December 2004

Analysis of Competing Data Structures: Does Ontological Clarity Produce Better End-User Query Performance?

Paul Bowen
The University of Queensland

Robert O'Farrell
The University of Queensland

Fiona Rohde
The University of Queensland

Analysis of Competing Data Structures: Does Ontological Clarity Produce Better End-User Query Performance?

Paul L. Bowen, Robert A. O'Farrell, and Fiona H. Rohde
UQ Business School
The University of Queensland
Brisbane, Queensland, Australia
p.bowen@business.uq.edu.au   f.rohde@business.uq.edu.au

Abstract

End users respond to the information requests of stakeholders by using SQL-based query tools to retrieve information from their organizations' data stores. The structure of these data stores impacts end users' performance (e.g., the accuracy of their queries).

Traditionally, logical data models with optional properties (referred to hereafter as more-traditional models) are used. Ontologically clearer conceptual models, however, have been shown to facilitate better under-standing of real-world application domains. The question now arises as to whether ontologically clearer (i.e., more precise) implementation representations also improve end-user query performance. This paper reports the results of an experiment that investigates the effect on query performance of more-traditional logical models compared to ontologically clearer logical models. End users of the ontologically clearer implementation made fewer semantic errors and were more confident in the accuracy of their queries. No statistical difference in time was observed between users of the two types of logical models.

Keywords: Ontology, information retrieval, end user performance

Introduction

End users increasingly use SQL-based query tools to provide timely responses to the information requests of stakeholders. Semantically accurate queries produce higher quality information for decision makers and, hence, lead to better decisions. Factors that affect the quality of SQL queries include data structure quality and the extent to which the application domain is understood by the information retrieval agent. Ontological research into conceptual data models indicates that people perform problem-solving tasks better using conceptual entity-relationship diagrams that evidence higher degrees of ontological clarity (Bodart et al. 2001; Burton-Jones et al. 1998; Gemino 1998; Wand and Weber 1999).

Conceptual data models are used as the basis for the design tasks that are performed when building or modifying the data schemas (Hoffer et al. 2004). This paper extends the research into the effects of ontological clarity of data models by investigating whether the positive results of greater ontological clarity for problem-solving tasks also hold for information retrieval tasks. That is, this research examines to what extent ontological clarity impacts query performance. The specific part of the Bunge-Wand-Weber ontology examined in this research is Weber's (1997) assertion that Bunge's (1977) ontology implies that optional properties should not be used in semantic modeling. Instead, subtypes with mandatory properties should replace each optional property (e.g., students are divided into postgraduate students and undergraduate students where only postgraduate students have the property of degree held).

The data models developed at the conceptual level have important consequences for the construction and use of information systems. For example, some CASE tools automatically transform conceptual models into logical (i.e., implementation) data

structures. If ontologically clearer conceptual models are directly transformed into implementation data models, the increased number of tables may cause cognitive overload for end users and their query performance may decline. Conversely, the enhanced precision associated with ontologically clearer implementation models may enhance query performance. The primary purpose of this research is to determine, for application domains of approximately the same size of prior research into conceptual models, whether the greater ontological clarity produces better information retrieval performance. We also analyzed specific characteristics associated with the two data structure representations to determine the specific aspects of the queries that caused more errors and problems for query developers.

Background

The Bunge-Wand-Weber Model

Bunge's (1977) theory of ontology, applied to information systems by Wand and Weber (1993), has gained widespread attention within both the Information Systems and Software Engineering conceptual modeling domains. Much of the attention has been focused on the application of the Bunge-Wand-Weber model to varied application domains (e.g., business and conceptual modeling). Early research into the adoption of Bunge's ontology produced a set of ontological constructs (Wand and Weber 1993; Weber and Zhang 1996), later denoted as the Bunge-Wand-Weber (BWW) model. This research provided guidelines that researchers are able to use to evaluate modeling grammars for ontological completeness (Shanks et al. 2003; Wand and Wang 1996).

An application of the BWW model was empirically tested by Bodart et al. (2001) using three experiments to examine whether optional properties should be used in conceptual modeling. The results indicate that for recall and comprehension, participants using diagrams with optional properties outperformed participants using diagrams with mandatory properties. For problem-solving tasks, however, participants using diagrams with mandatory properties outperformed participants using diagrams with optional properties. Formulating queries to satisfy information requests is a problem-solving task.

Creating Ontological Clarity Using the BWW Model

Optional Properties

Optionality is an important feature of many conceptual modeling grammars (Bodart et al. 2001). Wand and Weber (1999) and Weber (1997), however, argue *things* are perceived via the properties they possess, not via the properties they do not possess. To clearly convey the ontological meaning of the constructs in conceptual models, Wand and Weber state that optionality should be avoided. Within data modeling, two *optional* situations can arise. The first occurs when a thing may or may not possess an attribute (i.e., an optional attribute). The second occurs when a thing may or may not participate in a relationship with another thing (i.e., an optional relationship). These two situations can be avoided, and thus ontological clarity improved, through the use of the "IS-A" construct to represent subclasses.

To derive an ontologically clearer data model by removing optional properties, the data modeler creates a subclass that represents those things that possess the property and another subclass that represents those things that do not possess the property. Consider the following example in relation to optional attributes: a customer places a sales order. At the time of initially creating the sales order, the name of the employee who will fill the order is not known (i.e., picked_by_employee is an optional attribute at this point in the sales and delivery process; see Figure 1). The ontologically clearer solution is to model this optional attribute using two subtypes: one for orders that have been picked and are now awaiting shipment and orders that have not been picked (see Figure 2).

