A Query Interface for Ubiquitous Access to Database Resources

Subhash Bhalla and Masaki Hasegawa

Graduate School of Computer Systems,
University of Aizu,
Aizu-Wakamatsu,Fukushima 965-8580 (Japan)
email: bhalla@u-aizu.ac.jp

Abstract - Most information systems depend on a Database Management System (DBMS) to organize information. Access to such an information system is based on use of a database query language on the part of the user. This poses user skills (or skill levels), as problem. For example, for medical staff at a hospital, users can not spare time to learn a database query language. As a result medical staff tend to depend on specialists, and programmers for access to information. To eliminate the complexities of a query language and to make a ubiquitous access a web based information access system has been proposed for the end users. In many cases, the proposed system allows semi-skilled personnel pose queries at the same level as a trained database query language programmer.

1 INTRODUCTION

We propose to address a User interface problem that is faced by users while accessing database contents on the web (1), (2). These are -

- Users are not skilled at using query languages, such as XQuery or SQL.
- Query languages support a complex structure as a statement (Figure 1).
- Web users have difficulty to verify a computation or its outcome.
- Existing query interfaces do not attempt to take advantage of skills that most users posses (3), (6).
- Web based user interfaces provide inadequate support to express a query. For example, consider the query -

  Find Names of all patients, who were examined by doctors who did a check-up of patient 'Smith'

  A user may have a fuzzy idea, as one of the doctors who did a checkup of patient 'Smith'. In such cases, users rely on Information Requirement Elicitation' (IRE) (4), (5). The web user is likely to perform this query in steps, by seeing the names of doctors and selecting one that matches with a sketch in his memory.

We propose an alternative set of interactions based on IRE approach. This is a calculator oriented approach using an object-by-object query. The proposed interface aims to eliminate ambiguities in users' communication by virtue of step-by-step procedure. The main contributions of this interface are its simplicity to express a query and its expressive power.

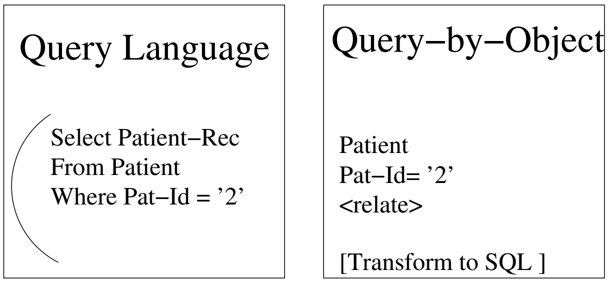

Figure 1: Web Based Query on Data Resources

2 BACKGROUND

2.1 HIERARCHICAL LINK FOR OBJECTS

Existing web-based information systems (WEB-IS) adopt a 'page and link' approach for access to data resources. Users select a related link as per their need. The use of such a hierarchical navigation helps the users in their search for the required piece of information.

The complexity of hierarchical navigation increases with increase in volume of data and information contents. The approach requires an extensive enumera-

International Conference on Management of Data
COMAD 2006, New Delhi, India, December 14–16, 2006
©Computer Society of India, 2006

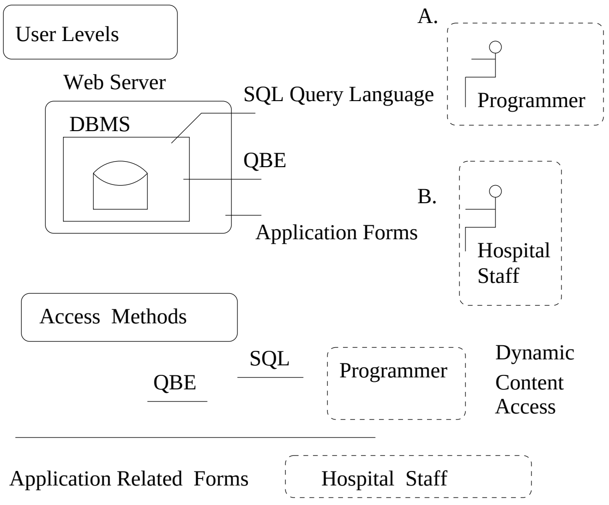

Figure 2: Web Users and Access Methods

tion of all possible query paths. It is not easy to combine multiple search criteria by using this approach (7).

2.2 NETWORKED LINK FOR OBJECTS

In view of the above difficulties, many WEB-IS provide 'input form and search' approach. The users are permitted to input selected key-words. Based on the inputs, the system attempts to locate the information contents for meeting the users' needs. The 'input form' approach helps to reduce the multiple criteria search difficulties that are posed by hierarchical navigation. However, this approach is suitable for restricted domains of information access as in a library or for ticket reservation. with complex queries expressed over large sized databases, the approach tends to have implementation difficulties (7).

2.3 INQUIRY BY RELATING OBJECTS

For accessing the dynamic contents through the medium of the web, the use of database query languages poses a new problem, as most web users are not skilled at the use of database query languages. Existing users at the non-programmer's level use application related forms (Figure 2.).

