic object. The system displays all the possible questions related to the table or tables as a Menu of questions. The questions will be in the Natural Language such as English.

d. The user selects the appropriate question by clicking on it.

e. The MIRDB analyzes the question selected by the user and retrieves the corresponding SQL Template and other details from the system configuration tables of the relational database.

f. The MIRDB then converts the SQL Template into executable SQL query by collecting details from the user.

g. The MIRDB sends the executable SQL query to MS Access (Microsoft Access) DBMS for execution.

h. MS Access DBMS executes the SQL query sent by MIRDB and retrieves the data from the relational database.

i. MS Access DBMS sends the retrieved data to MIRDB.

j. The MIRDB receives data from the MS Access DBMS

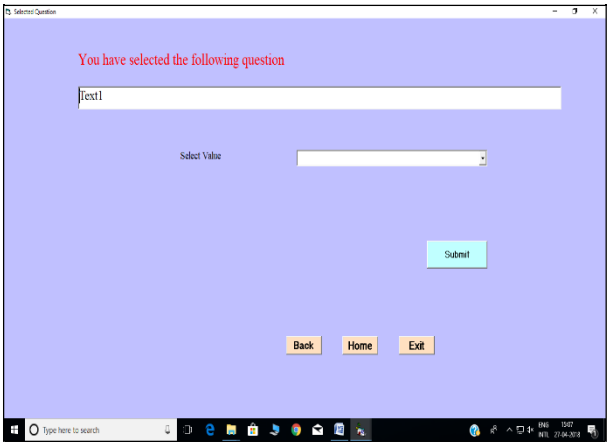
k. The MIRDB finally displays the resultant data as answer on the screen.

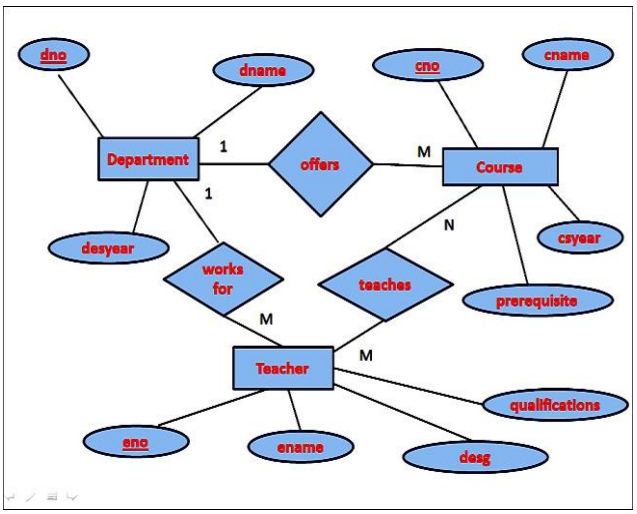
The proposed architecture classifies the questions into three types. They are as follows.

S.No.	Question Type	Format of question
1	0	General question. No values are present in the question.
2	1	Question will have one value. Users are enabled to modify this value.
3	2	Question will have two values. Users are enabled to modify these values.

The System Administrator develops four standard Forms to process all three types of questions and to display results.

For example, the standard form for processing questions of type 1 is as follows.



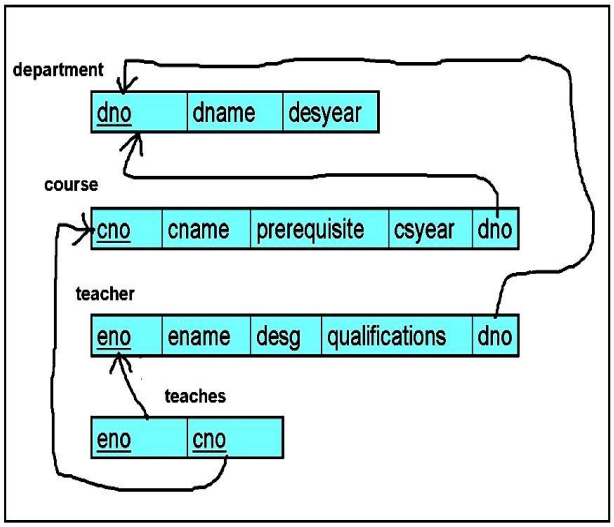


IV. THE IMPLEMENTATION OF THE ARCHITECTURE OF MIRDB AND EXPERIMENTAL RESULTS

The architecture for Multimodal interface to Relational Database (MIRDB) has been implemented using Visual Basic 6.0 as front-end and Microsoft Access 2003 as back-end.