To derive an ontologically clearer data model by removing optional relationships, the data modeler creates one subclass to represent those things that possess the properties of the thing being modeled and another subclass representing those things that represent the optional aspect. One such example relates to inventory items that have never appeared on a sales order and inventory items that have appeared on at least one sales order. A possible solution is to represent these two states by an optional relationship between two entities (see Figure 3). The ontologically clearer solution is to create one subclass of items containing those items that have had no sales and a second subclass for those items that have sales. Only those inventory items that have sales can participate in the relationship with sales order items (see Figure 4).
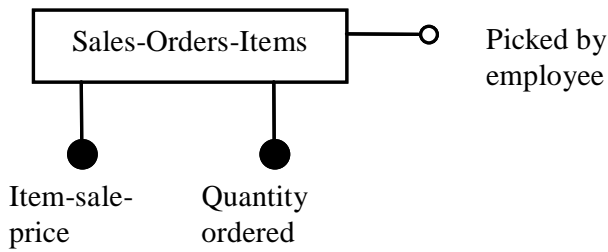
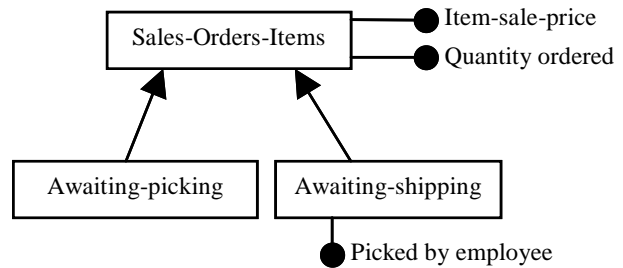Figure 1. Optional Property in Ontologically
Unclear Model

Figure 2. Optional Property in Ontologically
Clearer Model

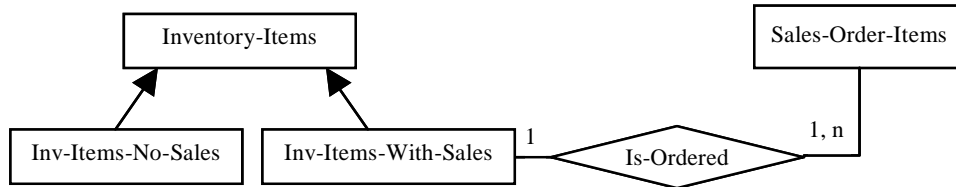Figure 3. Optional Relationships in Ontologically Unclear Model

Figure 4 Optional Relationships in Ontologically Clearer Model

Associative Entities

Weber suggests that using relationships with attributes will undermine the ontological clarity of conceptual models. He suggests replacing relationships with attributes by mutual properties to improve the ontological clarity of entity-relationship scripts. These mutual properties belong to the entities involved in the association. For example, the two real-world things *inventory items* and *sales orders* participate in a many-to-many association with each other. Details about *order items*, such as the quantity of a particular item ordered on a particular sales order needs to be recorded.

In practice, entity-relationship diagrams typically represent these extra details as attributes of the relationship *order* between two entities *inventory items* and *sales orders* (Chen 1976) (see Figure 5). According to Bunge's ontology, properties do not have properties, thus representing a relationship (i.e., a mutual property) with attributes in conceptual models is not ontologically clear. Wand et al. (1999) and Weber suggest that one solution is to think about *sales order items* as a thing that has certain properties. The three things then participate in relationships with each other (see Figure 6).

End-User Query Performance

A large body of research exists that investigates factors that affect end-user query performance (Axelsen et al. 2001; Borthick et al. 2001a; Borthick et al. 2001b). Query performance has been shown to be affected by end users' ability to understand the application domain (Chan et al. 2004; Jih et al. 1989; Rho and March 1997) and by end users' ability to translate their understanding of the query domain correctly into a query language (Chan et al. 1999; Chan et al. 1993; Suh and Jenkins 1992).
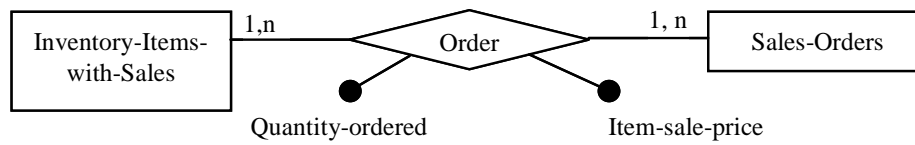
Figure 5. Relationships with Attributes in Ontologically Unclear Model
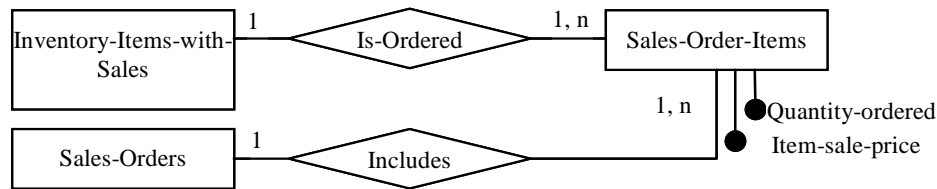

Figure 6. Relationships with Attributes Represented as Associative Entity in Ontologically Clearer Model

Many meanings of complexity can be found within the context of information processing. Researchers, however, are consistent in asserting a positive relationship between complexity and errors in task performance (i.e., as task complexity increases, performance decreases) (Borthick et al. 2001a, Borthick 2001b; Campbell 1988; Chan et al. 1999; Jih et al. 1989; March and Simon 1958; Rho and March; 1997). Furthermore, as task complexity increases, end users' confidence in their solution to the problem is likely to decrease (Campbell 1988; Gardner and Serra 1997). Problems arising from more complex queries include lower effectiveness, efficiency, and confidence.

