A Database Query Language for Visual Interfaces

Kyoji Kawagoe
C&C Systems Research Laboratories,
NEC Corporation
Miyazaki, Miyamae, Kawasaki,
Kanagawa 213, Japan

ABSTRACT
This paper describes a database query language needed to produce a so-called visual database interface, an important element in improving current database interfaces. The interface is based on a term input technique, which is sequence independent i.e. in which a user can input any predefined terms in any order to specify a query. The more he uses this interface, the better appropriate results he may be expected to obtain. This paper presents an interpretation technique, based on a conceptual graph, which is used to interpret a database query represented in this database query language. The language can be applied to the production of a visually-oriented interface, one in which terms are visualized.

1. Introduction
In the area of database studies, many query languages for accessing information have been proposed [CAEG76] [Zloo75] [Inte82] [HeSW75] [ErFr85] [Hero82]. However, they are hard to learn and to use, because of their complex syntax and of the needs for extensive knowledge about the database itself. For example, even in the popular relational database language SQL [CAEG76], users have to know the names of attributes and relations, as well as reserved keywords, such as "where" or "group-by".

It is with this problem in mind that the author presents here a new kind of language for end-users to use in obtaining easy access to their databases. This language is called IFUL/IR, which stands for Interactive and Flexible User Interface Language for Information Retrieval. IFUL/IR is a database interface with AI techniques, which is intended to be used as the basis of a visual interface. It is also intended to provide flexible interface to users of a query construction. The word input sequence used in constructing a query using IFUL/IR is completely free within the range of past usage, and users need to worry about neither the syntax nor keywords. This method is referred to as a "sequence independent" input technique. In order to produce IFUL/IR, the author developed a query interpretation technique, based on the conceptual graph presented by Sowa for natural language processing, employed as an internal query representation.

IFUL/IR is distinctive in its approach to query interpretation. When a user, who intends to ask a computer a question for the first time, constructs a query in a free order of words, the query is interpreted in a standard way. With each succeeding query interpretation, the query pattern -- frequently used words or favorite patterns for word sequences -- are accumulated. The proposed approach is completely different in regard to its basic principles from natural language database interface [GaLM85] [Inte82]. This principle is based on the concept that the natural languages used in knowledge engineering are hard and tedious to learn and use, because they are neither completely English nor Japanese. Therefore, it would be important and necessary to develop new intelligent interfaces, which would enable users to input what they think, entirely in the manner in which they normally think. IFUL/IR is just a beginning towards accomplishing this aim.

At first, the language, IFUL/IR, may not be fully appropriate for such an end-user's query, and its language interpreter would perform nothing appropriate in response to his query. The user has to specify the meaning of the query by selecting the most appropriate meaning from the candidates produced by the system. As the user teaches the system the way he/she wishes to specify his/her query, the system learns the appropriate usage method and automatically constructs language semantics and syntax, which are most appropriate for him/her. Finally, the language becomes the most efficient for this one specific end-user. Even when his/her usage method is changed, the system has the ability to adapt to his/her changing, utilizing the same mechanism as was the initial learning activity.

The IFUL/IR architecture is shown in Fig. 1. The main language user is considered to be computer novices, rather than database experts, who are familiar with QBE- or SQL-like DB languages. Intermediately skilled users can also use the IFUL/IR, in order to obtain desired information as efficiently as possible. Visual interfaces can be realized by assigning visual symbols and picture elements to the predefined terms of IFUL/IR. For visual interfaces, the sequence-independent input technique is very important and useful, because users need not understand the order involved in selection and specification needed for meeting their requirements.

This paper first presents a concept of "term" which is the essential concept for IFUL/IR. Then, an interpretation method, using a conceptual graph, is described. This paper also explains an application of this language.
Novices DB Experts

Term Seq.

Query Construction (SQL, QBE,...) Access

Fig. 1 IFUL/IR Architecture

2. Term oriented query language

First, a simple example is presented to ease understanding the proposed IFUL/IR language.

