3-2004

Effects of query complexity and learning on novice user query performance with conceptual and logical database interfaces

Keng SIAU
Singapore Management University, klsiau@smu.edu.sg

Hock Chuan CHAN

Kwok Kee WEI

Citation

SIAU, Keng; CHAN, Hock Chuan; and WEI, Kwok Kee. Effects of query complexity and learning on novice user query performance with conceptual and logical database interfaces. (2004). *IEEE Transactions on Systems, Man and Cybernetics Part A: Systems and Humans*. 34, (2), 276-281.
Available at: https://ink.library.smu.edu.sg/sis_research/9484

$\Pi_{\theta c}^{\nu} = \frac{[P(O_c^\theta|\lambda_\nu)]^\eta}{\sum_{k=1,k\neq\theta}^{M}[P(O_c^\theta|\lambda_k)]^\eta}$ the relative similarity measure between λ_ν and λ_θ against all competitors of λ_θ in $\mathbf{O}_c^\theta$.

$$\mathbf{u}_{jk}^{\theta c} = \sum_{t=1}^{T_c^\theta} \gamma_t^{\theta c}(j,k)\mathbf{o}_t^{\theta c}$$

$$\mathbf{R}_{jk}^{\theta c} = \sum_{t=1}^{T_c^\theta} \gamma_t^{\theta c}(j,k)\left(\mathbf{o}_t^{\theta c} - \mathbf{u}_{jk}^{\theta}\right)\left(\mathbf{o}_t^{\theta c} - \mathbf{u}_{jk}^{\theta}\right)'.$$

REFERENCES

[1] R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. 1986 IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Tokyo, Japan, Apr. 1986, pp. 49–52.

[2] H. Juang and L. R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Tech. J.*, vol. 64, no. 2, pp. 391–408, Feb. 1985.

[3] J. T. Chien, C. H. Lee, and H. C. Wang, "Hybrid algorithm for speaker adaptation using MAP transformation and adaptation," *IEEE Signal Processing Lett.*, vol. 4, pp. 167–169, June 1997.

[4] D. Diakoloukas and V. Digalakis, "Maximum-likelihood stochastic-transformation adaptation of hidden Markov models," *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 177–187, Mar. 1999.

[5] V. Digalakis, D. Rtischev, and G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 357–366, Sept. 1995.

[6] V. Digalakis and L. Neumeyer, "Speaker adaptation using combined transformation and Bayesian methods," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 294–300, July 1996.

[7] J. L. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 291–298, Apr. 1994.

[8] Q. H. He, S. Kwong, K. F. Man, and K. S. Tang, "Improved maximum model distance for HMM training," *Pattern Recognit.*, vol. 33, pp. 1749–1758, 2000.

[9] Y. Hao and D. T. Fang, "Speech recognition using speaker adaptation by system parameter transformation," *IEEE Trans. Speech Audio Processing*, vol. 2, no. 1, pp. 63–68, Jan. 1994.

[10] Q. Huo and C. H. Lee, "On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate," *IEEE Trans. Speech Audio Processing*, vol. 5, no. 2, pp. 161–172, Sept. 1997.

[11] B.-H. Juang, W.W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 5, pp. 257–265, May 1997.

[12] N. S. Soo Kim and C. K. Un, "Deleted strategy for MMI-based HMM training," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 299–303, 1998.

[13] T. Kosaka, H. Yamamoto, M. Yamada, and Y.Yasuhiro Komori, "Instantaneous environment adaptation techniques based on fast PMC and PMC-CMS methods," in *1998 IEEE ICASSP*, vol. 2, May 12–15, 1998, pp. 789–792.

[14] S. Kwong, Q. H. He, K. F. Man, and K. S. Tang, "A maximum model distance approach for HMM-based speech recognition," *Pattern Recognit.*, vol. 31, no. 3, pp. 219–229, 1998.

[15] L. R. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 5, pp. 729–734, Sept. 1982.

[16] M. Padmanabhan, L. R. Bahl, D. Nahamoo, and M. A. Picheny, "Speaker clustering and transformation for speaker adaptation in speech recognition systems," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 71–77, Jan. 1998.

[17] M. A. Petrie, "Probabilistic functions of finite state Markov chains," *Ann. Math. Statist.*, vol. 40, no. 1, pp. 97–115, 1969.

[18] A. Sanker and C. H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 190–202, 1996.

[19] S. Kwong, Q. H. He, K. F. Man, and K. S. Tang, "Speaker adaptation technique for HMM model," *Instit. Elect. Eng. Electron. Lett.*, vol. 35, no. 21, pp. 1817–1818, Oct. 14th, 1999.

[20] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process*, vol. 5, pp. 257–265, 1997.

