

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

12-1995

The effects of conceptual and logical interfaces on visual query performance of end users

Keng SIAU

Singapore Management University, klsiau@smu.edu.sg

H. CHAN

K. WEI

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#)

Citation

SIAU, Keng; CHAN, H.; and WEI, K.. The effects of conceptual and logical interfaces on visual query performance of end users. (1995). *Proceedings of the Sixteenth Annual International Conference on Information Systems (ICIS 95), Amsterdam, The Netherlands, December 10-13*. 225-236.

Available at: https://ink.library.smu.edu.sg/sis_research/9424

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylds@smu.edu.sg.

THE EFFECTS OF CONCEPTUAL AND LOGICAL INTERFACES ON VISUAL QUERY PERFORMANCE OF END USERS

Keng L. Siau

Faculty of Commerce and Business Administration
The University of British Columbia

Hock C. Chan

Kwok K. Wei

Department of Information Systems and Computer Science
National University of Singapore

Abstract

To the end users, the interface is the system. A better interface not only facilitates end user interaction with the database, it also enables them to formulate queries more efficiently and effectively. Two of the most important user-database interfaces are the conceptual and logical interfaces. With the conceptual interface, the user communicates with the database system in terms of entities, objects and relationships. On the other hand, the current user-database interaction is mainly based on the logical interface where the user expresses the queries in terms of relations and join operations. Because the concepts at the logical interface are abstract and convoluted to ordinary users, many researchers argue that end users will be better off with the conceptual interface. This research will test this claim by comparing the effects of conceptual and logical interfaces on the visual query performance of end users. The experimental study involves three tests: an initial test, a retention test and a relearning test. This allows us to assess the learning effect over time. The results show that users of the conceptual interface achieve higher accuracy, are more confident in their answers, and spend less time on the queries than users of the logical interface in all three tests.

1. INTRODUCTION

Databases are vital organizational resources. They contain information that is necessary for the functioning and survival of organizations. However, to fully utilize these resources, users have to be able to retrieve the data efficiently and effectively. The user-database interface is the key to the success of this process. A better interface increases the productivity of the users by allowing them to interact with the database system more efficiently and effectively (Chamberlain 1980). Gerlach and Kuo (1991) also stressed that end-user productivity is tied directly to functionality and ease of learning and use of the interface.

User-database interface can be broadly classified into the conceptual, logical, and physical interfaces. The physical interface is the lowest level while the conceptual interface is the highest. At the lowest level, the user is required to know the details of the data structures in the computer memory. An example of such an interface is assembly language. When using the logical interface, the user is required to know the layout of the logical data and the possible relationships among the data

elements. This is exemplified by the relational database interface (Hawryszkiewicz 1990; Navathe 1992). There are no physical pointers or physical files and the order of the columns and rows is not important. However, the user knows that by joining relations based on certain fields, it is possible to specify relationships that at the physical level are represented by physical pointers. In fact, it is common to refer to the joins as the logical pointers.

At the conceptual level, the database is supposed to know the user's world, in terms of entities and relationships. There are no logical pointers for the user to trace. If the user wants to specify a relationship, he may express it in a natural and straightforward manner such as "where supplier supplies part" rather than "where supplier.sno=sp.sno and sp.pno=part.pno." Data models suitable for this level of interaction are the entity relationship (ER) model (Chen 1976; Navathe 1992) and the object-oriented (OO) model (Kim 1990; Elmasri and Navathe 1989).

The current user-database interaction is mainly based on the relational interface, which is a logical interface, where the user

expresses the queries in terms of relations and join operations. The main interface for relational systems is undoubtedly SQL, a language that was developed almost twenty years ago (Chamberlain and Boyce 1974). Despite its popularity, SQL was found to be very difficult to use, even for trained users (Greenbalt and Waxman 1978; Welty and Stemple 1981). The problems with existing high level query languages such as SQL and QUEL have motivated the design of new graphical interfaces and query languages to bridge the gap between naive users and database systems (Kim, Korth and Silberschatz 1988). A number of graphical query languages have been proposed, such as GRAQULA (Sokut et al. 1993), PICASSO (Kim, Korth and Silberschatz 1988), CUPID (McDonald and Stonebraker 1975), and QBE (Zloof 1977). These languages, nevertheless, are still based on the logical interfaces.

Many researchers argue that a conceptual interface might be better for the end users. For example, Markowitz and Shoshani (1989) noted that,

In order to express database queries, users are often required to understand large, complex database structures. It is important to relax this requirement...so that they can manage with partial, or even no knowledge of the database structure.

With the increasing popularity of end-user computing and empowerment of end users, the ability to encode and express the queries directly and intuitively is even more important. This concern about data utility is also clearly reflected in the recent surveys of CIOs and managers by McCromick (1991) and Niederman, Brancheau and Wetherbe (1991) which list data utilization as one of the top MIS issues.

