A psychological study of query by example

by JOHN C. THOMAS and JOHN D. GOULD
IBM T. J. Watson Research Center
Yorktown Heights, New York

INTRODUCTION

Many different query systems have been proposed. One way to partition extant and hypothetical query systems is on the basis of how English-like they are. One approach for an easy-to-use query system is to allow the user to state a question in natural English. The system may then disambiguate the possible interpretations of this question on the basis of context or on the basis of feedback questions to the user. A second approach is to require the user to state his question in a formal language system, but one that uses an English-like grammar and vocabulary. IBM's Interactive Query Facility (IQF) and SEQUEL* exemplify this type of approach. A third approach is to require the user to state his question in a formal language system that does not attempt to appear "English-like." Zloof's Query By Example language is best described by this third approach. In this paper we demonstrate experimentally the ease and accuracy with which nonprogrammers learned and used this powerful Query By Example language.

It may seem that the first approach mentioned above, viz., allowing the user to state his question in natural English, must necessarily be the best approach and the one we should study. There are several reasons why this is not so. First, any natural English system in the foreseeable future is likely to place serious restrictions on the vocabulary and syntax of allowable, or at least useful, user inputs to the system. It may be difficult for the user to keep in mind such restrictions since daily conversation may tend to provide strong interference with the rules of such a system, and hence produce considerable forgetting. In this experiment, we provide evidence that Query By Example may be robust against such forgetting. Second, as a user becomes more familiar with a system, particularly if his job involves some percentage of fairly routine question asking, lengthy feedback dialogues may come to be perceived as a waste of time. Third, the need for this dialogue will increase the cost of the computer system.

A fourth factor relates to data representation. Should the system represent data relations in natural English or in a more formal representation? Although it might be assumed that natural English would be the optimal way to represent data to users, studies of problem solving provide evidence that sentential representations are often non-optimal for humans. Providing the user with a formal representation, such as Query By Example does with tables that implicitly supply relations among the data, may better help the user formulate and solve his problem.

Our primary purpose was to conduct an exploratory evaluation of a powerful query language that seemed promising for non-programmers to learn and use. Of specific interest were the time to learn the language, and the time, accuracy, and subjects' confidence in translating test questions stated in English into Query by Example. These test questions were selected from disparate regions of the space of potential questions, making sure to include some that required each of the features of the language. This was done to determine the factors that made this translation process difficult. We included some questions that involved concepts that previous work had shown are difficult for most people (e.g., conjunction vs. disjunction constructions and universal quantification) and some questions studied with other query languages.

METHOD

Subjects

Subjects were run in four successive groups; these consisted of four college students, 11 college students and recent graduates, and two groups of 12 high school students. Data on class standing and IQ were available for 23 of the 24 high school students. The mean class standing was 44/197 and the median IQ was 115. There were 15 males and 24 females. The subjects ranged in age from 16 to 24 years. Four of the college students had some minor programming in school courses. Aside from this, none of the subjects had any experience in using computers.

Design and procedure

Subjects received about one hour and 45 minutes training on the major features of Query By Example. They then wrote translations of 20 test questions; this took about 40 minutes. After a short break, subjects received another 70 minutes of instruction, followed by another 20-question test, which took about an hour. The second test contained some questions designed to assess understanding of the concepts presented during the second training period, and it also contained some questions similar to...
those on the first test. The purpose of these latter questions was to measure the effect of adding other features (for example, set inclusion) to a subject's understanding of the "easier" part of the language. Two weeks after initial learning, six college students were available for retest. These students were given a test of 20 questions without any retraining. They were then given an hour refresher and given another test of 20 questions.

During testing, subjects were not given feedback about the correctness of their answers. If they had been given such feedback after writing each query, they would undoubtedly have written a higher percentage of queries correctly. They also could not refer to instructional material or notes used in training. However, subjects were provided a list of abbreviations and operators in the front of the test booklet. The order of questions within a test was rotated in five different sequences among subjects so that fatigue or practice effects would not systematically affect the estimates of question difficulty.