Assume that there is an employee database and that a user wants to obtain some information from the database, using SQL. One of the queries is as follows:

SELECT NAME FROM EMP WHERE EMP.DNO = SELECT DEPTNO FROM DEPT
WHERE DEPT.MGR = "John"
(List the names of employees in the department whose manager is John.)

For the query, instead of the SQL query, some user may want to specify his query as follows:

NAME EMP DEPT MGR "John"

If he has enough time to specify his requirement with SQL, the statement will be the same as the above complete SQL statement. However, in this case, it is assumed that he wants to obtain the information as soon as possible. So, it is more desirable that SELECT, FROM, WHERE and = are all excluded from his SQL specification.

Another user may consider another requirement, as follows:

MGR "John" DEPT EMP NAME

This case is a reverse English language sequence, which means the user may be accustomed to using Japanese.

In addition, users, who are doing both updating and selection, may input "SELECT EMP DEPT MGR "John"", in order to ensure their intention specification.

The examples show that the difference in the word sequence lies in some aspects of users characteristics, including the users native language, idea representation method and concept formulation pattern or query targets, mostly constructed by the experience history for an individual user.

The sequence-independent input technique, which is presented in this paper, enables users to specify their queries in any of the above mentioned ways. So, users need not learn a database language, while they can still select any desired information utilization method.

The fundamental concept in the proposed interface is a term. Elements in a query are all terms. So, the term is either an entity name, an attribute name, a value domain, a value itself or any keyword. In the case of SQL, term examples are SELECT, EMP, WHERE, "Mr. XYZ", "35", Address and so on. The users only select or input those terms he wishes to use from a set of displayed or predefined terms in order. For the first previous example, they can select "NAME", "EMP" and "DEPT" and input a string "John", instead of specifying "SELECT NAME FROM EMP WHERE EMP.DNO = SELECT DEPTNO FROM DEPT WHERE DEPT.MGR = "John". A sequence of the input ordered terms forms a user query, as shown above. Any order can be permitted. However, for SQL, SELECT has to follow any other description, such as a WHERE clause or a target-list. From the terms input by users, the language processor transforms the term sequence into the formatted query representation of the SQL[CAEG76], INGRES[HeSW75] or other languages, employing the usage pattern which was used in the past.

As explained before, the Conceptual Graph plays an essential role in interpreting the language. The Conceptual Graph (CG), proposed by Sowa [Sowa84], is a kind of Semantic Network. As in the semantic network, the CG is composed of a set of arcs and nodes. However, in the CG, each node is associated with various types and values, while in the semantic network, an individual node is only identified by its name.

One CG example is shown in Fig. 2. As indicated in Fig. 2, the CG is well-structured, representing the semantics of sentences.

In order to accumulate data on usage patterns for individual users, the following Weighted Relationship Matrix (WRM) is utilized, whose elements correspond with usage frequency for the consecutive terms. In other words, Weighted Relationship Matrix (WRM) is defined as follows:

\[ WRM = \{ WRM_{ij} \} \]

where \( WRM_{ij} \) = frequency value used in the query so far for meaningful ordered pair \((term_i, term_j)\).

For instance, if \( term_i = EMP \) and \( term_j = NAME \) have been consecutively specified in queries five times, then, \( WRM_{ij} \) is set 5, because the pair \((EMP, NAME)\) is meaningful as an employee's name. Otherwise, if \( term_i \) and \( term_j \) have been consecutively used ten times, then \( WRM_{ij} \) has a value 10. However, even if meaning-

378
less pairs, such as EMP and PART, have frequently been used in many queries, the value of elements for these terms in WRM remains zero or the value is given an infinite symbol value. No value is accumulated for this pair. The decision as to whether a pair is meaningful or meaningless is made in interpreting queries and is discussed later.