Despite the importance of user-database interface, there is a paucity of research in this area. Kim noted that "high-level user interfaces remain an area of research not only for object-oriented databases, but also for conventional databases." In this research, we attempt to investigate the claim that the conceptual interface is superior to the logical interface by looking at the query performance of end users.

2. LITERATURE REVIEW

The study by Chan, Wei and Siau (1993) compares the conceptual level versus the logical level using the entity-relationship (ER) model and an ER query language (i.e., Knowledge Query Language) at the conceptual level, and the relational model and SQL at the logical level. The results show that users of the conceptual level were 38% higher in accuracy and 16% higher in confidence level, and took only 35% of the time taken by users of the logical level. However, the study examined the difference between conceptual and logical level interaction on a cross-

sectional basis. It does not test the effect of learning over time. In this study, we extend the study by Chan, Wei and Siau by looking at the learning effect over time. This is accomplished using three tests: an initial test, a retention test, and a relearning test.

Jih et al. (1989) compare user query performance at the ER and relational interfaces. The users were given either the ER or relational model but answered the queries using the same query language, Structured Query Language (SQL). The results indicated no significant difference in the semantic accuracy of the queries. Nevertheless, users of the relational interface took a longer time but made fewer syntactic errors. The experiment, however, did not make a clear distinction between the two interfaces. SQL was used in both the ER and relational groups because of the concern that there might be interaction effects between the data model and the query language. SQL is inherently designed for the relational interface. SQL users must understand the logical pointers and specify the join operations. Hence, the users of the ER interface, like the users of the relational interface, had to go down to the logical level and manipulate the logical pointers. This may explain the lack of difference in semantic accuracy. Furthermore, users answered only four queries, two simple and two complex ones. The two simple queries involved a single relation whereas the two complex ones both required a join across two relations. Complex queries involving nesting and the use of the "Not Exist" clause were not tested.

The study by Davis (1990) tested the standard documentation (i.e., list of table contents), data structure diagram and two variations of entity-relationship diagrams to assess their impact on performance of database queries. The results show that graphical forms of documentation are significantly better than the conventional textual documentation (i.e., list of table contents) but none of these graphical forms of documentation (i.e., diagrams) appeared to be superior to the others. One possible confound in the study is that the subjects had been introduced to relational databases during their MIS course and had been taught the functions of the principal types of relational query commands such as SELECT, PROJECT, and JOIN before they were assigned to one of the four experimental conditions. As such, the subjects may have been more familiar with the data structure diagram since it is a direct representation of the relational tables and less familiar with the entity-relationship diagrams because these have an additional relationship construct that is new to the students. This might explain for not finding a difference between the data structure diagram and the entity relationship diagrams.

In addition to these studies, several researchers have also looked at the effect of data models on modeling (e.g., Batra, Hoffer and Bostrom 1990; Jarvenpaa and Machesky 1989; Batra 1993; Batra and Sein 1994). For example, Batra, Hoffer and Bostrom studied user performance in database modeling. The results showed that

user performance in a representation task using the EER model, as compared to the relational model, was better. Jarvenpaa and Machesky examined the Logical Data Structure (LDS), which is based on the entity-relationship concept, and the Relational Data Model (RDM) in three learning experiments. Their findings indicate that LDS promotes the top-down approach and results in significantly higher accuracy than RDM.

This research will empirically evaluate the effect of conceptual and logical interfaces on query performance. Unlike previous studies, this research will look at the effect of learning the two interfaces over a period of time. Specifically, three tests (initial, retention and relearning tests) spanning a period of three weeks were conducted. Also, rather than looking at textual interfaces (e.g., SQL) as was normally the case, we examine visual interfaces (e.g., QBE) in this study. The examination of visual interfaces will allow us to complement the results from text-based studies.

3. A COMPARISON OF CONCEPTUAL AND LOGICAL INTERFACES

The difference between the two interfaces can be discussed in terms of the computer-human interface model developed by Hutchins, Hollan and Norman (1985). This model has been utilized frequently in the IS literature (e.g., Batra, Hoffer and Bostrom 1990; Suh and Jenkins 1992). The model explains the relationship between the cognitive effort required to accomplish a task and the distance between the user's goals and the way these goals must be specified to a system. One of the concepts in the model is the notion of semantic distance. Semantic distance concerns the relationship between the meaning of an expression in the interface language and what the users want to say (Hutchins, Hollan and Norman 1985). For example, a small distance means that the translation is simple and straightforward, that thoughts are readily translated into the database queries. Two important factors concerning semantic distance are:

- (a) Can the users say what they want to say in this language? In other words, does the language encode the concepts in the domain in the same way that the users think about them?
- (b) Can the users say what they want in a straightforward fashion, or must they construct a complicated expression to accomplish what they perceive as a conceptually simple task?