Query by example language

The Query By Example language that subjects were taught is described in detail elsewhere. In this experiment subjects were given empty tables, containing only the table names and column names. Subjects wrote a query by filling in some of the columns in one or more of the tables with examples (variables), with constants, and with operators. Table I illustrates some of the features of the language.

The first query in the table represents the question "Print the names and salaries of the people in the sports department." The "p." in the columns labelled NAME and SALARY shows that those are the items to be printed out. "Jones" and "23K" are examples. They are underlined because they are examples. "Sports" is not underlined because it is a constant.

The second query in Table I represents the question "Print out the names of those people who work in the same department as Riley." "Toys" is underlined to indicate that it is a variable, in this case a linking variable. It links "Riley" to the people that will be printed out. These two lines could be interchanged.

The last line in the top table combined with the bottom three tables is the query for "Who are the people who work in a department that sells items supplied by companies located in Massachusetts." This problem requires the use of four tables, and it illustrates how tables are linked together. Note that Massachusetts is the only constant in this question; the other words are examples (variables) and hence are underlined.

Training

Subjects were trained in groups. Training was based primarily upon a lecture during which subjects were shown, via an overhead projector, a series of examples of about 100 queries written in Query By Example. Each example was explained in the lecture, and subjects were required throughout training to code many English questions into Query By Example. To do this, they were given forms that contained outlines of the data tables on which the training queries were based. Besides the lecturer, there was a second instructor present, and throughout training both instructors continuously checked on the accuracy of the students' queries. In this way both the students and the
TABLE II—Example Queries Used in Evaluation

1. Print the names of employees whose salary is less than $12,000, are over 28 years old, and are managed by White.
2. I think there must be about 30 people managed by White, who make $12,000. So . . . list the people over 28 who are managed by White. Of course, I'm only talking about those who make less than $12,000.
3. I'm thinking about a raise for someone from White's group. I want only older people who are underpaid. For starters, find the people who work for White, are over 28, and make less than $12,000.
4. Who else works in the same department as Riley?
5. List the names of employees who are younger than Anders' manager.
6. List people who work in departments that sell at least one item supplied by a company located in Massachusetts.
7. Print the names of anyone who makes more than Anders' manager and is younger than Smith's manager.
8. The accounting people need to know how many married women over 30, with no dependents, work in Dept. 300; and how many single employees who have at most a Junior College degree have worked for us for more than 5 years and are employed in St. Louis, in Dept. 400.

Teachers had feedback while each concept of the language was being taught.

Task

Following training, each subject's task was to translate 40 questions stated in English into Query By Example. Table II provides some examples. These questions varied in several ways. Most were stated in a straightforward way. In some cases, formally identical problems were also stated in both a poorly expressed manner and in a way which included a rationale for asking the question. The first three examples provide examples of these variations. Queries also varied in the number of linking variables required: 0, 1, 2, 3, or 4. Sometimes these links were within a single table and sometimes between tables. Examples of queries requiring one linking variable within a table, two linking variables within a table, three linking variables among tables, and four linking variables within a single table are given as examples 4 through 7 in Table II. Questions also varied in the number of conjunctive constraints, the number of disjunctive sets, and in the number of operators used to compute quantities from a subset. Example 8 in Table II illustrates a questions with two disjunctive "sets", with each set requiring five conjunctive constraints. In addition, the question asks for an operation of counting to be performed.

Subjects were also required to write the time that they started reading each English question and the time that they completed writing each query. In addition, 35 subjects gave a confidence rating between 1 and 5 to indicate how sure they were that each query was correct. A "1" corresponded to "very sure correct", a "2" corresponded to "fairly sure correct", a "3" corresponded to "50-50 chance", a "4" corresponded to "fairly sure incorrect", and a "5" corresponded to "very sure it's incorrect."

Error analyses

The correctness of the queries that the subjects wrote was assessed by two people familiar with the syntax of the Query By Example system.