In the case of values, such as "10" and "John", value types, such as INTEGER, CHARACTER or domain names, such as PERSON-NAME are WRM elements, instead of these values.

Using the above two kinds of information, a term sequence can be interpreted. The brief interpretation mechanism is as follows:

One WRM example is shown in Fig. 3.

First, from the given term sequence, a query conceptual graph is generated by matching it with the given conceptual graph referring to the WRM. Then, the query conceptual graph is transformed into a query statement for a target database processing. Last, after query processing performed by the database management system, the WRM elements are updated, in order to unveil the query interpretation result for the next query processing. The first step is called term-and-semantic analysis. The second step is called query generation. The last is called WRM updating.

The query generation is very simple, as can be seen in Sowa's book [Sowa83]. The WRM updating is also very simple, since it is only necessary to modify existing values in the WRM elements, which correspond with a set of term pairs in the query conceptual graph. Therefore, the first term-and-semantic analysis step is described in the next section.

EXAMPLE

The above mentioned query interpretation mechanism is explained for the queries given previously.

Consider the query used to retrieve the name of an employee whose department manager is John. It is assumed that a user specifies the query as " ? EMP NAME DEPT MGR John". Moreover, it is also assumed that WRM and CG data have been obtained, as shown in Fig.4. Then, for the query example above, the query interpretation process becomes as shown below.
cepts. In this example, concepts, EMP, NAME, MGR and DEPT, are marked, and concept value, John, is set in the concept MGR. Then, the above graph is generated.

At this time, the natural language sentence for the query is first generated from the CG, then is presented to the user. The user is asked whether this query interpretation is correct or not. If a "yes" reply is indicated, then the system goes to the next step. Otherwise, it goes back to Step 1 for re-clustering. In this example, in case that the user lies this interpretation result, Step 1 generates another possible clustering, such as the following:

```
[ EMP [ MGR ? ] ] [ DEPT [ NAME "John" ] ]
```

[Step 3] (Database query construction with checking)
In order to obtain a database query, it is necessary to represent the query for the underlying database schema. So, we get

```
SELECT NAME IN EMP E WHERE E.DEPTID = SELECT ID IN DEPT D WHERE D.MGR = "John"
```

After this, unknown variables are found. In this example, such an unknown variable is NAME. Then, the query graph is navigated and the obtained database query statement is as follows:

```
SELECT NAME IN EMP E WHERE E.DEPTID = SELECT ID IN DEPT D WHERE D.MGR = "John"
```

[Step 4] (WRM updating)
Query CG, which resulted in Step 2, includes meaningful pairs of terms, which are (EMP, NAME), (EMP, DEPT), (DEPT, MGR), (MGR, TEXT) and (NAME, ?). Combining them with the input term sequence, ordered pairs, such as (EMP -> NAME) and (?, -> NAME) are obtained. Then, for each ordered pair, the corresponding element value in WRM is updated.

The query interpreting process for the above example may not seem to be complex. However, for a complex sentence, it is necessary to repeat the above processing from Step 1 to Step 3, while structuring the terms, generating the query CG and checking it. In order to modify the term structure and the CG, other clustering and matching techniques are used, which are not further discussed in this paper, due to space limitations.

3. Query Interpretation

The following describes how a query, a sequence of terms, is interpreted to generate the database query statement.

Interpreting the query, which is diagrammatically shown in Fig. 5, is carried out as follows:

[Step 1] Generating term structure, using Clustering
For each term in a given term sequence, check relationships among the terms, the previous term and the next term, using

```
Term Structure
Clustering
```

```
Matching with CG
```

```
Natural Language
Sentence Generation
and User’s Decision
```

```
Query Generation
```

```
Database Access
```

This is the process of clustering. The grouping process uses a method similar to the Clustering technique \[Rome86\]. Examples of distance function are:

- \( w_1 \cdot f(WRM_{ij}) + w_2 \cdot g(Distance of terms in the term sequence) \)
- \( f(WRM_{ij}) \cdot g(Distance of terms in the term sequence) \)

where \( w_1 \) and \( w_2 \) are appropriate weighting values.