According to the model, an interface that provides the users the ease of encoding and expressing a query will have a shorter semantic distance than one that is unnatural and artificial. At the conceptual level, the user expresses the concepts in the domain in the same way that he or she thinks about them (Chan, Wei and Siau 1993). The interface allows the user to use concise and transparent encoding of the queries without bothering about the

database structure. However, with the logical interface, the knowledge will need to be forced into its representational conventions in an artificial and uncomfortable way that is understandable to the system. In other words, the user has to map his or her psychological variables (i.e., objects and relationships) to those that are used by the system (e.g., relations). As for the expressive power, simple knowledge can be encoded easily using the conceptual interface whereas it is laborious and tedious to encode the same knowledge using the logical interface. Additional cognitive effort is required to do the transformation using the logical interface. Thus, it is hypothesized that the use of conceptual interface will reduce the semantic distance and in turn lead to better user performance.

Another popular model in cognitive science is the capacity model of attention. According to this model, successful task performance is generally seen as a matter of resource requirements in relation to resource availability. Navon (1984) defines resources as "any internal input essential for processing that is available in quantities that are limited at any point in time." Cognitive effort is defined as the percentage of the available capacity or resources allocated to a given task (Mitchell and Hunt 1989). Under this paradigm, performance of any task will be influenced by resource requirements of the task only if resource demand exceeds resource supply. That is, cognitive effort is theoretically relevant only when processing requirements outstrip the available processing capacity. When resources are limited, the theory clearly predicts that the fewer the resources required by a task, the greater the probability that sufficient resources will be available and the higher the probability of success.

The ease of encoding and the expressive adequacy of the conceptual interface result in less cognitive effort required. Because humans have limited working memory, this capacity model predicts that conceptual interface will produce better performance when the task is complex. In other words, when the interaction is simple as in the case of a simple query and the resource capacity is not exceeded, there will be no noticeable difference between the two interfaces. However, once the resources required exceed the capacity of the working memory, as in the case of a complex query, performance using the conceptual interface will be better because of the lesser need for limited resources.

Thus, the various theories favor the conceptual interface over the logical interface. The hypothesis, however, needs to be verified scientifically. The next section of the paper describes an experimental study conducted to test the hypothesis that the conceptual interface is indeed better than the logical interface.

4. RESEARCH MODEL

For this study, the independent variable is the interfaces and the dependent variable is the query performance. We controlled for

the task and user characteristics. This is in line with the research model proposed by Reisner (1981), where a survey of laboratory studies on query languages showed frequent use of tasks, data model, and user characteristics as factors affecting user performance.



Figure 1. Research Model

4.1 Interfaces

The two levels for this independent variable are the conceptual and logical interfaces. The two levels were operationalized as either the ER model with an ER query language or the relational model with a relational query language. The construct validity of choosing the ER model for the conceptual level and the relational model for the logical level is well documented in textbooks on database design (McFadden and Hoffer 1985; Elmasri and Navathe 1989; Hawryszkiewicz 1990; Teorey 1990; Hughes 1991; Batini, Ceri and Navathe 1992). As pointed out by Vossen (1991, p. 197), "the conceptual design of a database has so far been based on the entity-relationship model." Similarly, Navathe argued that the conceptual data model is developed using the ER model and the logical data model is typically one of hierarchical, network, or relational model.

Unlike the study by Jih et al., we made the level distinction complete by providing a query language, Visual Knowledge Query Language (VKQL), for the ER model. The query language for the relational model is Query By Example (QBE). Hence the conceptual interface was operationalized by the ER model and the ER query language VKQL whereas the logical interface was operationalized by the relational model and the relational language QBE (Zloof 1977).

For the ER model, there is no commonly used ER query language. In this study, we used VKQL as the ER query language. Examples of the VKQL query language and the details of its syntax can be found in Siau, Chan and Tan (1991, 1992). VKQL is a full language designed with the ER model in mind. It comprises both the definition language (VKDL) and the manipulation language (VKML). It includes concepts such as generalization, specialization, categorization and inheritances. Like QBE, VKQL allows for arbitrarily complex queries and it is relationally complete. It also supports nested queries and includes statistical functions such as count, average and sum.

QBE was selected because it is one of the most popular visual relational query languages. Thomas and Gould (1975) found that QBE subjects required about one-third the training time and appeared to be about as equally accurate as those using SEQUEL or SQUARE. Another study by Greenbalt and Waxman concluded that QBE was "superior to SQL in learning and application ease." Moreover, QBE's syntax closely resembles that of VKQL (Siau, Chan and Tan 1991). Both use the table-structure for specifying queries. This removes any unwanted extraneous factors such as menu versus graphics or graphics versus command-line.