[21] P. M. Baggenstoss, "A modified Baum-Welch algorithm for hidden Markov models with multiple observation speeches," *Int. J. Speech Commun.*, pp. 411–416, May 2000.

[22] A. C. Surendran and C. H. Lee, "Transformation based Bayesian predition for adaption of HMMs," *Int. J. Speech Commun.*, vol. 34, no. 1–2, pp. 159–174, April 2001.

[23] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993, ch. 6.

Effects of Query Complexity and Learning on Novice User Query Performance With Conceptual and Logical Database Interfaces

Keng L. Siau, Hock Chuan Chan, and Kwok Kee Wei

***Abstract*—Users see the database interface as the database system. A good interface enables them to formulate queries better. The semantics communicated through the interface can be classified according to abstraction levels, such as the conceptual and logical levels. With the conceptual interface, interaction is in terms of real-world concepts such as entities, objects and relationships. Current user-database interaction is mainly based on the logical interface, where interaction is in terms of abstract database concepts such as relations and joins. Many researchers argue that end users will perform better with the conceptual interface. This research tested this claim, as well as the effects of query complexity and learning, on the visual query performance of users. The experiment involved three tests: an initial test, a retention test and a relearning test. The results showed that, for complex queries, conceptual interface users achieved higher accuracy, were more confident in their answers, and spent less time on the queries. This is persistent across retention and relearning tests.**

***Index Terms*—Data models, query languages, relational database, user interfaces, visual languages.**

I. INTRODUCTION

Databases form a critical resource for organizations to function properly, to compete, and to survive. Efficient and effective usage of databases requires users to be able to formulate queries accurately and quickly [1], [2]. User productivity is also tied directly to the functionality, ease of learning, and ease of use of the interface [3]. However, query languages for novice users remain poorly explored [4]. This has become a critical issue as the spread of information technology resulted in a need to find interfaces for relatively untrained users [5].

Manuscript received October 19, 2001; revised April 22, 2003 and September 22, 2003. This paper was recommended by Associate Editor Y. Liu.

K. L. Siau is with the Department of Management, College of Business Administration, University of Nebraska, Lincoln, NE 68588-0491 USA (e-mail: ksiau@unl.edu).

H. C. Chan is with the Department of Information Systems, School of Computing, National University of Singapore, Singapore (e-mail: chanhc@comp.nus.edu.sg).

K. K. Wei is with the Department of Information Systems, Faculty of Business, City University of Hong Kong, Hong Kong (e-mail: isweikk@cityu.edu.hk).

Digital Object Identifier 10.1109/TSMCA.2003.820581

A user-database interface is for communication of semantics between users and the system. Based on the abstraction level of the semantics, an interface can be classified into the conceptual, logical, and physical levels. The lowest physical level interface requires the user to know details about physical storage and access structures used in that system. At the logical interface, the queries are specified in terms of abstract structures for data and operations [6]. This is exemplified by the relational database interface [7], where records are stored in tables, and constraints are expressed using primary keys. There are no physical pointers or physical files, and the order of the columns and rows is not important. However, the user must know that by joining relations based on certain fields, it is possible to specify relationships. Lack of this knowledge is a major source of user errors [8]. At the conceptual level, the database is programmed to know the user's world in terms of entities, objects, relationships, and attributes. A data model suitable for this level of interaction is the entity relationship (ER) model [7], [9]. It is stressed that conceptual interfaces "are designed for communication" [6].

The main interface for relational systems is undoubtedly SQL, proposed in [10]. SQL was found to be very difficult to use [11], [12]. Difficulties with query languages such as SQL and QUEL motivated the design of new graphical interfaces to bridge the gap between novice users and database systems [13]. A number of graphical query languages have been proposed, such as PICASSO [13], and QBE [14]. These languages, nevertheless, are still at the logical level.

Many researchers argue that a conceptual interface will be better for users. For example, researchers have noted that database users are often required to understand large, complex database structures [15]. A good objective is to enable users to manage with partial, or no knowledge of the database structure. With the increasing popularity of user computing and empowerment of users, the ability to encode and express queries directly and intuitively is even more important.

This research investigates the claim that the conceptual interface is better than the logical interface for the case of visual query languages. It also considers the effects of query complexity and learning over time on query performance. Section II provides a concise review of the relevant literature. Section III presents the aims of the study and the experiment procedure. Findings and discussions are given in Section IV, and the conclusion is given in Section V.