We propose query by relating objects. The proposed approach introduces an easier to use query interface that can express queries with more steps but with less complications of syntax and expression errors.

3 SYSTEM MODEL

Information Requirement Elicitation (IRE) is essential in wireless web service to elicit information requirements through interactive choice prompts. This paper presents the design of a high-level user interface using IRE, in the context of 'Hospital database queries' by a mobile web user. The prototype is based on the notion of Query-By-Object (QBO) approach of building a query using multiple user-level steps.

3.1 QUERY-BY-OBJECT INTERFACE

The Query-by-object (QBO) interface has been studied in (1) for Geographic Data Systems. A similar interface is being proposed for common database queries (2). In this system, users communicate with an RDBMS through a web interface. The user intent is captured via objects and path navigation through an option-based interface. Finally, a query is formulated and executed at a DBMS server after it is converted into its SQL equivalent.

The main options in the QBO interface are as follows :

1. Select an object - User selects an object. The object is included in the query bag (e.g, <Doctor>. The system utilizes displays a query bag to keep track of the user's actions.

2. Choose granularity - This option allows a user to see the contents of a database table that are instances or attributes of the selected object. The user can select an instance of the object (e.g., "Harry") or more than one instance (e.g., "Harry" and "John"). The object instance is then listed after the object name in the query bag (e.g., "<Doctor-Harry>".

3. Select another object - Users may select at most two objects before selecting an appropriate operator. This option allows the user to select the second object (e.g., "Patient") which will be listed after the first object in the query bag (e.g., <Doctor-Harry> <Patient>. The system further allows the user to select a subsequent object after completing each operation, in order to progressively build the query further.

4. Select an operation - Shows all possible pairwise operations for chosen objects (see Table I). The selected operation will then be reflected in the query bag (e.g., <Doctor-Harry;Patient - related visit>). A common requirement for most objects computations is the use of 'relate' as related operations. In this QBO interface, in order to simplify and standardize the menu options as much as possible, a join (or IN clause) operator is regularly employed as closely related binary operator. It has different connotations depending on the chosen objects it operates on (see Table I). For example, a relate operation on objects Doctor-Harry;Patient is used to test whether there are any Patients who were visited by Doctor-Harry.

5. Display result - This option allows a user to display the result of each query on the screen.
6. Reset - This option allows a user to start a new query (the query bag is subsequently emptied).

	Doctor	Check–up	Patient
Doctor	Union Intersect Complement	Relate	Relate
Check–up	Relate	Union Intersect Complement	Relate
Patient	Relate	Relate	Union Intersect Complement

Figure 3: WEB-IS query based on relational algebra.

3.2 QBO PROTOTYPE SYSTEM

A QBO prototype has been developed as a client-server application. The client part uses the Mobile Information Devices Profile (MIDP) of the Java 2 Platform, Micro Edition (J2ME), and the server part runs on a Java 2 Platform, Enterprise Edition (J2EE) application server. The application is designed as a basic three-tier architecture: Client Layer (MID let), Application Layer (Servlet, EJB), and Database Layer (PostgreSQL). The underlying database support is provided by an RDBMS PostgreSQL.

3.3 OBJECTS AND OPERATIONS

In order to support a user-level interface, objects and entities were stored in a database system. For this study a hospital example was used for preparing the database contents. The following database schema describes objects/relations of user interests. The primary key of each relation is represented in italics.

• Doctor = (*DOC-NO*, DOCTOR-NAME, SPECIALTY)

• Patient = (*PAT-ID*, PATIENT-NAME,PHONE-NO, DATE_OF_BIRTH)

• Check-up = (*DOC-NO, PAT-ID, DATE, TYPE*, ILLNESS, FEE)

The database contents can be described as a collection of objects (relations), as per the entity-relationship (ER) data model. In this case, Doctors, Check-ups, and Patients represent objects. An object instance is any specific record of an object, for example, "Harry" is an instance of object "Doctor". A set comprises two or more objects (or object instances). The different operations shown in Figure 3, show pairwise combinations of operations based on chosen objects. These are supported by operations available in an RDBMS.

4 IMPLEMENTATION OUTLINE

Consider an example,

Query

Find Names of all patients, who were examined by doctors who did a check-up of patient 'Smith'

Such a query, when expressed in database programming languages, such as, SQL (XQuery) or QBE requires complex steps, as shown -

SQL

```
SELECT  P2.Patient-Name
FROM Check-up AS C1, Patient AS P1,
         Check-up AS C2, Patient AS P2
WHERE P1.Patient-Name = 'Smith'
         AND  P1.Pat-ID =C1.Pat-Id
         AND   C1.Doc-No = C2.Doc-No
         AND   C2.Pat-Id = P2.Pat-Id
```

QBE

Patient	Pat-Id	Patient-Name	Phone-No
	_y	'Smith'	

Check -up	Doc -No	Pat -Id	Date	Type	Ill-ness	Fee
	_x	_y				

Check -up	Doc -No	Pat -Id	Date	Type	Ill-ness	Fee
	_x	_z				

Patient	Pat-Id	Patient-Name	Phone-No
	_z	.P	

In case of the proposed approach. The user aims at finding the response to the above query using natural (multiple) steps, as -

a) Names of doctors who did check-up of patient JAMES,

b) Names of patients, examined by these doctors (in part 'a)').