4.2 User Performance

The dependent construct of query performance was operationalized by three variables: the accuracy of the queries, the time taken to formulate the queries, and the subjects' confidence in their queries. The time taken was automatically captured by the computer program. The confidence level was self-reported by the subject for each query and was computer-recorded. The accuracy measure was an overall assessment of the correctness of the answer by two professors. Accuracy and confidence were measured on a scale of 0 to 5. Timing was recorded in seconds.

5. RESEARCH DESIGN AND PROCEDURES

This experiment involved three tests: the initial test, the retention test and the relearning test. This three-stage study allows us to assess the learning effect over time. The initial test was conducted two weeks prior to the retention and relearning tests.

- Initial Test: Test how easy it is to learn the query language. This test was given at the end of a training session.
- Retention Test: Test how easy it is to re-use the query language after a period of disuse (no training, no refresher).
- Relearning Test: Test how easy it is to re-learn the query language after not using it for a period of time.

5.1 Hypotheses

The hypotheses (stated in null forms) for this study are as follows:

Initial Test

- Hypothesis H1:** There will be no difference between the conceptual and logical interfaces in the initial test for the three dependent variables (i.e., time, accuracy, and confidence level).

Retention Test

Hypothesis H2: There will be no difference between the conceptual and logical interfaces in the retention test for the three dependent variables.

Relearning Test

Hypothesis H3: There will be no difference between the conceptual and logical interfaces in the relearning test for the three dependent variables.

The alternative hypotheses to the null hypotheses are that the two means will be different.

5.2 The Subjects

First-year computer science students were the subjects for the experiment. Fifty subjects were randomly selected from a population of 480 and randomly assigned to the conceptual and logical groups. The subjects were about twenty years old. They had some computing experience but no database experience. Novice subjects were selected because we wanted to study learning effect over time. Using intermediate or expert subjects would be problematic because of their prior exposure to database query languages, which may be a confounding variable.

We tried to motivate the subjects in performing the task by giving them some course marks based on their speed and accuracy, as well as the correlation between the accuracy of their queries and their self-reported confidence level. This would encourage them to report their confidence level honestly rather than to indicate excessive confidence. The number of subjects that turned up for each test is summarized in the table below.

Table 2. Number of Subjects that Participated in Each Test

Test	Logical	Conceptual
Initial	24	19
Retention	16	18
Relearning	16	18

Seven subjects were absent from the initial test. Nine subjects from the initial test did not turn up for the retention and relearning tests. A check at the results of the initial test shows that those subjects who were absent after the initial test were no different from other subjects in terms of test scores. As such, we do not think that their absence would bias the results. Only those

subjects that turned up for all three tests were included in the statistical analysis. This gave us a total of thirty-four subjects with eighteen subjects in the conceptual group and sixteen subjects in the logical group.

5.3 Training of Subjects

Two different training booklets were used for the study. The booklet for the logical group contained a brief description of the relational data model and an extensive illustration of the QBE query language. The conceptual group was provided with the booklet describing the ER model and the VKQL query language. Extra care was taken to ensure that the two booklets were identical in terms of depth, comprehensiveness and "reader-friendliness." To maintain consistency, the same database containing suppliers and parts and the same examples were used in both booklets. Both the conceptual and the logical groups had the same trainer. A training session lasting about 45 minutes was conducted for each group where all examples in the training booklet were explained and discussed. Subjects were then given a practice session that lasted about 30 minutes to familiarize themselves with the software by repeating all of the fourteen training examples in the booklet. This was to ensure that the time measured in the test reflects as accurately as possible the query formulation time rather than confounding it with the time spent learning the software.

5.4 System Characteristics

The characteristics of the system were controlled by having both of the systems on the Macintosh machines. The systems, written using Hypercard software, were essentially a simple interface, customized to display queries and record answers and other data. Examples of the screen interface are shown in the appendix. These systems have many advantages over a pencil and paper system: it is more realistic, it provides automatic timing, and the subject cannot go back to previous answers whereby timing will be seriously jeopardized. All the Macintoshes used in this experiment have the same configuration.

5.5 Initial Test

After the practice session, the subjects were given a ten minute break before taking the initial test. Ten questions on a different database domain were given one by one on the screen. Subjects had to enter the answers on the screen. They were allowed to refer to the training materials and to use paper and pencil for rough work. Timing was done by the computer. The timing started when the subject clicked on the New Query Button displayed on the screen and ended when the subject clicked on the Done or Ready Button. After each question, the subjects had to enter their confidence in their answer. This is an integer value ranging from 0 (zero confidence) to 5 (absolute confidence). The

subjects were told that the time taken to fill in the confidence level would not be counted as part of the timing.

