COMMONSENSE KNOWLEDGE SUPPORT
IN DATABASE DESIGN EXPERT SYSTEMS

by

JIE DING

B.Eng., Tsinghua University, 1990

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
(Faculty of Commerce and Business Administration)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September 1994

© Jie Ding, 1994
In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature)

Department of Management Information Systems

The University of British Columbia
Vancouver, Canada

Date Sep. 21, 1994
ABSTRACT

Conceptual database design is the most critical and difficult phase in designing a database centered application. It usually requires database design experts which are hard to find and expensive. There has been some effort in building expert systems, including the View Creation System (VCS), to support this design process. However, all of these systems lack commonsense knowledge human experts have and therefore can not provide effective support to the user. A prototype commonsense knowledge base, the Commonsense Business Reasoner (CBR), has been built for the VCS. But it is not fully implemented and not connected to the VCS. The usefulness of commonsense knowledge in a database design expert system has not been studied. In this paper, a new domain of relevance ontology was proposed to store domain of relevance for commonsense knowledge. Combined with the Naive Semantics ontology used in the CBR, a new commonsense knowledge base structure was built and implemented. This was integrated into the original VCS and can provide interactive commonsense knowledge support during the design process. Other improvements were also done to the VCS to make it more user friendly. A full scale empirical study was conducted to test the effectiveness of the commonsense module. The results indicated subjects perceived the system with commonsense knowledge easier to use and finished the task in less time. However, there is no statistically significant difference in the design quality. Explanations to these results are discussed as well as future research directions.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ix</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Objective of the Thesis</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Outline of the Thesis</td>
<td>3</td>
</tr>
<tr>
<td>2. CONCEPTUAL DATABASE DESIGN</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Overview of Conceptual Database Design</td>
<td>4</td>
</tr>
<tr>
<td>2.2 E-R Model</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Relational Model</td>
<td>5</td>
</tr>
<tr>
<td>2.4 Process Model of Conceptual Database Design</td>
<td>6</td>
</tr>
<tr>
<td>2.5 Design Process of Expert vs. Novice</td>
<td>8</td>
</tr>
<tr>
<td>3. EXPERT SYSTEMS FOR CONCEPTUAL DATABASE DESIGN</td>
<td>10</td>
</tr>
<tr>
<td>3.1 Overview</td>
<td>10</td>
</tr>
<tr>
<td>3.2 The View Creation System (VCS)</td>
<td>10</td>
</tr>
<tr>
<td>3.2.1 Creation of the Initial E-R Model</td>
<td>10</td>
</tr>
<tr>
<td>3.2.2 Creation of the Final Relational Model</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Problems with the VCS</td>
<td>11</td>
</tr>
<tr>
<td>3.3.1 Lack of Commonsense Knowledge</td>
<td>11</td>
</tr>
<tr>
<td>3.3.2 Poor User Interface</td>
<td>12</td>
</tr>
<tr>
<td>4. COMMONSENSE KNOWLEDGE</td>
<td>13</td>
</tr>
</tbody>
</table>
4.1 Overview

4.2 Commonsense vs. Expert Knowledge
   4.2.1 Structure
   4.2.2 Generality
   4.2.3 Availability
   4.2.4 Usability

4.3 Naive Semantics

4.4 Commonsense Business Reasoner
   4.4.1 Has Attribute
   4.4.2 Has Key
   4.4.3 Synonym
   4.4.4 Ako (A Kind Of)
   4.4.5 Subject
   4.4.6 Object

4.5 Domain of Relevance for Commonsense Knowledge
   4.5.1 Organization of the Domain of Relevance Knowledge
   4.5.2 Dimensions of Relevance
   4.5.3 Some Examples

5. THE NEW VCS

5.1 Architecture of the New VCS

5.2 Consultation of Commonsense During Design Process
   5.2.1 Stage One
   5.2.2 Stage Two

5.3 Other Improvements
   5.3.1 Direct Input Method
   5.3.2 Smart Cursor Positioning
   5.3.3 Context Sensitive Message
   5.3.4 Configurable Features