RESULTS

Overall

Mean training times varied between two hours and two hours and 55 minutes for the four groups. The proportion of correct queries was 0.67; mean time to complete a query was 1.6 minutes overall (s.d. = 1.09); mean confidence rating was 1.8 (s.d. = 0.82).

Individual differences

All subjects were able to learn the language and complete the experiment. The performance of the college students and high school students was nearly identical on the above four measures, and there also were no sex differences. The range of individual differences was 0.33 to 0.93 on proportion of queries correct (F(38,1482) = 5.24; p < 0.001); and 1 to 2.3 on confidence ratings (F(34,1326) = 12.12; p < 0.001). Thirty-four of the 39 subjects were correct on over 50 percent of the questions. Subjects who were confident also were relatively accurate (r = 0.63; p < 0.001). Subjects who were relatively fast in writing queries were not significantly more confident (r = -0.19; p > 0.05). The correlation between Otis IQ and accuracy on the first 20-question test was 0.69 (p < 0.01) and 0.50 (p < 0.05) on the second 20-question test.

Problems

The proportion of subjects writing correct queries for each of the 40 questions varied from 0.26 to 1.0 (F(39,1482) = 12.59; p < 0.001). Twenty-six of the 40 questions were done correctly by the majority of subjects. The mean time to write queries varied from 0.83 minutes for the fastest problem to 3.64 minutes for the slowest problem (F(39,1482) = 26.12; p < 0.001). The mean confidence ratings varied from 1.3 to 2.5 (F(39,1326) = 10.24; p < 0.001). The correlation on the first 20-question test between mean accuracy and mean time to write a query for that question was -0.70 (p < 0.001). On that same test, problems on which people were accurate were problems which they were also confident about.
**TABLE III—Proportion of Queries Correct for Various Confidence Ratings**

<table>
<thead>
<tr>
<th>CONFIDENCE RATINGS</th>
<th>Very Sure Correct</th>
<th>Fairly Sure Correct</th>
<th>50-50 Chance</th>
<th>Fairly Sure Incorrect</th>
<th>Very Sure Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportion Correct</td>
<td>0.83</td>
<td>0.62</td>
<td>0.44</td>
<td>0.17</td>
<td>0.20</td>
</tr>
<tr>
<td>Total N</td>
<td>581</td>
<td>585</td>
<td>187</td>
<td>36</td>
<td>10</td>
</tr>
</tbody>
</table>

($r = -0.92; p < 0.001$); problems which they did quickly also were associated with high confidence ratings ($r = 0.78; p < 0.001$). These correlations were significant ($p < 0.01$) but lower on the second 20-question test.

Table III shows the probability of a problem being correct, given a particular confidence rating. As is clear from the table, a computer system could make a useful prediction about the correctness of a particular query by asking a user how likely he thinks it is that the query is correct.

**Accuracy predictions**

Three types of measures were used to predict the proportion of correctly written queries.

First, a linear multiple regression based on four parameters of the correct answer, viz., the number of columns required by a correct answer, the number of different operators, the number of rows, the number of linking variables, produced an $R$ of 0.74 ($p < 0.001$). Addition of more Query By Example parameters would probably not increase the proportion of variance accounted for (54 percent) dramatically.

The second type of measure was the mean confidence rating of the subject on a question. This correlated with proportion correct ($r = 0.86; p < 0.001$). When combined with the above four parameters, $R$ reached 0.92 ($p < 0.001$).

Third, two simple measures that reflected the particular English formulation of the question were calculated. One was simply the number of words in the English question. The other was the number of occasions in which an operator necessary in Query By Example was only implicit in the English phrasing of the question. For simplicity, this variable will be called "mismatches". If these predictors are included, then $R$ reached 0.95 ($F(9,29) = 26.98; p < 0.001$).