The same set and order of questions were given to the two groups. The conceptual subjects were given a picture of the ER model on paper. The logical subjects were given the relational schema on paper. The same database domain about departments and employees was used for both groups. The test questions attempt to cover all the basic queries that can be made on the ER model and the relational model. The ER and relational schemas as well as the test questions for the two groups are shown in the appendix.

5.6 Retention Test

The retention test was conducted after two weeks of disuse (no studying, no refresher). The same ten questions on the same database domain used for the initial test were given to the subjects. No training or practice session was provided. The subjects were allowed to refer to the training booklets and to use paper and pencil for rough work.

It should be noted that this retention test was slightly different from those conducted by Welty and Stemple (1981), Thomas and Gould (1975), Reisner (1977), and Greenbalt and Waxman (1978). Those retention tests were closed book tests whereas in this case the retention test was an open book test. We felt that a closed book retention test was not realistic and of little practical value. In practice, even professional programmers refer to software manuals rather than relying entirely on their memory. As such, the aim of the retention test is to test the ease of reusing the query language after a period of disuse rather than the ease of memorizing the query language.

5.7 Relearning Test

After the retention test, the subjects were given a ten minute break. This was followed by a practice session using the computer software. The same set of fourteen questions that was used for the practice session prior to the initial test was used for this practice session. Questions from the subjects were answered by the trainer. Immediately after the practice session, the subjects were asked to take the test again. Similarly, the subjects were allowed to refer to the training booklets and to use paper and pencil for rough work.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The accuracy of the students' answers was determined independently by two markers. Both markers are university professors with an average teaching experience of four years. Each of the students' answers would yield a maximum of five marks and a minimum of zero marks. Marks were awarded based on both the

syntactic and semantic accuracy of the answers. The marks assigned by the two markers were very close with at most a two-point difference; the overall correlation coefficient of the two markers is 0.95. This indicates a very high reliability for the measure of accuracy. In the statistical analyses that follow, the average of the marks assigned by the two markers was used.

The next table shows the means and standard deviations (given in brackets) for the three dependent variables of time, accuracy and confidence. The maximum values for confidence and accuracy are both five. Time is given in seconds.

All the null hypotheses were rejected at the 0.01 level. In other words, the subjects performed significantly better using conceptual interface for all three tests and for all of the dependent variables. These results supported the basic hypothesis that the users perform better using conceptual interface than logical interface.

The results of this study confirm previous results by Chan, Wei and Siau which show that users of the conceptual level perform better than users of the logical level in terms of accuracy, confidence, and time. However, the results are different from those of Jih et al., which show little difference in performance when the ER and relational model were compared. We identify two reasons for the discrepancies. The main reason is that a special ER query language, VKQL, was used in this study, making the distinction between ER and relational interfaces complete. Jih et al. used SQL for both models, hence the distinction was not clear.

The other possible reason is that we tested the subjects with a more comprehensive set of queries — from simple to complex queries. In fact, for simple queries that involve only one relation, our study also indicates no significant difference in accuracy because most of the subjects score the full score of five points. This is in accordance with the prediction by the capacity model which states that cognitive effort is theoretically relevant only when capacity requirements outstrip the available processing capacity. For simple queries, the working memory capacity is not exceeded and hence there is no difference in performance between the ER and the relational interfaces. However, the same cannot be said about the complex queries. Due to the limited amount of working memory, an interface that requires a query to be expressed in a complex and convoluted way takes up more working memory space and therefore leaves little working memory for the actual processing of the problem solving task. For those complex queries where more cognitive capacity is required, our results show that there is a huge difference between conceptual and logical interfaces.

The percentage differences of the conceptual group over the logical group are shown in Table 3.

Throughout the three tests, the VKQL group is consistently better than the QBE group in all three dependent variables. In the initial test, the VKQL queries were formulated in 49% (i.e., 69.54/

Table 2. Scores for Dependent Variables in the Three Tests

Test	Dependent Variable	QBE Mean (Std. Dev.)	VKQL Mean (Std. Dev.)	t	p(Prob > t)
<i>Initial</i>	Time	141.36 (103.93)	69.54 (44.16)	8.11	0.0001
	Accuracy	4.07 (1.11)	4.59 (0.85)	-4.88	0.001
	Confidence	4.06 (1.25)	4.69 (0.59)	-5.92	0.001
<i>Retention</i>	Time	116.82 (72.42)	65.56 (41.44)	7.88	0.0001
	Accuracy	4.25 (1.08)	4.61 (0.79)	-3.47	0.0006
	Confidence	4.09 (1.19)	4.62 (0.94)	-4.52	0.0001
<i>Relearning</i>	Time	65.24 (44.37)	41.22 (21.38)	6.23	0.0001
	Accuracy	4.36 (1.01)	4.69 (0.66)	-3.59	0.0004
	Confidence	4.42 (1.11)	4.83 (0.52)	-4.31	0.0001