II. Literature Review

Comparison of the conceptual level versus the logical level using the entity-relationship (ER) model and an ER textual query language at the conceptual level, and the relational model and SQL at the logical level had been studied in an experiment [16]. The results showed that users of the conceptual level exhibited 38% higher accuracy and 16% higher confidence, and took only 35% of the time taken by users of the logical level. Another experiment compared user query performance at the ER and relational interfaces [17], where the users were given either the ER or relational model but answered the queries using the same query language, SQL. The results indicated no significant difference in accuracy. Nevertheless, users of the relational interface took longer time but made less syntactic errors. 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. Together, these two studies demonstrated the importance of the interface [16], [17]. It is not essential to make the two interfaces exactly the same. In fact, the interfaces should differ to the extent that they use their different abstraction levels for best effects. This guided the operationalization of the two visual interfaces in this study.

Another study [18] tested standard documentation (i.e., list of table contents), data structure diagram, and two variations of entity-relationship diagrams to assess their impact on query performance. The results showed that graphical forms of documentation were significantly better than conventional textual documentation, but none of these graphical forms of documentation appeared to be superior to the others. One possible confound in the study was that the subjects had been introduced to relational databases during their MIS course and had been taught the functions of the principal relational operations. As such, the subjects might be more familiar with the data structure diagram since it was a direct representation of the relational tables and less familiar with the entity-relationship diagrams. In addition to these studies, several researchers had also looked at the effect of data models on modeling [19]–[22]. For example, [19] studied user performance in database modeling, showing that user performance using the EER model, as compared to the relational model, was better.

The review points to a need to study the effect of visual conceptual and logical interfaces, so as to extend the results from text-based studies. Furthermore, the effect of query complexity and the effect of learning over a period of time should be explored.

III. Research Variables and Procedure

The research studies the relationship between two independent variables (interfaces and query complexity) and three dependent performance variables (accuracy, time, and confidence). We controlled for other variables such as task and user characteristics. This was in line with the research model proposed in [23].

A. Independent Variables

Interface is set at two levels: conceptual and logical interfaces, which were operationalized as the ER model with an ER query language, visual knowledge query language (VKQL) [4], [27], [28], and the relational model with a relational query language, query by example (QBE) [14]. The construct validity of choosing the ER model for the conceptual level and the relational model for the logical level is well documented in the literature on database design [7], [24]–[26], [29]. [7] states that the conceptual data model is developed using the ER model and the logical data model is typically a relational model. Unlike the study in [17], we made the level distinction clearer by providing a query language for the ER model.

There is no commonly used ER query language. VKQL is a full language designed for the ER model [27], [28]. It has a model definition language and a manipulation language. It includes concepts such as generalization, specialization, categorization and inheritances. Like QBE, VKQL allows for arbitrarily complex queries and is relationally complete. It also supports nested queries and includes statistical functions.

QBE was selected because it is one of the most popular visual relational query languages. QBE represents a visual approach for accessing information in a database through the use of query template [29]. In [30], it was 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 [11] concluded that QBE was "superior to SQL in learning and application ease." Moreover, QBE and VQKL have similar syntax. Both use the table-structure for specifying queries. This avoids extraneous factors such as menu versus graphics or graphics versus command-line.

Query complexity has three levels: simple, medium and complex. In the literature, it is common to see a two-level distinction. For example, the study in [17] defined a simple query as a one-relation query, and a difficult query as one with a join between two relations. Since this study includes queries with joins and other operations such as nesting, a three-level distinction will be more appropriate. Thus, a simple query is defined as one-relation query, a medium query has two relations with one join operation, and a complex query has more relations, more join operations or a nested operation.

B. Dependent Variables

Query performance was assessed by three variables—query accuracy, time taken to formulate the queries, and the subjects' confidence in their queries. These variables are commonly used to assess user query performance [11], [16], [17], [31]. The time taken, measured in seconds, 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 0–5.

C. Subject Characteristics

Subjects were randomly selected from a population of 480 computer science students and randomly assigned to the conceptual and logical groups. (Other students were assigned to other project works.) The conceptual group had 18 subjects and the logical group had 16 subjects. Subjects received course credit based on their query performance. They were told that their performance was based on speed and accuracy, as well as the correlation between accuracy and their self-reported confidence. This would encourage them to report their confidence level honestly rather than to indicate excessive confidence. In most experiments on database query, the number of subjects was quite small. For example, the study in [32] used 26/27/27 for three groups; [16] had 23/24 subjects for two groups; [17] had 26/27 subjects for two groups; and [33] had 12 subjects in one repeated design experiment, and eight subjects in another repeated design experiment. One reason for the small number of subjects is that researchers are interested in big differences that are of practical value. A difference in accuracy score of, say, 0.2 out of five, will not have much practice significance for users, even if it is found to be statistically significant.