A portion of a University database has been taken for implementation and experimenting the MIRDB architecture. The following is the Entity-Relationship (ER) diagram of a University database.

Tables:



The following sample data has been used for experimenting the MIRDB architecture for department table.

department		
dno	dname	desyear
1	History	1998
2	Computer Science	2000
3	Philosophy	2003
4	English	2010
5	Library Science	1999
6	Management	2012

The following is the Algorithm for processing the question asked by the user.

Algorithm Process_User_Question:

1. Accept the question selected by the user, say Q
2. Accept any values related to the question that are specified by the user
3. Retrieve the corresponding SQL Template from the system configuration tables, say T
4. Populate the SQL Template T with suitable values specified by the user to create executable SQL query, say R
5. Send SQL query R to the DBMS (e.g. MS-Access) for execution
6. Receive data from the DBMS
7. Display the results on the screen

The above algorithm has been implemented using Visual Basic 6.0 as front-end and MS-Access as back-end and tested.

Results:

The MIRDB allowed the users to retrieve all the data that is available in a relational database. The users could retrieve desired data **100% accurately** without any errors.

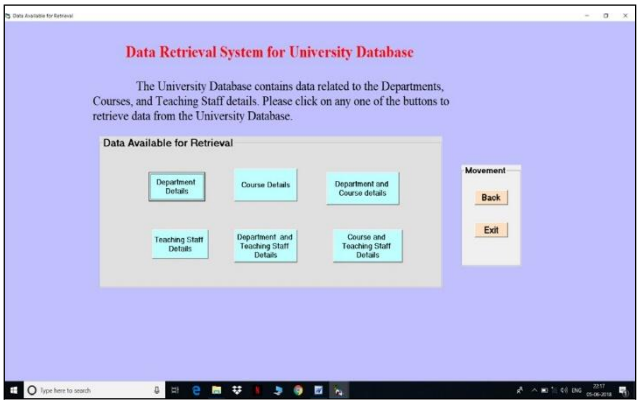
Testing screenshots:

Example question of type 1:

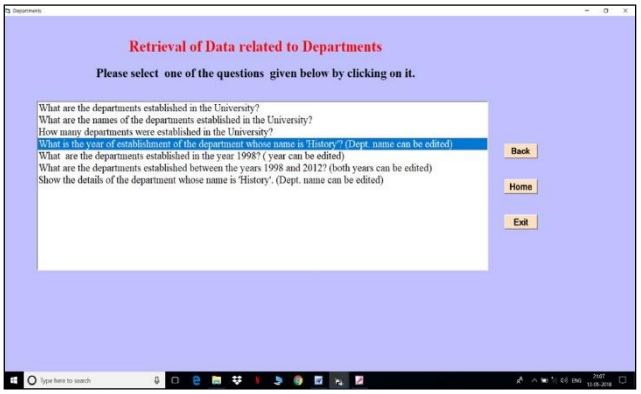
Assumption:
User wants to know the year of establishment of the department whose name is 'Computer Science'

The Steps will be as follows:

Step1: User clicks on 'Department Details' **Graphic object** as shown below.



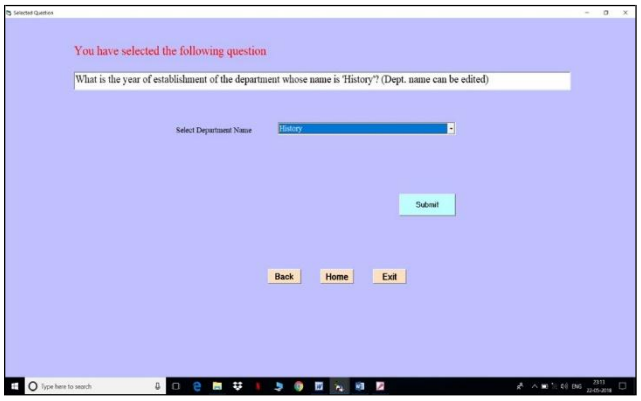
The system displays all the possible questions related to the department table in a **Natural Language** i.e., English in the form of a **Menu** as shown below.