Table 3. Percentage Differences Between the Two Groups

Test	Dependent Variable	QBE Mean	VKQL Mean	Percentage Difference
<i>Initial</i>	Time	141.36	69.54	-50.81
	Accuracy	4.07	4.59	12.78
	Confidence	4.06	4.69	15.52
<i>Retention</i>	Time	116.82	65.56	-43.88
	Accuracy	4.25	4.61	8.47
	Confidence	4.09	4.62	12.96
<i>Relearning</i>	Time	65.24	41.22	-36.82
	Accuracy	4.36	4.69	7.57
	Confidence	4.42	4.83	9.27

141.36) of the time taken for the QBE queries; they were about 13% more accurate and the VKQL users were 16% more confident in their answers. The timing of the QBE group improved in the retention test but the percentage difference between the QBE and VKQL groups is still a huge 44%. The percentage difference in timing after relearning is 37%, suggesting that the queries using the conceptual interface are easier to express and encode than the queries using the logical interface. The huge difference in timing between the two interfaces confirms the theoretical prediction that the conceptual interface has a much shorter semantic distance than the logical interface.

Although the percentage differences in accuracy and confidence between the two groups narrowed in the retention and relearning tests, they still ranged from 7.57% to 12.96%. The narrowing of percentages between the conceptual group and the logical group is probably due to the "ceiling" effect of the conceptual group. Since most of the subjects in the conceptual group did very well

in the initial test, there was little room left for improvement in the retention and relearning tests. This suggests that the conceptual group is able to attain high performance much faster than the logical group.

7. CONCLUSION

Based on the cognitive theories by Hutchins, Hollan and Norman and by Navon, it is hypothesized that the conceptual interface is better for the users because the interaction conforms to the user's views of the world without consideration of the physical implementation characteristics of the system. This basic hypothesis is then empirically tested in a comprehensive study involving three tests: initial test, retention test and relearning test. This "longitudinal" design enables us to investigate the effect over time. The results indicate that users of the conceptual level not only exhibited higher accuracy and higher confidence in all three tests, they also took less time than users of the logical level in the tests.

The results thus provide strong empirical evidence that conceptual interface is indeed better than relational interface. What is the implication of the results for practitioners? Our experimental results show that users' productivity, in terms of accuracy and time, can be significantly improved when they switch from a logical interface such as relational to a conceptual interface such as entity-relationship. With the conceptual interface, end users can also expect to have more control over their database resources.

8. ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for the many valuable comments and suggestions.

9. REFERENCES

- Batini, C.; Ceri, S.; and Navathe, S. B. *Conceptual Database Design: An Entity Relationship Approach*. Menlo Park, California: The Benjamin/Cummings Publishing Company Inc., 1992.
- Batra, D. "A Framework for Studying Human Error Behavior in Conceptual Database Modeling." *Information and Management*, Volume 25, 1993, pp. 121-131.
- Batra, D.; Hoffer, J. A.; and Bostrom, R. P. "Comparing Representations with Relational and EER Models." *Communications of the ACM*, Volume 33, Number 2, February 1990, pp. 126-139.
- Batra, D., and Sein, M. K. "Improving Conceptual Database Design Through Feedback." *International Journal of Human-Computer Studies*, Volume 40, 1994, pp. 653-676.
- Chamberlain, D.D., "A Summary of User Experience with the SQL Data Sublanguage." *Proceedings of the International Conference on Data Bases*, July 1980, pp. 181-203.
- Chamberlain, D. D., and Boyce, R. F. "SEQUEL: A Structured English Query Language." *Proceedings of the 1974 ACM SIGMOD Workshop on Data Description, Access, and Control*, Ann Arbor, Michigan, May 1974.
- Chan, H. C.; Wei, K. K.; and Siau, K. L. "An Empirical Study on End-Users' Update Performance for Different Abstraction Levels." *International Journal of Human-Computer Studies*, Volume 41, 1994, pp. 309-328.
- Chan, H. C.; Wei, K. K.; and Siau, K. L. "User-Database Interface: The Effect of Abstraction Levels on Query Performance." *MIS Quarterly*, Volume 17, Number 4, December 1993, pp. 441-464.
- Chen, P. P. "The Entity-Relationship Model: Toward a Unified View of Data." *ACM Transactions on Database Systems*, Volume 1, Number 1, 1976, pp. 166-192.
- Davis, J. S. "Experimental Investigation of the Utility of Data Structure and ER Diagrams in Database Query." *International Journal of Man-Machine Studies*, Volume 32, 1990, pp. 449-459.
- Elmasri, R., and Navathe, S. B. *Fundamentals of Database Systems*. Reading, Massachusetts: Addison Wesley, 1989.
- Gerlach J. H., and Kuo F. Y. "Understanding Human-Computer Interaction for Information Systems Design." *MIS Quarterly*, Volume 15, Number 4, 1991, pp. 526-549
- Greenhalt, D., and Waxman, G. "A Study of Three Database Query Languages." In B. Shneiderman (Editor), *Databases: Improving Usability and Representativeness*. New York: Academic Press, 1978.
- Hawryszkiewicz, I. T. *Relational Database Design*. Englewood Cliffs, New Jersey: Prentice-Hall, 1990.
- Hughes, J. G. *Object-Oriented Databases*. Englewood Cliffs, New Jersey: Prentice-Hall, 1991.
- Hutchins, E. L.; Hollan, J. D.; and Norman, D. A. "Direct Manipulation Interfaces." *Human-Computer Interaction*, Volume 1, 1985, pp. 311-338.
- Jarvenpaa, S. L., and Machesky, J. J. "Data Analysis and Learning: An Experimental Study of Data Modeling Tools." *International Journal of Man-Machine Studies*, Volume 31, 1989, pp. 367-391.
- Jih, W. J.; Bradbard, D. A.; Snyder, C. A.; and Thompson, N. G. A. "The Effects of Relational and Entity-Relationship Data Models on Query Performance of End-Users." *International Journal of Man-Machine Studies*, Volume 31, 1989, pp. 257-267.
- Kim, H. J.; Korth, H. F.; and Silberschatz, A. "PICASSO: A Graphical Query Language." *Software Practice and Experience*, Volume 18, Number 3, March 1988, pp. 169-203.
- Kim, W. "Research Directions in Object-Oriented Database Systems." *Proceedings of Ninth Symposium on Principles of Database Systems*, April 1990, pp. 1-15.
- Markowitz, V. M., and Shoshani, A. "Abbreviated Query Interpretation in Entity-Relationship Oriented Databases." In F. H. Lochovsky (Editor), *Proceedings of the Eighth International Conference on Entity-Relationship Approach*, 1989, pp. 40-58.