The subjects were about 20 years old, with some computing but no database experience. Novice subjects were selected because we wanted to study learning effect over time. The experiment was conducted during the early part of the first semester of their first year in university. Thus subjects would be representative of users who are computer literate, and have minimum training in computing skills. It is noted that other experiments on users also used students as subjects [11], [14], [17], [34], [35].

D. System Characteristics

The characteristics of the system were controlled by having both systems on the Macintosh computer. The systems were essentially a simple interface, customized to display queries and record answers and other data. Care was taken to ensure that the QBE interface appeared and functioned the same way as that proposed in [14]. 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.

E. Research Procedure

1) Training of Subjects: A different training booklet was used for each group. One booklet contained a concise description of the relational data model and illustrations of the QBE query language, using a set of 14 queries. This set of queries covered all the concepts used in the tests. The other booklet was designed to be as similar as possible, and contained a concise description of the ER model and the VKQL query language. A training session of about 45 min was conducted for each group by the same trainer. All 14 examples in the booklet were explained and discussed. The training time allowed all subjects to complete all the examples. Subjects were then given a practice session of about 30 min to familiarize themselves with the software by repeating all the examples.

2) Initial Test (Session 1): After the practice, the subjects were given a ten minutes break before taking the initial test. Ten questions, as shown in the Appendix, 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 started when the subject clicked on the New Query Button displayed on the screen and ended when the subject clicked on the Done Button. After each question, subjects had to enter their confidence in their answer: 0…5 (0: zero confidence, 5: full confidence). The same set and order of questions were given to both 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 ER model and the relational schema are informationally equivalent [36]. In other words, all the information in one is inferable from the other, and vice versa. This is important as it ensures that we are not comparing apples to oranges [36].

3) Retention Test (Session 2): 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. By choosing to reuse the same set of questions, we aimed to test retention effects, whether the subjects could remember what they had learned and practiced. Designing a new set of questions of equal difficulty is difficult as there is no clear method to measure or ensure equivalent difficulty. It should be noted that this retention test was slightly different from the studies in [11], [12], [30], which were closed book tests. In practice, even professional programmers refer to software manuals.

4) Relearning Test (Session 3): After the retention test, the subjects were given a 10 min break. This was followed by a practice session using the computer software. The same set of 14 questions that was used for the practice session prior to the initial test was used for this practice session. Questions by 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.

IV. Experiment Results and Discussion

Query accuracy was determined independently by two graders, who were university professors with an average teaching experience of four years. Each answer was rated from 0–5, based on both the syntactic and semantic accuracy. The numbers assigned by the two graders were very close with at most a two-point difference. The overall correlation coefficient of the two graders is 0.95. A test for reliability assuming that both scores have the same mean, the same true score variance, and the same error variance (using SPSS software and the recommendations in

TABLE I
PERFORMANCE MEASUREMENTS

	QBE means (std dev)			VKQL means (std dev)		
Query Complexity	**simple**	**Medium**	**Complex**	**Simple**	**Medium**	**complex**
Accuracy						
Session 1	4.91 (0.38)	4.36 (0.70)	3.69 (0.76)	5.00 (0.00)	4.58 (0.49)	4.46 (0.45)
Session 2	5.00 (0.00)	4.42 (0.97)	3.94 (0.81)	5.00 (0.00)	4.57 (0.47)	4.50 (0.40)
Session 3	5.00 (0.00)	4.70 (0.53)	4.03 (0.72)	5.00 (0.00)	4.72 (0.35)	4.58 (0.39)
Confidence						
Session 1	4.81 (0.44)	3.81 (0.98)	3.89 (0.67)	4.94 (0.24)	4.61 (0.47)	4.64 (0.28)
Session 2	4.84 (0.40)	4.00 (1.05)	3.86 (0.91)	5.00 (0.00)	4.75 (0.43)	4.44 (0.59)
Session 3	4.97 (0.13)	4.50 (0.82)	4.21 (0.95)	4.94 (0.24)	4.75 (0.60)	4.82 (0.31)
Time (seconds)						
Session 1	39.1 (12.2)	147.7 (67.5)	173.3 (33.4)	28.4 (6.9)	78.4 (24.1)	80.3 (14.3)
Session 2	62.1 (26.5)	122.1 (58.9)	133.3 (25.2)	29.1 (9.4)	82.3 (28.0)	72.1 (16.8)
Session 3	24.6 (5.7)	50.9 (17.8)	83.6 (20.9)	19.7 (5.2)	45.9 (12.0)	46.8 (9.5)

[37]), showed a scale reliability of 0.969. Checks for QBE and VKQL accuracy separately also showed similarly high correlations (0.970 and 0.963, respectively). In the statistical analyses that follow, the average of the accuracy assigned by the two graders was used.