An intermediate user interface may need to support query language with step-by-step closure property to support users calculations. It must allow the user to perform the following steps with ease.

```
1. Choose object.
2. Inquire details.
3. Choose second/next object.
4. Choose an operation on these objects
5. Use closure property to continue
   at step 3.
```

4.1 Ease of Operations

The multiple steps approach to accessing a database is a naturally occurring, user level approach. It is easy for users (skilled and unskilled) to verify the unit steps and their outcome. Thus, in contrast to the earlier approaches, the proposed approach based on QBO is a simpler and easier approach, for web users.

4.2 Role of Web Interface

The support provided by the web interface software enables the functions, such as -

1. support for user level objects and operations,
2. 'Information Requirement Elicitation' (4), (5), for progressive formation of a query in case of the web users,
3. system support for prompts to include possibilities, of choosing related objects and possible operations, w.r.t. the chosen (first/available) object.
4. support for choosing granularity and enquiry about the attributes of objects,
5. support for relational algebra with closure property for the user objects

The above features enable the system to offer an easy to use query language interface.

5 CONCLUSIONS

With the advent of the web based information systems, it has become necessary to support a high level language for user interactions. The medium of user interaction must allow the user to express DBMS queries. The proposed system presents a high level language for user interaction for DBMS applications that are supported through the WWW. The step-wise navigation in the proposed language is based on tracking objects and paths logically and is supported by the SQL support provided by a RDBMS.

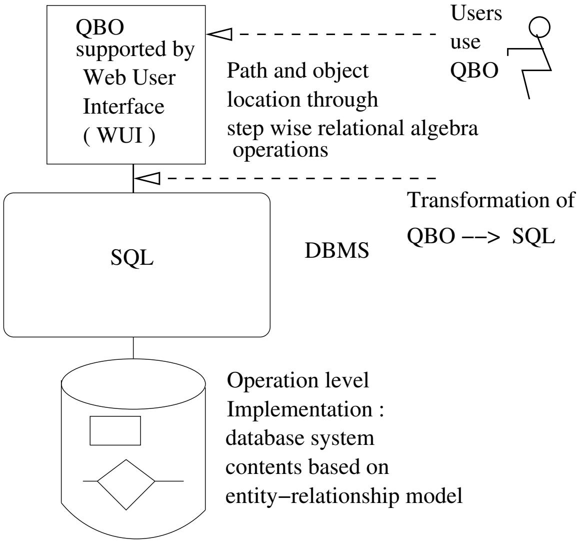
QBO supported by Web User Interface (WUI)
Users use QBO
Path and object location through step wise relational algebra operations
Transformation of QBO --> SQL
SQL
DBMS
Operation level Implementation : database system contents based on entity–relationship model

Figure 4: Implementation Sketch.

References

[1] S. A. Rahman, S. Bhalla, and T. Hashimoto, "Query-by-object Interface for Information Requirement Elicitation in M-commerce", Proceedings of 7th IEEE International Conference on E-commerce (CEC 2005), July 2005, Germany.

[2] S. Bhalla, M. Hasegawa, N. Berthouze, A Framework for a High Level User Interface for Accessing Dynamic Contents on the Web", 6th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Hosted by University of Milan, 16-18 September 2002, Italy.

[3] M. Derthick, J. A. Kolojejchick, and S.F. Roth, "An Interactive Visual Query Environment for Exploring Data," *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '97)*, ACM Press, October 1997, pp 189-198.

[4] Sun, J. Information Requirement Elicitation in M-Commerce - An Interactive Approach to Facilitate Information Search for Mobile Users, *Communications of ACM*, Vol. 46, No. 12, Dec. 2003, pp. 45-47.

[5] Sun, J., H.P. In, and K.A. Sukasdadi, A Prototype of Information Requirement Elicitation in M-Commerce, *Proceedings of IEEE International Conference on E-Comerce (CEC 03)*, pp. 53-56, June 2003.

[6] A. Labrinidis and N. Roussoppoulos, "Generating Dynamic Content at Database-backed Web Servers: cgi-bin vs mod_perl," *SIGMOD Record*, Vol. 29, No. 1, March 2000.

[7] A. Silberschatz, H. Korth, and S. Sudershan, "Database System Concepts," *McGraw-Hill Book Company*, Chapters on 'Hierarchical Data Model and Network Data Model', 5th Edition, 2005.