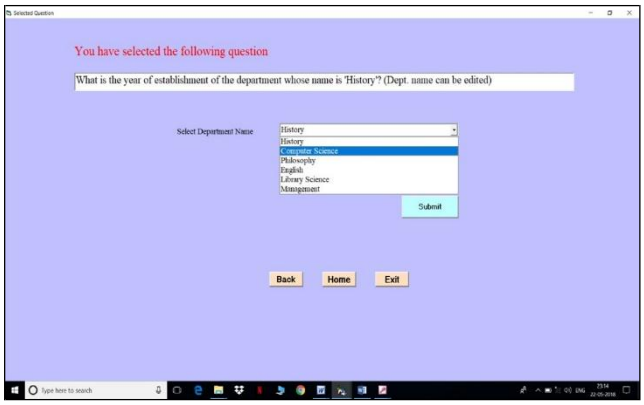
Step2: User selects the following question by clicking on it.

What is the year of establishment of the department whose name is 'History'? (Dept. name can be edited)

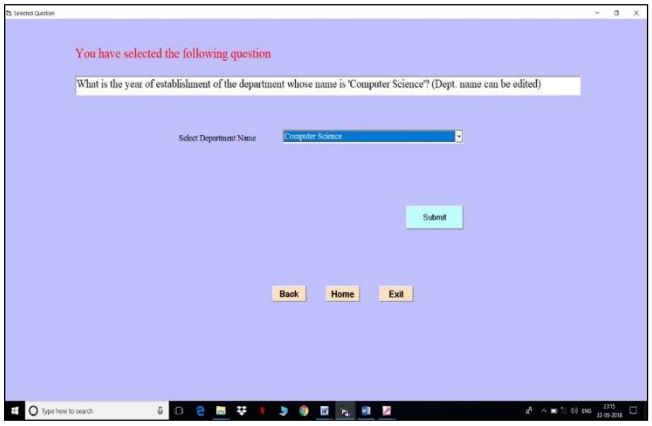
The system displays the selected question in a **Form** as shown below and enables the user to change the value of the question i.e., 'history'.



Step3: User can select a valid value from the list of values provided and change the value of the question. For example user can select the value 'Computer Science' as shown below.



The system changes the value of the selected question as shown below.

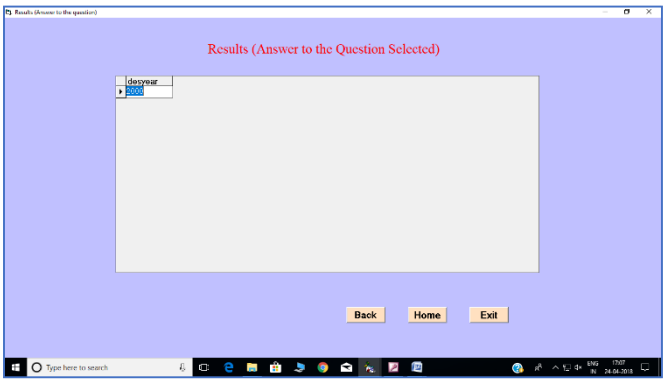


Step4: User clicks on submit button to get the data.

The system retrieves the related information from the system configuration tables and generates the following SQL query.

SQL Query Generated :select desyear from department where dname="Computer Science"

The system finally executes the SQL query and displays the results as shown below.



V. COMPARATIVE ANALYSIS OF MULTIMODAL USER INTERFACE AND OTHER USER INTERFACES TO RELATIONAL DATABASES

a. Command-based user interfaces

Main Characteristic	Users specify their request for data from database by entering textual commands. E.g. SQL query
Knowledge required for the users	• Users should have the knowledge of syntax of a query language. • Users should also have the knowledge of the structure of database such as table names, attribute names, relationships to formulate a query.
Users' memory load	Users should remember the syntax of the query language and database structure details.
Use of users' language	No users' language is used in the interaction. Only artificial query language such as SQL is used.
Accuracy of results	Depends on the correctness in the formulation of query.
Prevention of errors	No facility is provided to prevent the users' errors. Users may submit wrong queries.
Ease of use	Not easy to use for the technically unsophisticated users.
Example	Structured Query Language (SQL) interface of DBMSs.