- McCromick, J. J. "CIOs reassess Priorities." *Information Week*, December 16, 1991, p.13.
- McDonald, N., and Stonebraker, M. "CUPID: The Friendly Query Language." *Proceedings of the ACM Pacific '75 Conference*, April 1975, pp. 127-131.
- McFadden, F. R., and Hoffer, J. A. *Data Base Management*. Menlo Park, California: The Benjamin/Cummings Publishing Co., 1985.
- Mitchell, D. B., and Hunt, R. R. "How Much 'Effort' Should Be Devoted to Memory?" *Memory and Cognition*, Volume 17, Number 3, 1989, pp. 337-348.
- Navathe, S. B. "Evolution of Data Modeling for Databases." *Communications of the ACM*, Volume 35, Number 9, 1992, pp. 112-123.
- Navon, D. "Resources: A Theoretical Soupstone?" *Psychological Review*, Volume 91, Number 2, 1984, pp. 216-234
- Niederman, F.; Brancheau, J. C.; and Wetherbe, J. C. "Information Systems Management Issues for the 1990s." *MIS Quarterly*, Volume 15, Number 4, 1991, pp. 475-500.
- Reisner, P. "Human Factors Studies of Database Query Languages: A Survey and Assessment." *Computing Surveys*, Volume 13, Number 1, March 1981, pp. 13-31.
- Reisner, P. "The Use of Psychological Experimentation as an Aid to the Development of a Query Language." *IEEE Transactions on Software Engineering*, Volume 3, Number 3, 1977, pp. 218-229.
- Siau, K. L.; Chan, H. C.; and Tan, K. P. "Visual Knowledge Query Language as a Front-end to Relational Systems." *Proceedings of The Fifteenth Annual International Computer Software and Applications Conference*, 1991, pp. 373 - 378
- Siau, K. L.; Chan, H. C.; and Tan, K. P. "Visual Knowledge Query Language." *IEICE Transactions on Information and Systems*, Volume E75-D, Number 5, 1992, pp. 697-703.
- Socket, G. H.; Burns, L. M.; Malhotra, A.; and Whang, K. Y. "GRAQULA: A Graphical Query Language for Entity-Relationship or Relational Databases." *Data and Knowledge Engineering*, Volume 11, Number 2, October 1993, pp. 171-202.
- Suh K. S., and Jenkins A. M. "A Comparison of Linear Keyword and Restricted Natural Language Data Base Interfaces for Novice Users." *Information Systems Research*, Volume 3, Number 3, 1992, pp. 252-272.
- Teorey, T. J. *Database Modeling and Design: The Entity-Relationship Approach*. Morgan Kaufmann Publishers, Inc., 1990.
- Thomas, J., and Gould, J. "A Psychological Study of Query By Example." *Proceedings of the National Computer Conference*. Los Alamitos, California: AFIPS Press, 1975, pp. 439-445.
- Vossen, G. *Data Models, Database Languages and Database Management Systems*. Reading, Massachusetts: Addison-Wesley, 1991.
- Welty, C., and Stemple, D. W. "Human Factors Comparison of a Procedural and Nonprocedural Query Language." *ACM Transactions on Database Systems*, Volume 6, Number 4, December 1981, pp. 626-649.
- Zloof, M. M. "Query-By-Example: A Data Base Language." *IBM System Journal*, Volume 16, Number 4, 1977, pp. 324-343.