Confidence was measured based on one question, which had also been used by many other researchers in database experiments [16]. For example, the study in [35] used one question "asking the subjects to express overall confidence in the solution they prepared." Table I shows the means and standard deviations (given in brackets) for time, accuracy and confidence. A MANOVA (multivariate analysis) was done with all the dependent and independent variables, for each session alone and for all sessions compared. For the combined session analysis, session (one–three) is used as an independent variable, and its interaction effects with interface and complexity. Significant differences were found for all main and interaction effects in the MANOVA, and separate univariate analyses were done to identify the specific differences. The F and p values in the following sections are from the univariate analyses. The F values have 1 degree of freedom for hypothesis, and 32 degrees of freedom for error, as generated by the SPSS tests.

For the performance measure of accuracy, for each of the three sessions, interface had a main effect, either significant or near significant ($F = 8.3$, $p = 0.007$; $F = 2.5$, $p = 0.123$; and $F = 3.2$, $p = 0.081$ for session 1, 2 and 3 respectively). Complexity had a main effect ($F = 33.6$, $p = 0.001$; $F = 26.9$, $p = 0.001$; and $F = 36.0$, $p = 0.001$ for session one-three, respectively), simple queries were more accurate than medium queries, which in turn were more accurate than complex queries. There was an interaction effect between interface and complexity ($F = 5.7$, $p = 0.023$; $F = 3.5$, $p = 0.072$; and $F = 7.2$, $p = 0.011$ for session one-three, respectively). Closer examination of the interaction effect on the subgroups indicated that accuracies were significantly different between the interfaces only for complex queries.

For the combined session analysis, the significant effects were: interface effect ($F = 5.5$, $p = 0.025$), complexity effect ($F = 59.5$, $p = 0.001$), interaction effect between interface and complexity ($F = 9.8$, $p = 0.004$), and session effect ($F = 5.2$, $p = 0.029$). Each session increased accuracy performance, and session three was significantly better than session one. Interactions of session with interface and/or complexity had no significant effects (p values are all >0.24). More specifically, the F/p values were: 1.43/0.240 for interface * session, 1.28/0.285 for complexity * session, and 0.19/0.663 interface * complexity * session.

Fig. 1 shows the main and interaction effects of interface and complexity on accuracy in session one. Although there was a gap between the two interfaces for simple and medium queries, the gap was not statistically significant. Subsequent sessions reduced the gap for all levels of query complexity, but the gap for complex level remained significant.

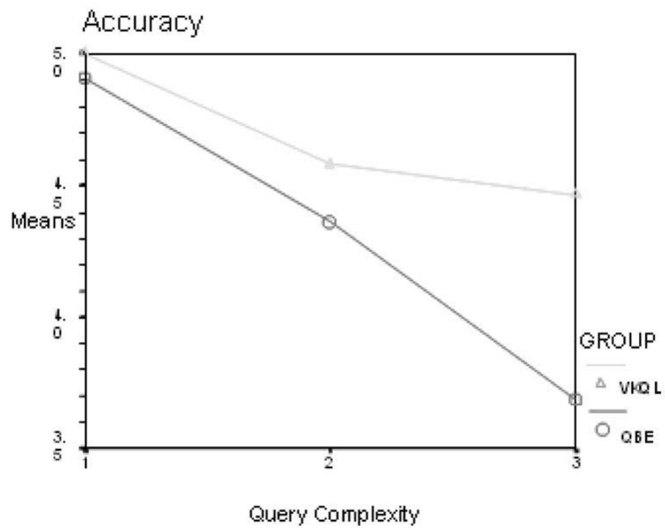

Fig. 1. Main and interaction effects on accuracy.

A main conclusion from these findings is that interface had a persistent effect for complex queries. Extra training and practice (retention and relearning) did not nullify the advantage of a higher level interface.

For confidence, results similar to those for accuracy were found. There were main and interaction effects from interface and complexity. For each of the three sessions, interface had a main effect, either significant or near significant ($F = 13.8$, $p = 0.001$; $F = 7.8$, $p = 0.009$; and $F = 3.0$, $p = 0.094$ for session one-three, respectively). Complexity has a main effect ($F = 25.8$, $p = 0.001$; $F = 22.8$, $p = 0.001$; and $F = 10.3$, $p = 0.001$ for session one-three, respectively). There were significant or near significant interaction effects between interface and complexity ($F = 6.5$, $p = 0.016$, $F = 3.4$, $p = 0.072$; and $F = 5.0$, $p = 0.032$ for session one-three, respectively).