[Step 2] Matching structure with CG
After grouping the term sequence, structured term structures are matched with CG. The basic matching process is as follows. First, for each term in the term structure, mark a concept related with the term. Then, if the term represents a value, not a type, generate the value in the concept. If the concept has already been marked, then either a new node, that is a new concept, should be generated, representing the same concept node, or the current term from the query interpretation should be neglected, considering consistency. Finally, a query CG is generated by connecting nodes with reference to the original CG. In case that there is a term, with which no concept in the CG is related, a new node is created. Moreover, the query CG is constructed so as to meet constraints from the clustering result. That is, for each cluster, no path between concepts within the cluster must contain other concepts related with another cluster. After the CG generation, the natural language query sentence is generated from the CG for confirmation by the users. The query sentence generation is shown in Sowa’s book \[Sowa83\]. If the sentence is confirmed to be correct: in other words, if the query recognized by the system is the same as the query that the user wants to present, then the processing continues to the next step. Otherwise, the query interpretation is incorrect, and it has to be processed in another way. The alternatives are generated by changing the weighting values in a distance function or by changing the threshold value in clustering calculation.
3. Generating database query statement

The Query CG, which is a result of the matching, is used for generating a database query statement. The query generation first involves transforming the query CG into a query graph for the underlying database schema. Next, unknown variables are detected. Then, nodes are navigated from nodes with unknown variables via query CG, and the query is formed through node navigation. If the query CG has some error, such as no unknown variables or recursive query, this process is halted. Then, in order to change the term structure or the query CG, backtracking has to be carried out. That is, go to Step 1 or Step 2, depending on change possibility. When no change can be made, but when query CG is still not understandable, the interpretation process will fail and it is necessary to change the original CG or changing the clustering technique is required.

After generating the database query, normal query processing is carried out by conventional database management systems.

4. Updating WRM

By unveiling the query interpretation result, WRM is updated for the next query processing. Consider a pair of terms, which are consecutive and are connected with meaningful arcs in query CG. In other words, there are no nodes between two nodes corresponding the pair of the terms, or there are several nodes between them, but these nodes don’t correspond with any terms in the given set of terms. For this term pair, the value of the corresponding element in the WRM is increased by one.

This interpretation method has the following features:

1) Avoiding ambiguity with Clustering, CG and WRM

Motro indicated in [Mor86] that the token oriented query construction causes ambiguity in queries. He proposed an approach to resolving these ambiguities in query interpretation. This approach involves the use of priority, interaction with users and user decision. However, these ways to approach ambiguities are so complicated and informal that an efficient and easy query construction by users would be discarded if it were formed. In contrast with his method, the author’s WRM and CG usage makes it possible to detect an appropriate term among several alternatives within the range of past experience. CG plays an important role in regard to checking and constructing queries from a semantic viewpoint, while WRM decreases the number of ambiguities, based on experience.

2) Capable of complex sentence interpretation, with Clustering and CG:

In order to handle more complex sentence, such as complex and compound sentences, the method presented in this paper includes pre-processing of a given term sequence. This method breaks down the sequence into a set of term groups. Then, semantic processing, using the Conceptual Graph, checks it and constructs a reasonable users’ query.

For a complex sentence, such as "Find all employee names in Bob’s department", any free order query such as EMP NAME ? Bob DEPT, can be interpreted in the same way as was involved in the above technique, which is discussed later.

4. Visual Interfaces using IFUL/IR

Some applications of using the IFUL/IR with visual interface facility are presented.

The first application deals with a simple database on weather information, in several cities in Japan.

An interface, as in Fig. 6, is used to access the database. In Fig. 6, the display area is divided into four sections, one for general term selection at the top, one for city selection, second from the top, one for selected term sequence display, third from the top, and one for the result from information retrieval with the formatted query specification at the bottom.