Effects of Ontological Clarity on Performance

The data structure (or data model) portrays the logical organization of the database to users. To develop a query, end users map the constructs in the information request to the attributes in the database. More-traditional data structures and ontologically clearer data structures yield different numbers of entities and relationships. To remove an optional property requires the use of subtyping. Unfortunately, the use of subtyping results in the number of entities in the data models increasing markedly. To remove an associative entity also implies the creation of additional roles. When formulating queries for these ontologically clearer data structures, end users often need to reassemble the fragmented data. This data reassembly results in queries that contain more terms and more complicated logic.

More-traditional data structures contain fewer entities and thus require less reassembly of data. More-traditional data structures, however, contain optional properties. To obtain the correct data from more-traditional data structures often requires the appropriate use of IS NULL or IS NOT NULL in the WHERE CONDITION clause. The correct formulation of such WHERE CONDITIONS requires more cognitive effort than the corresponding queries of ontologically clearer data structures where the IS NULL or IS NOT NULL statements are not required. Thus, while the more-traditional data structure is smaller and requires less data reassembly, an additional type of complexity is imposed through the use of the optional properties. Because of the positive and negative aspects of both alternatives, the hypothesis is stated in the null form.

> **Hypothesis 1:** End users querying ontologically clearer data structures do not make a different amount of semantic errors than end users querying more-traditional data structures.

More complex tasks require more exacting logic. The increased complexity of both types of data structures and the resulting increased probability of making query formulation errors leads to an increase in the time taken to compose queries. While equally proficient query developers make different numbers of attempts to complete queries, the length of time taken is important from a business perspective . More-traditional and ontologically clearer data structures present end users with different challenges and, therefore, both data structures affect end users' time taken to compose queries. The hypothesis, stated in the null, is

> **Hypothesis 2:** End users querying ontologically clearer data structures do not take a different amount of time to construct queries than those querying more-traditional data structures.

For effective decision making, end users must first extract the correct information from the database. After extracting information from the database, end users' confidence in the correctness of their queries affects their willingness to rely on the information generated by the queries. That is, if they are not confident that their query formulations are correct, then they may not fully rely on the information obtained. More complex tasks require more exacting logic. The more exacting logic makes end users more aware of the possibility that errors may occur and thus reduces their confidence in the correctness of their queries. More-traditional and ontologically clearer data structures present end users with different challenges and, therefore, both data structures affect end users' confidence in the accuracy of their queries. The hypothesis, stated in the null form, is

> **Hypothesis 3:** End users querying ontologically clearer data structures are no more or less confident than those querying more-traditional data structures.

Research Method

Research Design, Participants, and Data Collection

In a laboratory experiment, participants composed and executed queries in SQL for one of two Oracle databases. Both data structures satisfied third-normal form. The first database was a more-traditional data structure. The second equivalent data structure differed only in that optional properties and relationships were removed and thus it was ontologically clearer (Appendix A). Forty-six advanced undergraduate and Master's level Commerce students participated in the experiment. All participants were familiar with general computing concepts and activities and, prior to the experiment, had received training in developing SQL queries. All participants received a set of instructions containing the scenario and the details of tasks to be performed (Appendix B contains the information requests). To control for experience and education effects, participants were assigned to one of two groups according to their GPA. The person with the highest GPA was ranked 1, the next ranked 2, etc. Participants were assigned to groups according to their rank (i.e., 1 to group A, 2 to group B, 3 to group B, 4 to group A, etc.). This method of assignment was intended to make the two groups as equivalent as possible. The groups were then randomly assigned to a treatment.

The participants had two hours to construct, as accurately as possible, appropriate queries for as many of the 14 information requests (Appendix B) as they could. Participants received 7 percent course credit for participating. Participants were informed that they would be marked on the accuracy of each of the queries they completed. Because the correct query formulations were generally increasing in complexity, participants were encouraged to do their best on each query before moving to the next information request. Each information request had two correct formulations: one for the more-traditional data structure and one for the ontologically clearer data structure. The experiment was designed so that for the even-numbered information requests, the correct query of the more-traditional data structure was shorter than the correct query of the ontologically clearer data structure. For the odd-numbered information requests the reverse occurred. By adopting this design, neither data structure was biased relative to complexity.

Participants used a UNIX shell script that recorded their entire session. Each participant was presented the information requests in the same order. After each query attempt was executed, the system displayed the SQL result (i.e., either the rows returned by the query or a syntax error message). Participants could revise their queries as many times as they wished. When they indicated that they were satisfied with the result they obtained for a particular request, participants were prompted to specify their confidence that the query results were correct. After indicating their confidence levels, participants proceeded to the next information request. Once an information request had been deemed completed by the participants, they could not return to it. The use of feedback during the experiment allowed the querying experience to better approximate that of real life. Similarly, the examination of only the last query is also important. In a business context, query developers are allowed as many attempts as they wish to obtain the desired information, however, they are usually working toward a time deadline. The results of only the final query are presented to the person making the information request.