**Comparative data**

Seven questions used by Gould and Ascher in a behavioral evaluation of an IQF-like language were also used in the current experiment. These questions differed in the number of sets of records that were required to be retrieved (1, 2, or 3) and in the number of conjunctive modifiers required for each set (1, 3, or 5). Mean proportion correct for these questions was 0.54 for Query By Example and 0.30 for Gould and Ascher. The mean total time for these seven queries was 2.1 minutes in Query By Example and 7.7 minutes in the Gould and Ascher experiment. Subjects in the latter experiment were required to write a formulation and a plan prior to coding a query, and this coding time was 3.6 minutes. Times to write queries in the two systems were highly correlated ($r = 0.98; p < 0.001$). The correlation between the total number of modifiers and the time to write a query was significant both for the Gould and Ascher study ($r = 0.99; F(1,7) = 251.5; p < 0.001$) and for Query By Example ($r = 0.99; F(1,5) = 235.7; p < 0.001$).

**Particular English wording**

In order to assess the effects of the particular English wording of a question, three special sets of questions were used. Each set consisted of three different English formulations ("straight", "poorly expressed", and "rationale") that were intended to map into the same formal Query By Example. Providing subjects with a rationale did not have any apparent effect on time or accuracy. The irrelevant information in the "poorly expressed" questions produced some reduction in accuracy for all three question sets, but in no case were these differences statistically significant. Confidence ratings and times were also not consistently affected by wording of the question. In one problem, subjects took twice as long (2.2 minutes) for the "poorly expressed" question as for the "straight" version. In this case, the difference was significant ($t(76) = 4.23; p < 0.001$). It should be noted, however, that variance for the "poorly expressed" question was 15 times the variance for the "straight" version. It would seem that the inclusion of irrelevant material in a question to be translated has a large effect, but only on a subset of the population.

**Interference effects**

Three pairs of questions of identical objective complexity and equally well-stated were included in the experiment. For each pair, one question was included on the first 20-question test and one on the second 20-question test. The purpose of these pairs was to assess the effects of the second training session (primarily how to specify universal quantification) on the ability of subjects to retain what they had learned during the first training session. There were no major differences in proportion correct, mean time, or mean confidence rating on these pairs of questions. These findings demonstrate that net fatigue, practice or interference effects were minor. However, during the second test, there were 18 cases in which universal
quantification constructions were used when they were not needed.

Output classification

The output that would have resulted from each query, if an implemented system had been used, was divided into several categories (not mutually exclusive); e.g., exact output, no output, wrong output, superfluous output. The main result was that 18 percent of the 33 percent incorrect queries would produce no output. Eighty-two percent of the time that a subject made an error, some set of records would have been returned.

Error types

An incorrect query could contain more than one error. Table IV indicates the relative proportion of occasions when a particular Query By Example construction was required but was either omitted or used incorrectly. The data indicated, for example, that subjects made mistakes one-fourth of the time when they needed to use a universal quantification operator. Subjects also made mistakes on one-fourth of the occasions when they needed to use COUNT, SUM, AVERAGE, or COMPACT-COUNT operators. Overall, subjects made errors on 3 percent of the occasions in which comparison operators (e.g., >, <, >, <) were needed. Subjects made underlining mistakes on 1 percent of the occasions that they needed to decide whether to underline. The subjects made accurate decisions about "and/or" constructions and seldom used the wrong value. A common source of error not shown in the table were "mismatches" between Query By Example and English. For example, an English question might ask for "the people who have worked for us for more than five years." The tables used had an attribute labelled "Year of Hire." This question, therefore, required subjects to translate the English phrase "more than five" into "<1969".

Although not counted as an error, subjects many times left off the periods that were meant to terminate operators. Subjects were not penalized because a well-engineered implementation would include buttons that would automatically add terminators. An interesting type of error that arose on some problems was labelled "sexist syntax." For example, some subjects did not specify sex when asked to retrieve records for female secretaries.