In Fig. 7(a), a user has just selected some icons and cities from a set of icons of the top area and from the picture in the second area from the top. The query shows "What is today’s weather in Tokyo". As the icons and cities in the picture are corresponded with terms, the query is represented by a term sequence "TODAY WEATHER TOKYO". The result after query processing is also shown in Fig. 7(b). Another query, with the same query content as the previous one is also shown in Fig. 7(c). The result is the same as the previous query result.

As this application shows, users don’t need to learn an entire query language syntax. They only need to select visualized symbols from a set of icons and pictures presented to them, in order to indicate their intentions. The more often the language is used, the more the time involved in the icon or picture selection is decreased, due to the language adaptability.
(a) Initial IFUL/IR Display

(b) Display after a user constructs a sequence of terms (upper left area) from a set of icons (middle left area) The term sequence given in English: "TV-PROGRAM" "PERFORMER" "Mr. Aoki" NOT-EQUAL "SPORTS" (END)

(c) Display after the first query interpretation The interpretation result is described in Japanese, which is shown in the middle left area. The sentence means Is "Display TV non-sport program where a performer is Mr. Aoki" OK?

(d) Display after the second query interpretation, because the user gave the system "No" in (c). The second sentence means the following: Is "Display TV sport program where a performer is not Mr. Aoki" OK?

(e) Display after database access for the second query interpretation. The result is briefly displayed in image form. The image shown in the upper right area is one of the TV programs in newspapers.

(f) Display of more detail information on TV programs. The result for details is performed by selecting one of the sub-images in the TV program image on the upper left area. The image shown can be scrolled up/down/right/left with a scroll menu down the TV program image.
values for the attributes previously applied, there are many icons hidden under the icons are displayed and can be input. The upper left area is to be used for TV program information retrieval. In the lower left area a set currently displayed icons. By selecting an applicable button for displaying term sequence histories, while the right half on a "mouse" or on special purpose input instrument, other image.

A set of figures in Fig. 8 shows user operation for obtaining information employing a sequence of terms. Therefore, benefits of IFUL/R, connected with a visual representation, like icons and pictures, are as follow:

1) Visual representation of terms provides end-users with an easy way to use the language, understand it more easily and freely employ their own communication methods.
2) The representation can easily be applied to a new innovative media technology, such as images and voices, which is necessary to be appropriately manipulated within the database languages range.
3) All user concepts, from generalized to a basic ones, can be represented in a unified way, with visual representation of terms.

5. Discussion

1) Related work

Recently, Motro has presented a new query language similar to the authors' approach [Mot86]. He uses a concept, called 'token' and an interpret method, using a Dependency Graph. A query is constructed as a set of tokens. Each token represents either a value, a relation name, or an attribute name. However, his approach differs in regard to the following points.

One difference lies in the definition of tokens. The token should not contain any other word, except values, entity and attribute names. The authors permit employing any word useful in constructing a query, such as SELECT or AND. The second difference is the query complexity. His interpretation method is so simple that more complex queries, such as 'SELECT DNO FROM EMP A WHERE A.AGE = SELECT AGE FROM EMP B WHERE B.NAME = "John"', could not be processed. Instead, the author introduces a Conceptual Graph [Sowa84] and Weighted Term Relationship Matrix for complex-query interpretation, which makes it possible to process a complex sentence or a context. The last difference, which is the most point stressed, is query processing adaptability. Motro's query processing method is very rigid and independent for query methods from specific individuals. However, the authors insist on the need for adaptability in query processing, as explained above. To realize this capability, a method is also presented with the above weighted relationship matrix. Usage patterns for an individual user are stored in the matrix element as usage frequency.