Operationalizing the Variables

Dependent Variables

Two experienced coders independently determined the minimum number of changes, if any, required to make each query from each participant semantically correct. After the two individuals independently performed this task, they cross-checked their error

coding sheets (Appendix C) for correctness and consistency and resolved any differences.[1] The total number of each type of semantic error was determined by counting the number of each type of semantic error in the last attempt for each information request. These errors were then summed to derive the total number of semantic errors. This error count is the dependent variable for hypothesis 1.

The dependent variable for hypothesis 2 is the total time taken to compose the query for the information requests. The value for this variable was determined by examining the log files.

The dependent variable for hypothesis 3 is each participant's self-assessed confidence level for each query. Recall that participants were allowed to attempt a particular information request as many times as they wished. When participants indicated that they were satisfied that they had completed the particular request (i.e., did not wish to make another attempt), participants were prompted to specify their confidence that the developed query produced results that satisfied the information request. Participants entered this measure on the following scale: 86 to 100 percent, 71 to 85 percent, 56 to 70 percent, 41 to 55 percent, 26 to 40 percent, 11 to 25 percent, and 0 to 10 percent. The values were transformed to a seven point scale as follows: ratings of 86 to 100 percent transformed to 7, ratings of 71 to 85 percent to 6, ratings of 56 to 70 percent to 5, ratings of 41 to 55 percent to 4, rating of 26 to 40 percent to 3, rating of 11 to 25 percent to 2, and ratings of 0 to 10 percent to 1.

Independent Variables

The independent variable was the treatment group. Group was a categorical variable with the values of more-traditional and ontologically clearer. The information requests were generally of increasing complexity[2] and, thus, information request (query) number was a categorical variable with values from 1 to 14. Grade point average (GPA) was also used as a covariate.

Results

Summary Performance

Table 1 summarizes the participants' characteristics and performance by data structure. Table 1, Panel A, indicates that, in absolute terms, participants generated fewer errors, took less time, and were more confident when querying the ontologically clearer data structure. Table 1, Panel B, indicates that, in absolute terms, participants generated more SELECT, WHERE CONDITION, GROUP BY, and HAVING errors when querying the more-traditional data structure.

The Effect of Ontological Clarity on Total Semantic Errors

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that the number of semantic errors was significantly associated with the level of ontological clarity ($F_{12,408}$= 2.27, p = 0.0085, two-tail test) (Table 2, Panel A). The means results (Table 1, Panel A) confirm that the end users querying the ontologically clearer data structure made significantly fewer semantic errors than end users querying the more-traditional data structure. Additional *post hoc* tests confirmed that for those queries where the ontologically clearer group were favored, the ontologically clearer participants made significantly fewer errors. On the other hand, for those queries where the more-traditional group were favored, neither set of participants made significantly fewer errors than the other.

---

[1]When the two coders compared their solutions, the possible outcomes were initial total agreement, one coder being deemed correct, or both coders changing their solution. Given the criteria of making the minimum number changes to reach a semantically correct solution, after re-examining each query and each coder's solution, the coders were always able to reach agreement on the number of errors, if any, in each participant's query. To give an indication of the process, the two coders recorded the outcomes for 80 queries. The coders were in initial total agreement on 85 percent of the queries. On the remaining 12 queries, the more experienced coder was deemed correct on 6 of the queries, the less experienced coder was deemed correct on 4 of the queries, and both coders changed their error counts on 2 of the queries.

[2]The order of the information requests took into account the challenges encountered by participants when composing a query. A query containing a sub-query or outer join, for example, is likely to be shorter than a query joining multiple tables. Participants, however, often find the shorter query more difficult and thus more challenging.

Table 1. Participant Characteristics and Performance

Panel A: Participant Characteristics and Overall Performance		
	Data Structure	
	More-Traditional	**Ontologically Clearer**
Grade Point Average (7-point scale, 7 highest)		
Mean	4.8928	4.7763
Standard deviation	0.8579	0.8544
Gender		
Number of males	16	14
Number of females	7	9
Semantic Errors per Request Attempted		
Mean	4.3285	3.5347
Standard deviation	8.3982	7.7947
Confidence Level per Request Attempted		
Mean	5.5169	6.1337
Standard deviation	1.5164	1.1957
Time per Information Request		
Mean	11.1292	10.5756
Standard deviation	8.8682	8.8097
Panel B: Performance for Different Error Types		
	Data Structure	
	More-Traditional	**Ontologically Clearer**
Number of Select Errors		
Mean per attempted query	0.8792	0.6832
Standard deviation	2.1833	2.1737
Number of From Errors		
Mean per attempted query	0.3719	0.4109
Standard deviation	1.1242	1.2113
Number of Where Join Errors		
Mean per attempted query	1.0435	1.5247
Standard deviation	2.9350	3.6148
Number of Where Condition Errors		
Mean per attempted query	1.2077	0.3911
Standard deviation	2.3608	1.6117
Number of Group By Errors		
Mean per attempted query	0.3091	0.1584
Standard deviation	1.4007	0.8779
Number of Having Errors		
Mean per attempted query	0.3913	0.1832
Standard deviation	2.1870	1.3349

The Effect of Ontological Clarity on Time to Compose Queries

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that the time taken to compose queries was not significantly associated with the level of ontological clarity ($F_{12,408} = 0.96$, $p = 0.4830$, two-tail test) (Table 2, Panel B).