For the combined session analysis, the significant effects on confidence were: interface effect ($F = 8.7$, $p = 0.006$), complexity effect ($F = 30.9$, $p = 0.001$), interaction effect between interface and complexity ($F = 7.0$, $p = 0.013$), and session effect ($F = 12.2$, $p = 0.001$). There were near significant interaction effect between interface and session ($F = 3.6$, $p = 0.067$), and between complexity and session ($F = 3.6$, $p = 0.067$). The interaction among interface, complexity and session had no effect ($F = 1.66$, $p = 0.207$).

Examination of the subgroups reviewed that during the first session, the VKQL interface group had significantly higher confidence for medium and complex queries. This was maintained for session two, indicating that retention did not affect the results. However, session three (with relearning) boosted the confidence of QBE interface group, such that the only significant difference was for complex queries. Consistent with the findings for accuracy, the interface had a persistent effect on confidence for complex queries.

For time, there were main and interaction effects from all factors. Interface had a main effect ($F = 54.8$, $p = 0.001$; $F = 33.7$, $p = 0.001$; and $F = 22.8$, $p = 0.001$ for session one-three, respectively). Complexity has a main effect ($F = 105.0$, $p = 0.001$; $F = 53.5$, $p = 0.001$; and $F = 134.1$, $p = 0.001$ for session 1, 2, 3 respectively). There was an interaction effect between interface and complexity for session one and three ($F = 18.7$, $p = 0.001$ for session 1, $F = 2.7$, $p = 0.111$ for session two, and $F = 24.0$, $p = 0.001$ for session three).

For the combined session analysis, the significant effects on time were: interface ($F = 72.4$, $p = 0.001$), complexity ($F = 197.4$, $p = 0.001$), interaction between interface and complexity ($F = 23.3$, $p = 0.001$), session ($F = 72.8$, $p = 0.001$), interaction between interface and session ($F = 14.0$, $p = 0.001$), interaction between complexity and session ($F = 16.4$, $p = 0.001$), and interaction among interface, complexity and session ($F = 5.6$, $p = 0.025$).

Examination of subgroups reviewed that differences between interfaces for all simple, medium and complex queries and for all sessions were significant, except for medium queries in the last session. Again, interface had a persistent effect on time for complex queries, and also for simple queries.

The experiment was controlled so that both groups had about the same training time. One may ask what would happen if the QBE group had been given slightly more training time. Would they then be comparable or better than the VKQL group? A possible answer is to compare session three of QBE group with session one of VKQL group. By then, the QBE group had more training and practice. Accuracy and confidence differences for complex queries were significant, indicating that more practice and relearning for QBE did not close this gap.

The results of this study confirmed previous results in [16] which showed that users of the conceptual level performed better than users of the logical level in terms of accuracy, confidence, and time. However, the results were different from that in [17] which showed little difference in performance when the ER and relational model were compared. The main reason for the discrepancy was that a special ER query language, VKQL, was used in this study, making the distinction between ER and relational interfaces complete. Although the two database structures (ER model and relational schema) are informationally equivalent, as discussed earlier, they are not necessarily computationally equivalent for the users [36]. Two representations are computationally equivalent if the same information can be extracted from each with about the same amount of computation (e.g., roughly the same time) [36]. With a tailored ER query language (VKQL) for the ER model, the computational inequivalence between the conceptual and logical interfaces becomes more obvious and significant. The results show that a good match of query language and database structure leads to better performance.

This study also shows that differences between groups were mainly for complex queries, and were persistent through subsequent practice and relearning. Simple and medium queries did not have much difference. The effect of more practice and relearning was to raise the performance for both groups, the QBE group more than the VKQL group. However, the gaps for complex queries remained significant. It is possible to project further, and hypothesize that with thorough and long training for QBE users, they will be as good as VKQL users. This would likely be correct; however the point is that one group will need much more training than the other group and that is neither effective nor efficient [38].

V. Conclusion

This research provides empirical evidence supporting the superiority of the conceptual level interface compared to the logical level interface. This research also supplements and complements the study of [16]—the conceptual and logical interfaces are extended from textual to visual, and the effects of learning and retention are added into the experimental design. This "longitudinal" design enables us to investigate the effect over time. The results indicated that users of the conceptual level not only exhibited higher accuracy and higher confidence in all the three tests, they also took less time than users of the logical level in the tests. The results thus provided strong empirical evidence that the conceptual interface is better for users.

What is the implication of the results for practitioners? Our experimental results implies 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. The study also shows the importance of having a query language that can take advantage of the conceptual interface.

With faster and more accurate retrievals, the conceptual interface can empower the users and contribute toward efficient and effective use of organizational databases.

Appendix

Test Questions for VKQL and QBE.

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

References

[1] D. D. Chamberlain, "A summary of user experience with the SQL data sublanguage," in *Proc. Int. Conf. Data Bases*, 1980, pp. 181–203.