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Error Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantification</td>
<td>0.25</td>
</tr>
<tr>
<td>Count/Sum/ Or = Count</td>
<td>0.22</td>
</tr>
<tr>
<td>&gt;, &lt;, &gt;, &lt;</td>
<td>0.03</td>
</tr>
<tr>
<td>Underlining</td>
<td>0.01</td>
</tr>
<tr>
<td>And vs. Or</td>
<td>0.003</td>
</tr>
<tr>
<td>Wrong value</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Retest data

One group of 11 subjects was asked to return after a two-week interval and six of them did. On their original tests these six subjects were slightly above average. (Their mean accuracy was 75 percent correct.) On an immediately given 20-question retest, (consisting of 10 questions from the original tests and 10 new but comparable questions) they were 55 percent correct. One subject wrote no correct queries because she forgot to underline variables. A second subject reversed the sense of '<' and '>' and wrote only 25 percent correct queries. The other four subjects were as accurate on the fourteenth day after training as they were on the first day. After an hour review of Query By Example, these six subjects wrote correctly 66 percent of their queries on a different (equally difficult) 20-question test.

DISCUSSION

Comparative data

The results indicate Query By Example is easy to learn and use in a relative sense and perhaps in an absolute sense also. Compared to the Gould and Ascher data on an IQF-like language, subjects using Query By Example required about one-third the training time, were about twice as accurate in writing queries, and were somewhat faster. Compared with Reisner's pilot data on two new query languages called SEQUEL and SQUARE, our subjects required about one-third the training time and appear to be about equally accurate as those using SEQUEL or SQUARE. No subjects failed to complete training in Query By Example, unlike with these other three languages. Of course, these comparisons are only approximate because the separate experiments differed in several ways, e.g., training techniques, subject populations, test questions, stringency of scoring errors.

Absolute considerations

In an absolute sense, it is hard to imagine a powerful formal language system which could be learned much more rapidly than Query By Example. Indeed, even a "natural English" query system, which at first blush might seem to require no training time, would in fact require considerable time for subjects to learn the many exceptions and restrictions that any foreseeable "natural language" system will require. Also, the time to write a "natural language" query, and then have a computer system, through interaction with the user, disambiguate it, would probably exceed the time to write in Query By Example. There are no solid reasons for believing that accuracy in writing "natural language" queries would be substantially higher than that found here for Query By Example. Further, the fact that subjects returned in two weeks and wrote queries nearly as correctly as they did when they initially learned the language suggests that
people will retain Query By Example better than they would retain a “natural language” system.

Predicting errors

The probability of a subject’s making an error in writing a query is largely predictable from four parameters of Query By Example (the number of different columns used, the number of different operators, the number of rows, and the number of comparison operators) and from how confident a subject was about his query being correct. These five predictors accounted for 84 percent of the accuracy variance, but this would somewhat overstate the case in real life due to “shrinkage”. It should be noted that these predictions are not based upon the actual parameters of the queries that subjects wrote, but rather upon those of a correctly written query. While we have not done the former analysis, informal inspection indicates that about the same predictive accuracy would result. When the SEQUEL experiment is complete, it will be interesting to learn whether the probability of an error on a particular test question translated into SEQUEL could also be predicted from the parameters required for that question if it were translated into Query By Example.

Feedback

The practical significance of predicting an erroneous query is that a computer system can make a useful prediction on whether a user’s query is in fact the one he intended. If there is a high probability of his making an error, then this prediction can trigger various kinds of feedback to the user that might help him catch his error. This is particularly important in view of the finding that a syntactic check would have only stopped 18 percent of the incorrectly written queries in this experiment. This feedback could be tailored to the type of likely error. For example, the computer might restate the user’s question in another form. Or, the computer might produce alternative, but similar, queries and the user would then verify whether the one he wrote was in fact the one he intended. Or, the computer might display a small data table. Selection of particular cases within the table, either by the user or by the computer, might help in distinguishing between an intended query and an unintended one.

Weak points

Subjects made two major types of errors. First, they frequently confused SUM, COUNT, and COMPACT-COUNT operators. Presumably this error can be reduced through system constraints, feedback, and modification of the names of the operators. Second, they had difficulty in correctly specifying universal quantification when it was required. For example, subjects might render the question “List companies whose entire line of items is sold by Toys” into formal queries that really meant “List companies which have an item sold by Toys” or “List companies which supply the entire line of items sold by Toys”.