Table 2. Main Model and Effects of Data Structure on Accuracy, Efficiency, and Confidence

Panel A. Effects of Data Structure on End-User Accuracy					
Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.3615	24	403.72	9.06	0.0001
Error		384	44.56		
Query		11	764.55	17.16	0.0001
Group (Query)		12	101.32	2.27	0.0085
GPA		1	245.95	5.52	0.0193
Panel B. Effects of Data Structure on End-User Efficiency					
Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.3981	24	528.12	10.58	0.0001
Error		384	49.89		
Query		11	1078.96	21.62	0.0001
Group (Query)		12	48.29	0.96	0.4830
GPA		1	564.14	11.31	0.0008
Panel C. Effects of Data Structure on End-User Confidence					
Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.2913	24	9.71	6.58	0.0001
Error		384	1.47		
Query		11	15.55	10.54	0.0001
Group (Query)		12	4.54	3.08	0.0004
GPA		1	15.70	10.64	0.0012

The Effect of Ontological Clarity on Confidence

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that end users' confidence was significantly associated with the level of ontological clarity ($F_{12,408} = 3.08$, p = 0.0004, two-tail test) (Table 2, Panel C). The means results (Table 1, Panel A) confirm that end users querying the ontologically clearer data structure were significantly more confident in the accuracy of their queries than end users querying the more-traditional data structure.

Covariates

GPA and query complexity (proxied by query number) were significantly associated with each of the dependent measures (Table 2, Panels A, B, and C). Participants with higher GPAs performed the query tasks significantly more accurately, took significantly less time, and were significantly more confident in the correctness of their queries. More complex queries contained significantly more errors, took significantly more time, and left participants significantly less confident in the accuracy of their queries.

The Effect of Ontological Clarity on Different Types of Semantic Errors

Hypotheses 1 to 3 test the net effects of the different challenges associated with the two types of data structures. Different challenges affect different sections of the SQL queries. End users querying the ontologically clearer data structure are required to perform more joins on more fragmented schemas. These join operations require them to nominate the correct tables and then perform join restrictions. Thus, the FROM and WHERE JOIN clauses are most likely to be affected. The more-traditional end users are required to consider the effects of optional properties. Optional properties require them to include NULLS and NOT NULLS in the WHERE CONDITION clauses of their queries. Including optional relationships requires the end users of the more-

traditional data structure to include subqueries more often in their queries. Thus, more-traditional end users are more likely to make more SELECT errors because of the necessity of using subqueries. The remainder of the query (i.e., the GROUP BY, HAVING, and ORDER BY clauses) are less affected.

The Effect of Ontological Clarity on SELECT Errors

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that the number of SELECT errors was significantly associated with the level of ontological clarity ($F_{12,408}$ = 2.85, p = 0.0009, two-tail test) (Table 3, Panel A). The means results (Table 1, Panel B) confirm that the end users querying the ontologically clearer data structure made significantly fewer SELECT errors than end users querying the more-traditional data structure. Querying the more-traditional data structure required the use of more subqueries and thus the result is consistent with expectations.

The Effect of Ontological Clarity on FROM Errors

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that the number of FROM errors was significantly associated with the level of ontological clarity ($F_{12,408}$ = 2.00, p = 0.0230, two-tail test) (Table 3, Panel B). The means results (Table 1, Panel B) confirm that end users querying the ontologically clearer data structure made significantly more FROM errors than end users querying the more-traditional data structure. This finding indicates that participants had more difficultly selecting the appropriate tables when using the ontologically clearer data structure. The ontologically clearer data structure contains more entities (thus tables) and the result is consistent with expectations.

The Effect of Ontological Clarity on WHERE JOIN Errors

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that the number of WHERE JOIN errors was significantly associated with the level of ontological clarity ($F_{12,408}$ = 4.88, p = 0.0001, two-tail test) (Table 3, Panel C). The means results (Table 1, Panel B) confirm that end users querying the ontologically clearer data structure made significantly more WHERE JOIN errors than end users querying the more-traditional data structure. As the data in the ontologically clearer data structure is more fragmented, the queries often require additional join conditions in the WHERE clause. The increased requirement of joins is consistent with a corresponding increase in the likelihood that the query developers make more WHERE JOIN errors using the ontologically clearer data structure.

The Effect of Ontological Clarity on WHERE CONDITION Errors

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that the number of WHERE CONDITION errors was significantly associated with the level of ontological clarity ($F_{12,408}$ = 3.23, p = 0.0002, two-tail test) (Table 3, Panel D). The means results (Table 1, Panel B) confirm that end users querying the ontologically clearer data structure made significantly fewer WHERE CONDITION errors than end users querying the more-traditional data structure. The inclusion of optional properties in the more-traditional model and their subsequent removal in the ontologically clearer model is a fundamental difference between the two data structures. This difference requires more extensive use of IS (NOT) NULL in the queries using the more-traditional data structure, whereas additional joins are required for queries using the ontologically clearer data structure. That the users of the more-traditional data structure make more WHERE CONDITION errors is consistent with expectations.

The Effect of Ontological Clarity on GROUP BY and HAVING Errors

Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that the number of GROUP BY errors was not significantly associated with the level of ontological clarity ($F_{12,408}$ = 1.44, p = 0.1466, two-tail test). Comparing the more-traditional with the ontologically clearer group, nested ANCOVA results indicate that end users querying the ontologically clearer data structure made significantly fewer HAVING errors than end users querying the more-traditional data structure ($F_{12,408}$ = 1.89, p = 0.0337, two-tail test). These results should be interpreted with caution as there were only a small number of queries requiring HAVING clauses.