[2] R. Ramakrishnan, *Database Management Systems*. New York: McGraw-Hill, 1997.

[3] J. H. Gerlach and F. Y. Kuo, "Understanding human-computer interaction for information systems design," *MIS Quart.*, vol. 15, no. 4, pp. 526–549, 1991.

[4] K. Siau, "A visual object-relationship query language for user-database interaction," *Telemat. Informat.*, vol. 15, no. 1–2, pp. 103–119, 1998.

[5] J. MacGregor, "A comparison of the effects of icons and descriptors in videotex menu retrieval," *Int. J. Man-Mach. Stud.*, vol. 37, pp. 767–777, 1992.

[6] T. Halpin, *Conceptual Schema & Relational Database Design*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[7] S. B. Navathe, "Evoltion of data modeling for databases," *Commun. ACM*, vol. 35, no. 9, pp. 112–123, 1992.

[8] J. B. Smelcer, "User errors in database query composition," *Int. J. Human-Comput. Stud.*, vol. 42, no. 4, pp. 353–381, 1995.

[9] P. P. Chen, "The entity-relationship model: Toward a unified view of data," *ACM Trans. Database Syst.*, vol. 1, no. 1, pp. 9–36, 1976.

[10] D. D. Chamberlain and R. F. Boyce, “SEQUEL: A structured english query language,” in Proc. ACM SIGMOD Workshop Data Description, Access, Control, Ann Arbor, MI, 1974.
[11] D. Greenblatt and J. Waxman, “A study of three database query languages,” in *Databases: Improving Usability and Representativeness*, B. Shneiderman, Ed. New York: Academic, 1978, pp. 76–87.
[12] C. Welty and D. W. Stemple, “Human factors comparison of a procedural and nonprocedural query language,” *ACM Trans. Database Syst.*, vol. 6, no. 4, pp. 626–649, 1981.
[13] H. J. Kim, H. F. Korth, and A. Silberschatz, “PICASSO: A graphical query language,” *Softw.—Pract. Exp.*, vol. 18, no. 3, pp. 169–203, 1988.
[14] M. M. Zloof, “Query-By-Example: A data base language,” *IBM Syst. J.*, vol. 16, no. 4, pp. 324–343, 1977.
[15] V. M. Markowitz and A. Shoshani, “Abbreviated query interpretation in entity-relationship oriented databases,” in *Proc. 8th Int. Conf. Entity-Relationship Approach*, F.H Lochovsky, Ed., 1989, pp. 40–58.
[16] H. C. Chan, K. K. Wei, and K. L. Siau, “User-Database interface: The effect of abstraction levels on query performance,” *MIS Quart.*, vol. 17, no. 4, pp. 441–464, 1993.
[17] W. J. Jih, D. A. Bradbard, C. A. Snyder, and N. G. A. Thompson, “The effects of relational and entity-relationship data models on query performance of end-users,” *Int. J. Man-Mach. Stud.*, vol. 31, pp. 257–267, 1989.
[18] J. S. Davis, “Experimental investigation of the utility of data structure and ER diagrams in database query,” *Int. J. Man-Mach. Stud.*, vol. 32, pp. 449–459, 1990.
[19] D. Batra, J. A. Hoffer, and R. P. Bostrom, “Comparing representations with relational and EER models,” *Commun. ACM*, vol. 33, no. 2, pp. 126–139, 1990.
[20] S. L. Jarvenpaa and J. J. Machesky, “Data analysis and learning: an experimental study of data modeling tools,” *Int. J. Man-Mach. Stud.*, vol. 31, pp. 367–391, 1989.
[21] D. Batra, “A framework for studying human error behavior in conceptual database modeling,” *Inform. Manage.*, vol. 25, pp. 121–131, 1993.
[22] D. Batra and M. K. Sein, “Improving conceptual database design through feedback,” *Int. J. Human-Comput. Stud.*, vol. 40, pp. 653–676, 1994.
[23] P. Reisner, “Human factors studies of database query languages, a survey and assessment,” *Comput. Surv.*, vol. 13, no. 1, pp. 13–31, 1981.
[24] H.C. Chan, K.K. Wei, and K. Siau, “The effect of a database feedback system on user performance,” *Behav. Inf. Technol.*, vol. 14, no. 3, pp. 152–162, May–June 1995.
[25] ——, “A system for query comprehension,” *Inf. Softw. Technol.*, vol. 39, pp. 141–148, 1997.
[26] H. Chan, K. Siau, and K. Wei, “The effect of data model, system and task characteristics on user query performance—An empirical study,” *DATA BASE Advances Inform. Syst.*, vol. 29, no. 1, pp. 31–49, Winter 1998.
[27] K. L. Siau, K. P. Tan, and H. C. Chan, “A CASE tool for conceptual database design,” *Inform. Softw. Technol.*, vol. 34, no. 12, pp. 779–786, December 1992.
[28] K. L. Siau, H. C. Chan, and K. P. Tan, “Visual knowledge query language,” *IEICE Trans. Inform. Syst.*, vol. E75-D, no. 5, pp. 697–703, 1992.
[29] T. Connolly, C. Begg, and A. Strachan, *Data Systems—A Practical Approach to Design, Implementation and Management*. Reading, MA: Addison-Wesley, 1996.
[30] J. Thomas and J. Gould, “A psychological study of query by example,” in *Proc. National Computer Conf.*, 1975, pp. 439–445.
[31] H. C. Chan, K. K. Wei, and K. L. Siau, “An empirical study on end-users’ update performance for different abstraction levels,” *Int. J. Human-Comput. Studies*, vol. 41, pp. 309–328, 1994.
[32] A. F. Borthick, P. L. Bowen, S. T. Liew, and F. H. Rohde, “The effects of normalization on end-user query errors: An experimental evaluation,” *Int. J. Accounting Inform. Syst.*, vol. 2, pp. 195–221, 2001.
[33] J. M. Boyle, K. F. Bury, and R. J. Evey, “Two studies evaluating learning and use of QBE and SQL,” in *Proc. Human Factors Society 27th Annu. Meeting*, 1982, pp. 663–667.
[34] K. S. Suh and A. M. Jenkins, “A comparison of linear keyword and restricted natural language data base interfaces for novice users,” *Inf. Syst. Res.*, vol. 3, no. 3, pp. 252–272, 1992.
[35] C. Liao and P. C. Palvia, “The impact of data models and task complexity on end-user performance: An experimental investigation,” *Int. J. Human-Comput. Studies*, vol. 52, pp. 831–845, 2000.
[36] K. Siau, “Informational and computational equivalence in comparing information modeling methods,” *J. Database Manage.*, vol. 15, no. 1, pp. 73–86, 2004.
[37] D. George and P. Mallery, *SPSS for Windows Step by Step. A Simple Guide and Reference*. Boston, MA: Allyn & Bacon, 2001.
[38] K. Siau, “Information modeling and method engineering: A psychological perspective,” *J. Database Manage.*, vol. 10, no. 4, pp. 44–50, 1999.