The possible reasons for quantification errors are numerous, and include the possibility that subjects did not have sufficient conceptual abilities to use quantification constructions, that they were not taught that part of Query By Example well, or that they did not understand those test question statements. Alternatively, perhaps they understood the test questions and knew the language well enough, but they could not “put it all together”. Perhaps they knew how to use the quantification constructions, but they did not know when to use them.

The subjects’ task in translating from English into a formal language depended upon the number of “mismatches” between the particular English formulation of the question and the exact way it needed to be stated in Query By Example with the particular data tables used. This indicates that in real world settings two considerations would reduce errors. First, the data tables should be labelled to reflect the way people think about and express a subject area, and, second, the person who formulates the question in English should write the query himself or at least be aware of the ways of expressing it in the formal language. Since Query By Example can be learned so easily and queries written so quickly, it is hoped that the person formulating the question will enter his own query directly.

Good language features

In Query By Example, subjects rarely confused disjunctive and conjunctive queries, and they had little trouble linking attributes within a single table or linking multiple tables together. We observed during training that subjects sometimes seemed to understand a complex English question more easily if it were shown in Query By Example than when it was stated in English.

Positive characteristics

What are the characteristics that make Query By Example good? We are not sure, but we believe the following to be important. First, the user is given an explicit representation in which to formulate his query; he does not have to generate it free style. While this may be particularly true for the small tables (few columns) used in this experiment, it may not be true for much larger tables. (Zloof has designed an extension of Query By Example to allow the user to select certain portions of a large data base which would then be queried.) Second, the particular type of representation given, i.e., tabular, may be especially helpful (compared with, for example, a hierarchical or set representation). Third, the system is nearly wordless. This prevents many natural language confusions, e.g., the use of an “and” in specifying disjunc-
tive concepts ("all the physicists and children"). Fourth, the system is easy enough to learn so that people's motivations are high. (Some subjects thanked us for teaching them this language that they will probably never use in real life.) The training technique of showing subjects a series of example queries, rather than providing them with extended explanations of how the language worked, seemed quite effective. Fifth, the language is "behaviorally extendable", i.e., a novice user need only learn a small part of it to write successful queries for simple questions. Subsequently, if his problems require it, he can build upon this knowledge by learning more of the language. APL is also like this.

**Query language users**

Most query systems in present use have been designed for people who know their application well and regularly use the system as a main part of their job. As in the case of airline reservations systems, these users are expected to spend days or weeks learning the system because they then can use several short-cuts that lead to system efficiencies. At the other extreme is Codd's "casual user", who can be thought of as a browser or dilettante who may even sample information or problems and applications he knows nothing about. Somewhere in between are people, including "professionals", who wish to use a computer creatively and in a flexible and powerful manner on applications they know about, but without spending an inordinate amount of time learning to use the system. Our results suggest Query By Example should be especially useful for this latter group.

**Research problems**

Formal behavioral investigations prior to implementing a query or programming language are rare. This experiment demonstrates feasibility, and our experience is that computer scientists take seriously the results and implications. There remain many important behavioral research problems on the design and use of query systems.

In this experiment subjects *translated* a test question into a single query. In real life, people often *generate* their own questions, sometimes a related series of them, to gain the information they need. The cognitive processes involved in translating and generating questions are not identical. We plan to study how people generate questions, and we hope to learn about how people use the information they have already obtained, how they determine when they have arrived at an acceptable answer, and how the characteristics of their problem affect the types of questions they ask.

The types of questions people would ask of a data base in natural language need to be identified and compared with those that people would ask using formal languages. Data need to be collected not only on how non-experts (as in this experiment) write queries but also on how experts would query large data bases to formulate and answer complex questions of real importance to them.

**ACKNOWLEDGMENTS**

We wish to thank Brian Madden for helping to grade the 1560 queries and for his many suggestions for system implementation. We also wish to thank Vivian Clingman and Cay Dietrich for their excellent secretarial services, and Bill Scholz and John Parkman for their many suggestions on this manuscript.

**REFERENCES**