Test Questions for VKQL and QBE

1. Show the names and numbers of all employees.
2. Show the departments' names and cities.
3. Show the engineers' numbers, names and professions.
4. Show the names of employees who head any project.
5. Show the names of employees who work in the research department.
6. Show the names of departments which have the same city as the Sales department.
7. Show the names of employees with higher salary than Jack.
8. List the names and professions of engineers who head more than one project.
9. List the names of engineers who do not head any project.
10. List the names and ranks of managers who do not manage any department.

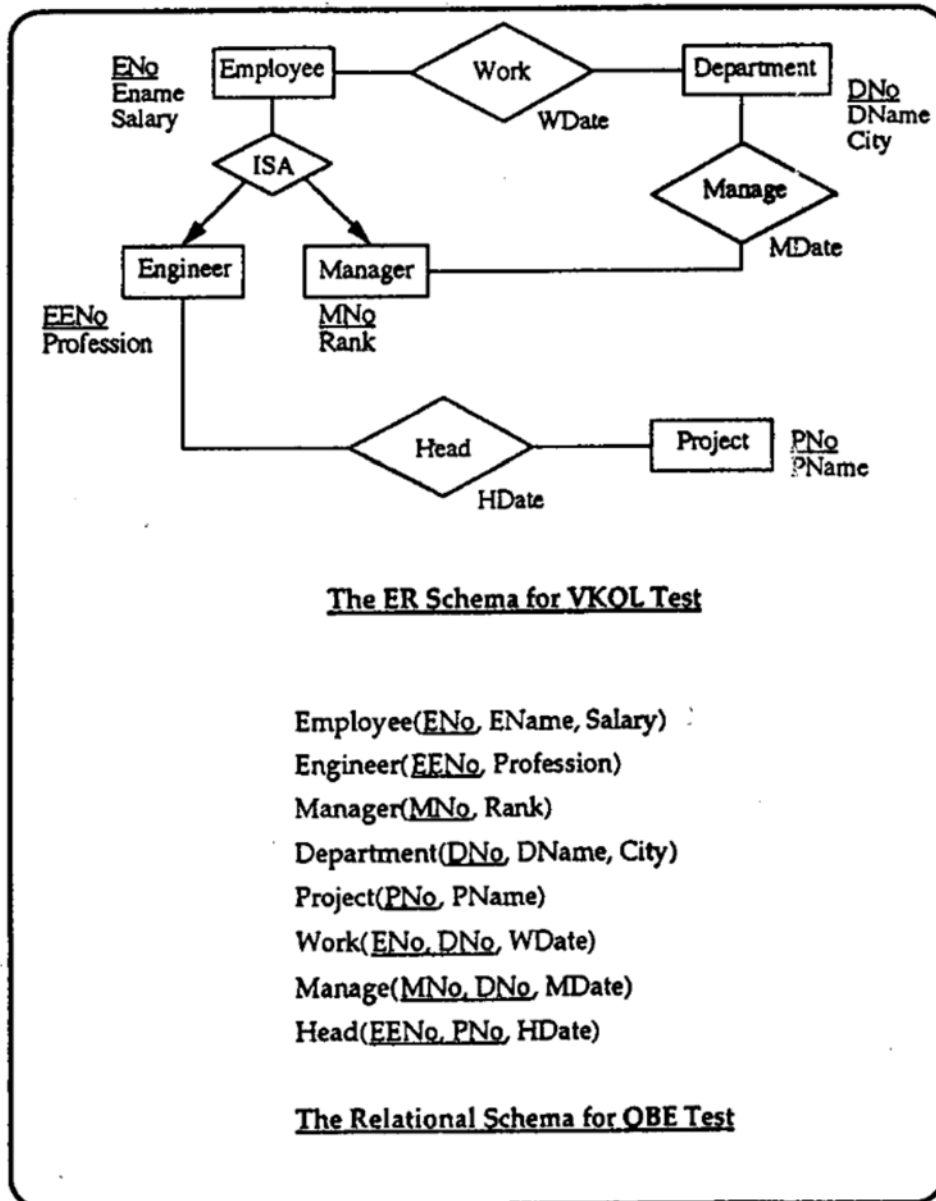


Figure A3. Database Schemas for VKQL and QBE Test