Several papers have been published putting adaptable capability into database query languages [WoKu82], [GGKZ85], [Zloo75], [King85]. The adaptable methods were crucially related with the registration of a sequence of commands or changes in names. The language syntax and semantics cannot be modified and tuned to individual users at all.

Regarding visual database interfaces, several prototype systems with visual database interfaces, such as IBS [ErFr85] and CUPID [Mcdo82], have already been presented. These systems tend to demonstrate possible visual interfaces for specific applications. So, they contain neither concept nor mechanism for general-purpose visual Database interfaces.

On the other hand, many database research reports [GGKZ86], [King85], [WoKu82] have focused on a visually oriented interface using a diagrammatical data model, such as the Entity-Relationship model. With the systems, the user can directly select and manipulate the diagrams on displays in order to gain access to databases. An example of such systems is GUIDE, which uses a fixed network to support the Entity-Relationship model. Common problems involved in these approach are that the user needs a lot of time to understand the model and model manipulation, before actually using the system. He also has to operate a number of selection, pointing and data entries for taking advantage of the diagrammatical interfaces. It is very difficult to combine diagrams and other visual information, such as images and drawings.

As stated first in this paper, natural language interfaces [Inte82], [IYH87] have also been developed in an attempt to make systems more user friendly. However, no existent systems have understood and supported the every-day languages in much the same way as human beings communicate. Rather, quite a lot of times is consumed in specifying a query in a restricted language with low-level input instruments.

2) Complex sentence case

Assume that the user wants to present a query:
List the names of employees whose department manager's age is 50.
by specifying the following term sequence.
NAME EMP ? DEPT MGR AGE 50
The query in this term sequence is corresponded with more than a single conceptual graph, because the term AGE is either employee age or manager age. It is important to determine which is appropriate for the user. The purpose of the clustering task, which is the first step in the query interpretation, is to reduce ambiguity of terms used in consideration of term distance obtained from his WRM. The result of clustering for the same WRM as in the previous section, is as follows:

[ [ EMP [ NAME ? ] ]

When the user happens to indicate "employee age is 50 " , not " manager age is 50 " , the first clustering structure makes mistakes. However, another possible clustering structure can be generated, finally enabling a user to select an appropriate result.

3) Implementation Consideration
Up to now, the author has explained concepts and mechanisms for IFUJJIR. However, for implementation of the interface, several items have to be taken into account.
- When and how CG is to be defined and how WRM is to be assigned.
- The CG is defined when a database manager designs database structures. The content of the CG is obtained from the database structure with some simple transformation. On the other hand, WRM is initially set to default values, so as to reflect concept relationships in the CG. Any user has his own WRM, so the system can process a query construction appropriate for individual users.
- How a large number of icons is displayed.

In order to display a number of icons representing terms, the icons are classified and segregated. The clustering method is also useful for the classification.
- How low is the interpretation costs
Interpretation costs are composed of clustering cost, matching cost and query generation cost. Among them, the clustering cost seems to be the largest. However, it is considerably lower than that of the natural language processing. As an example of the clustering cost, less than one second in CPU time is required on a UNIX workstation.

6. Conclusion
This paper has presented a concept of a term based query construction. The language enables constructing a query in any sequence or term order, so that a user can individually select a term representing what is desired. To construct a user query from a sequence of terms, the author has presented a query interpretation mechanism, using the clustering method and the Conceptual Graph. To provide adaptability, the author also has proposed weighted Relationship Matrix.

Implementing this language is under further development, while it has already been developed to a successful demonstration level. From the limited reaction reported by a few users, it is felt that this kind of language is very useful for both end-users and experts. However, several problems remain to be solved, including a more flexible mechanism and more extensive usage pattern extraction. It is hoped that this flexible and sequence-independent input technique as a basis for achieving visual interfaces, will become a next generation database language.

Acknowledgement
The author wishes thank Dr. S. Goto and Dr. T. Ishiguro for their continual support to this research, and referees for their useful comments on the draft.

REFERENCES