Table 3. Effect of Ontological Clarity on Different Types of Semantic Errors

Panel A. Effect of Ontological Clarity on Total Number of Select Errors					
Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.2838	24	22.89	6.34	0.0001
Error		384	3.61		
Query		11	37.77	10.46	0.0001
Group (Query)		12	10.27	2.85	0.0009
GPA		1	25.44	7.05	0.0083

Panel B. Effect of Ontological Clarity on Total Number of From Errors					
Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.2256	24	5.22	4.66	0.0001
Error		384	1.12		
Query		11	8.95	7.99	0.0001
Group (Query)		12	2.24	2.00	0.0230
GPA		1	0.46	0.41	0.5231

Panel C. Effect of Ontological Clarity on Total Number of Where Join Errors					
Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.3338	24	61.55	8.02	0.0001
Error		384	7.68		
Query		11	93.29	12.15	0.0001
Group (Query)		12	37.44	4.88	0.0001
GPA		1	0.34	0.04	0.8336

Panel D. Effect of Ontological Clarity on Total Number of Where Condition Errors					
Source	R^2	df	Mean Square	F Value	Pr > F
Model	0.3679	24	26.64	9.31	0.0001
Error		384	2.86		
Query		11	47.88	16.73	0.0001
Group (Query)		12	9.24	3.23	0.0002
GPA		1	8.96	3.13	0.0776

Conclusions, Limitations, and Future Research

Using an experiment, this study examined, in detail, relationships between the level of ontological clarity of data structures and query performance. In relation to efficiency, end users querying a more-traditional data structure did not take significantly longer to compose their queries than end users querying an ontologically clearer data structure.

The results indicate that end users querying the more-traditional data structure made significantly more semantic errors when composing their queries than end users querying the ontologically clearer data structure (i.e., ontologically clearer data structures were associated with greater query effectiveness). End users querying the more-traditional data structure made significantly more SELECT and WHERE CONDITION errors than end users querying the ontologically clearer data structure. End users querying

the more-traditional data structure, however, made significantly fewer FROM and WHERE JOIN errors than end users querying the ontologically clearer data structure. The results further indicate that end users querying the ontologically clearer data structure were significantly more confident in the accuracy of their queries than end users querying the more-traditional data structure.

This paper provides significant contributions to research into the effects of ontological clarity and human-computer interactions. First, this research examined the influence of ontological clarity on end-user query performance (effectiveness, efficiency, and confidence). Second, the research extends Bodart et al.'s (2001) research relative to problem solving by confirming that end users made fewer semantic errors and were more confident in the accuracy of their queries when retrieving data from an ontologically clearer data structure.

This study has several limitations. First, the usual caveats associated with laboratory experiments limit the generalizability of the results. Second, the research used students as participants. These participants had, however, received training in information technology and business-related subjects. Their level of query proficiency was likely to be typical of end users in many organizations. Third, the size of the sample may not have been large enough to have sufficient power to reject some of the null hypotheses. Fourth, the time pressures in the experiment may be different from those faced by query developers in a business environment.

Future research is needed to improve end users' abilities to extract the information they need. Two examples of such research are noted here. First, this experiment needs to be replicated in organizational settings. Second, this research could be extended to examine the effects of ontological clarity on larger, more realistically-sized data structures.

References

Axelsen, M., Borthick, A. F., and Bowen, P. L. "A Model for and the Effects of Information Request Ambiguity and End-User Query Performance" in *Proceedings of the 22nd International Conference on Information Systems*, V. Storey, S. Sarkar, and J. I. DeGross (Eds.), New Orleans, LA, December 2001, pp. 537-542.

Bodart, F., Sim, M., Patel, A., and Weber, R. "Should Optional Properties Be Used in Conceptual Modeling? A Theory and Three Empirical Tests," *Information Systems Research* (12:4), 2001, pp. 384-405.

Borthick, A. F., Bowen, P. L., Jones, D. R., and Tse, M. H. K. "The Effects of Information Request Ambiguity and Construct Incongruence on Query Development," Decision Support Systems, (32:1), 2001a, pp. 3-25.

Borthick A. F., Bowen P. L., Liew, S., and Rohde, F. H. "The Effects of Normalization on Query Developer Query Errors: An Experimental Evaluation," *International Journal of Accounting Information Systems* (2:4), 2001b, pp. 195-223.

Bunge, M. *Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World*, Reidel, Boston, MA, 1977.

Burton-Jones, A., and Weber, R. "Understanding Relationships with Attributes in Entity-Relationship Diagrams," in *Proceedings of the 19th International Conference of Information Systems*, R. A. Hirschheim, M. Newman, and J. I. DeGross (Eds.), Helsinki, 1998, pp. 214-228.

Campbell, D. J. "Task Complexity: A Review and Analysis," *Academy of Management Review* (13:1), 1988. pp. 40-52.

Chan, H. C., Siau, K. L., and Wei, K. K. "Effects of Query Complexity and Learning on Novice User Query Performance with Conceptual and Logical Database Interfaces," *IEEE Transactions on Systems, Man and Cybernetics, Part A (Systems and Humans)* (34:2), 2004, pp. 276-281

Chan, H. C., Tan, B. C. Y., and Wei, K. K. "Three Important Determinants of User Performance for Database Retrieval," *International Journal of Human Computer Studies* (51:1), 1999, pp. 895-918.