Internet Scheduling Environment With Market-Driven Agents

Benjamin P.-C. Yen and Owen Q. Wu

***Abstract*—This paper describes a new generation scheduling paradigm, the Internet scheduling environment. It is formed by a group of Internet scheduling agents which share computational resources to solve scheduling problems in a distributed and collaborative manner. We propose a migration scheme to transform existing standalone scheduling systems to Internet scheduling agents that can communicate with each other and solve problems beyond individual capabilities. To coordinate computational resource collaboration among agents, we introduce the market-based control mechanism in which self-interested agents initiate or participate in auctions to sell or buy scheduling problems. Efficient allocation of computational resources is achieved through the auctions. This paper also describes a prototype Internet scheduling environment named LekiNET, which is migrated from LEKIN®, a flexible job shop scheduling system. The experiments on the LekiNET testbed demonstrate that the agent-based market-driven Internet scheduling environment is feasible and advantageous to future scheduling research and development.**

***Index Terms*—Agent, distributed resource collaboration, Internet scheduling environment, market-based control.**

I. Introduction

A. Needs for Distributed Resource Collaboration

Scheduling problems abound in manufacturing factories, transportation systems, hospitals, publishing houses, and so on. As demands on businesses become greater, companies are faced with increasingly complex tasks.

For example, in British Aerospace’s largest factory in Broughton, U.K., there are around 2000 staff producing 180 sets of Airbus wings and 40 sets of Hawker 800 fuselages and wings per year. They are constantly seeking improvement in production schedules that could bring substantial savings in reducing work-in-process inventories and late deliveries.

For another example, S&A Food’s Derby center of operations employs some 1000 staff to produce 1.1 million meals—or 650 tons of prepared food—per week. The schedules must be dynamically updated in response to the instant change of demands and priorities. The schedulers must also be able to answer what-if questions to prepare ahead for

Manuscript received January 1, 2000; revised February 13, 2003 and October 12, 2003. This work was supported in part by Hong Kong Government RGC under Grant 6076/00E. This paper was recommended by Associate Editor Y. Narahari.

B. P.-C. Yen is with the Faculty of Business and Economics, University of Hong Kong, Hong Kong.
O. Q. Wu is with the Sauder School of Business, University of British Columbia, Vancouver, BC V6T 1Z2 Canada.

Digital Object Identifier 10.1109/TSMCA.2003.822273