Chan, H. C., Wei, K. K., and Siau, K. L. "User-Database Interface: The Effect of Abstraction Level on Query Performance: A Field Experiment," *MIS Quarterly* (17:4), 1993, pp. 441-464.

Chen, P. S. "The Entity-Relationship Model—Toward a Unified View of Data" *ACM Transactions on Database Systems* (1:1), 1976, pp. 9-36.

Gardner, W. B., and Serra, M. "An Object-Oriented Layered Approach to Interfaces for Hardware/Software Codesign of Embedded Systems," in *Proceedings of the 31st Annual Hawaii International Conference on System Sciences, Volume 7, Software Technology*, IEEE Computer Society Press, Los Alamitos, CA, January 1997, pp. 197-203.

Gemino, A. "To Be or May To Be: An Empirical Comparisoin of Mandatory and Optional Properties in Conceptual Modelling," in *Proceedings of the Annual Conference of the Administrative Science Association of Canada, Information Systems Division*, Saskatoon, Saskatchewan, 1998, p. 33-4.

Hoffer, J. A., George, J. F., and Valacich, J. S. *Modern Systems Analysis and Design* (4th ed.), Addison Wesley Longman, Reading, MA, 2004.

Jih., K., Bradbard, D., Snyder, C., and Thompson, N. "The Effects of Relational and Entity-Relationship Data Models on Query Performance of Query Developers," *International Journal of Man-Machine Studies* (31:3), 1989, pp. 257-267.

March, J., and Simon, H. A. *Organizations,* John Wiley, New York, 1958.

Rho, S., and March, S. T. "An Analysis of Semantic Overload in Database Access Systems Using Multi-Table Query Formulation," *Journal of Database Management* (8:2), 1997, pp. 3-14.

Shanks, G., Tansley, E., and Weber, R. "Using Ontology to Validate Conceptual Models" *Communications of the ACM* (46:10), 2003, pp. 85-89.

Suh, K. S., and Jenkins, A. M. "A Comparison of Linear Keyword and Restricted Natural Language Database Interfaces for Novice Users," *Information Systems Research* (3:3) 1992, pp. 252-272.

Wand, Y., and Wang, R. Y. "Anchoring Data Quality Dimensions in Ontological Foundations," *Communications of the ACM* (39:11), November 1996, pp. 86-95.

Wand, Y., and Weber, R. "On the Ontological Expressiveness of Information System Analysis and Design Grammars," *Journal of Information Systems* (4:4), 1993, pp. 299-330.

Wand, Y., Storey, V. C., Weber, R. "An Ontological Analysis of the Relationship Construct in Conceptual Modeling," *ACM Transactions on Database Systems* (24), 1999, pp. 494-528.

Weber, R. "Ontological Foundations of Information Systems," Coopers and Lybrand Accounting Research Methodology Monograph, Melbourne, Australia, 1997.

Weber, R., and Zhang Y. "An Analytical Evaluation of NIAM's Grammar for Conceptual Schema Diagrams" *Information Systems Journal* (6:2), 1996, pp. 147-70.

Appendix A. Data Structures
More-Traditional

Ontologically Clearer

Stocktake_results
Item_no*
Stocktake_date*
System_count
Actual_count

Inventory_items
Item_no*
Merchandise_line
Item_description
Item_cost_price
Item_sell_price
Qty_on_hand
Re_order_point
Re_order_qty

No_stocktake_results
Item_no*

Employees
Staff_no*
First_name
Surname
Street
City
State
Postcode
Phone
DOB
Date_employed
Function

Customers
Customer_no*
First_name
Surname
Street
City
State
Postcode
Phone
Type
Credit_limit
Deliver_days

Inventory_items_no_sales
Item_no*

Inventory_items_with_sales
Item_no*

Awaiting_pick
Sales_order_no*
Item_no*

Awaiting_ship
Sales_order_no*
Item_no*
Picked_by_employee

Sales_order_items
Sales_order_no*
Item_no*
Item_sale_price
Qty_ordered

Awaiting_accept
Sales_order_no*
Item_no*
Picked_by_employee
Qty_shipped

Sales_orders
Sales_order_no*
Customer_no
Cust_order_no
Staff_no
Order_date
Want_date

Accepted
Sales_order_no*
Item_no*
Picked_by_employee
Qty_shipped
Qty_accepted

Not_shipped_and_not_paid
Sales_order_no*

Not_shipped_and_paid
Sales_order_no*
Paid_date
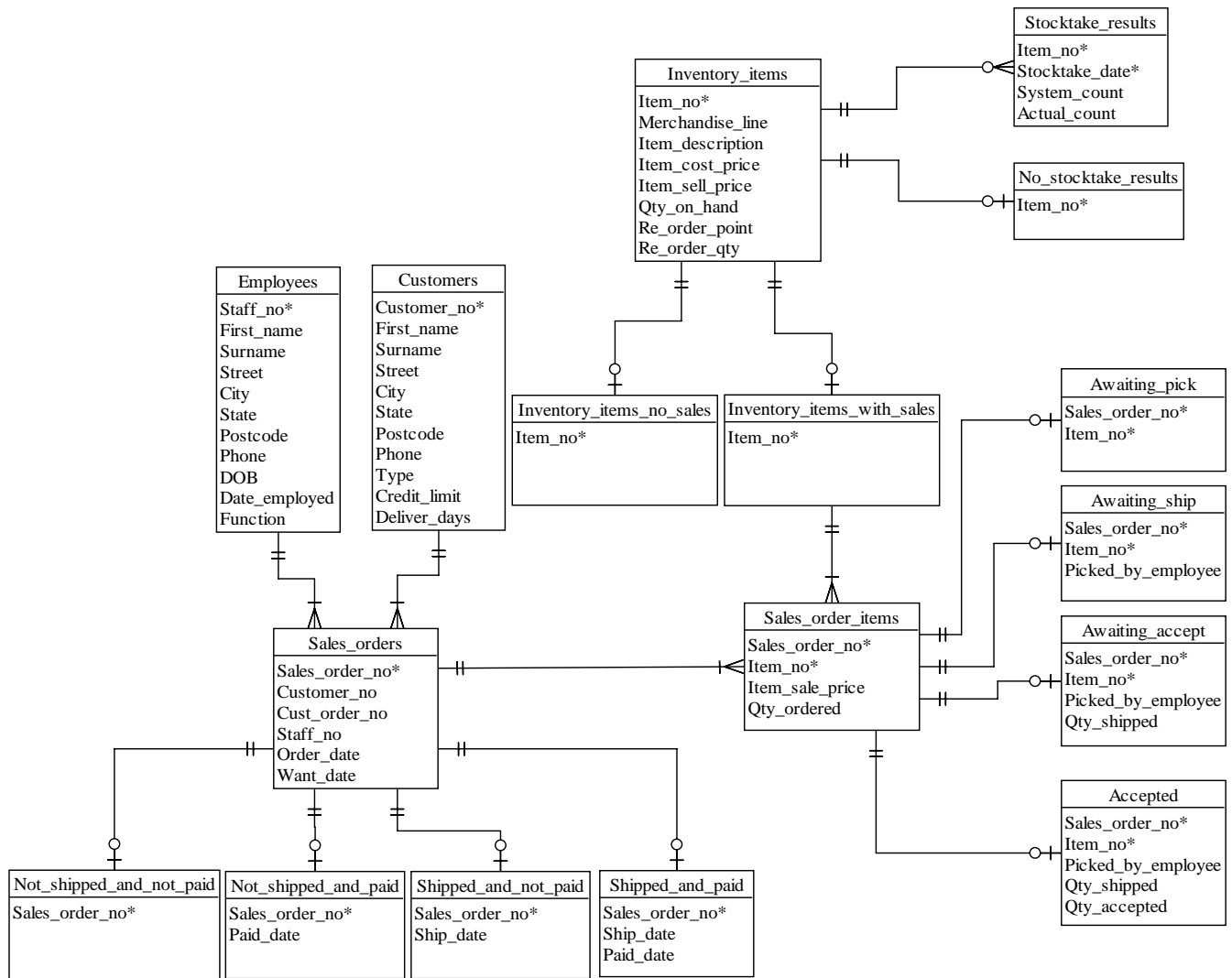
Shipped_and_not_paid
Sales_order_no*
Ship_date

Shipped_and_paid
Sales_order_no*
Ship_date
Paid_date

Appendix B. Information Requests

Q1.   List staff number, first name, and surname for employees employed after 10 August 2002.

Q2.   List the sales order number, customer number, want date, and ship date for orders that are shipped and not paid.

Q3.   List the sales order number and the number of sales order items for items that are yet to be picked.

Q4.   List employee first name, surname, and the number of sales orders for sales orders that are paid but not shipped.

Q5.   For items that have been accepted, list item number, sum of quantity shipped, and sum of quantity accepted.

Q6.   For sales orders that have not yet been shipped, list sales order number, customer number, customer first name, and customer surname.

Q7. For inventory items that have not been picked and have never been included in a stock take, list inventory item number, item description, and quantity on hand.

Q8. For orders placed after 1 August 2003 that have been accepted, list customer number, customer names, and the number of distinct orders where the quantity shipped of one or more items was more than 100 units less than the quantity ordered.

Q9. List the item number, item description, and number of orders for items that have never been included in a stock take but have been ordered more than five times and are on orders that have been both shipped and paid.

Q10. List sales order number, item number, percent of the item rejected, and the first name and surname of the employee who picked the item for sales orders which were paid before they were shipped and 50 percent or more of the item was rejected.

Q11. List customer number, sales order number, the number of items not yet picked, and the number of items not yet shipped for sales orders with number of items not yet picked less than 3 and the number of items not yet shipped greater than 10.

Q12. List customer number, customer first name, customer surname, and percent of sales orders shipped after the want date for customers with more than 10 sales orders.

Q13. List merchandise line, the percent of items in that merchandise line that have never been sold, and the percent of inventory cost of the items that have not been sold relative to the total inventory cost for that merchandise line. List only those merchandise lines with either percent of items that have never been sold greater than 10 percent or the percent of dollar cost of items that have never been sold greater than 20 percent.

Q14. For all customers, list the customers first name and surname and, if they placed orders in September 2003, also include the total amount they have paid for orders placed during September 2003, and the total amount they owe for items they have accepted but have not paid for orders placed during September 2003.

Appendix C. Error Counting Form

Name	Question Number	Attempts

SEMANTIC
Keywords and Logical Operators

View	Select	From	Where	Group by	Having	Order by

Set Operators

Where	Union	Intersect	Minus

Symbols and Relational Operators

View	Select	From	Where	Group by	Having	Order by

Tables

View	Select	From	Where	Group by	Having	Order by

Attributes

View	Select	From	Where	Group by	Having	Order by

Values

View	Select	From	Where	Group by	Having	Order by