6. HYPOTHESIS DEVELOPMENT

6.1 The Research Model
   6.1.1 Dependent Variables
   6.1.2 Independent Variables

6.2 Commonsense Knowledge Effect

6.3 Task Complexity Effect

6.4 Interaction Effect
## 7. RESEARCH METHODOLOGY

### 7.1 Experimental Design

### 7.2 Subjects

### 7.3 Independent Variables
- **7.3.1 Commonsense Knowledge**
- **7.3.2 Tasks**

### 7.4 Dependent Variables
- **7.4.1 Time Taken to Finish the Task**
- **7.4.2 Design Quality**
- **7.4.3 Perceived Ease of Use**

### 7.5 Experimental Procedure and Data Collection
- **7.5.1 Pre-Test Questionnaire**
- **7.5.2 Introduction and Tutorial**
- **7.5.3 Understanding the Test Case Description**
- **7.5.4 Creation of the E-R Model**
- **7.5.5 Post-Test Questionnaire**

## 8. DATA ANALYSIS

### 8.1 Commonsense Knowledge Effect
- **8.1.1 Time and Perceived Ease Of Use**
- **8.1.2 Design Quality**

### 8.2 Task Complexity Effect
- **8.2.1 Time**
- **8.2.2 Design Quality**
- **8.2.3 Perceived Ease Of Use**

### 8.3 Interaction Effect

## 9. CONCLUSION AND FUTURE RESEARCH

### 9.1 Limitation of the Study

### 9.2 Future Research

## 10. BIBLIOGRAPHY

## APPENDIX A  GRADING SCHEME FOR DESIGN QUALITY

## APPENDIX B  STANDARD DESIGN
LIST OF TABLES

Table 1 Time taken to complete task by groups (minute) 49
Table 2 Quality of design by groups (percentage) 49
Table 3 Perceived ease of use by groups (total score on the EOU questionnaire, ranges from -21 to 35) 49
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Process model of conceptual database design (Batra and Davis, 1992)</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Structure of the new VCS</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>Research model</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>Average number of extra entities and relationships entered by each subject</td>
<td>53</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I own an unpayable debt of gratitude to Professor Robert Goldstein, my thesis advisor. His academic and personal support made this thesis possible. I also want to thank the two other committee members, Professors Carson Woo and Dean Uyeno for their insightful comments and suggestions. Professor Izak Benbasat gave many valuable suggestions to the experimental design and data analysis. Professor Andy Trice provided feedback on the domain of relevance ontology. Special thanks to fellow graduate students Becky Lau, Kai, Hao Zhao, and Jiye Mao for testing the system and commenting.

Finally, I want to thank my wife Jennifer, who has given me tremendous emotional support throughout the whole program. Without her encouragement and sacrifice, this paper would not have been possible.
1. INTRODUCTION

Database design is the process of developing database structures from user requirements for data (McFadden and Hoffer, 1988). It usually includes requirements analysis, conceptual design and physical design. Among them, conceptual design is the most critical and difficult. In this stage, the designer needs to represent the system structure and user requirements in a set of conceptual data constructs (Date, 1990). The conceptual design is usually accomplished by a database design expert working together with a domain expert. With the advance of end user computing, some expert systems (Storey and Goldstein, 1988; Choobineh et al., 1988; Dogac et al., 1989) were developed to allow the end user, who can be regarded as a domain expert but a novice database designer, to design a conceptual data model independently.

However, the results of studies on effectiveness of these expert systems and other decision support systems in general have been mixed (Sharda, et al., 1988; Gilbert, 1993). Some external factors are proposed to explain the results, e.g., differences in task, application, personality and experimental design. These external factors certainly to some extent affect the research results, but internal characteristics of the expert system should be examined as well.

One explanation is that while these expert systems possess a high degree of expertise in the database design domain, they usually know very little about the application domain and the ‘real world’ (Goldstein and Storey, 1991). This causes several problems. First, the communication between the user and the expert system is often impaired because the user has to spend much time entering ‘trivial’ information. This leads to low efficiency and poor user
INTRODUCTION

satisfaction. Second, current expert systems completely rely on the user to conceptualize domain structures. This often leads to inferior design quality because of the user’s inexperience with conceptual data constructs. A human expert, on the other hand, already has some commonsense knowledge and does not need to ask the end user every detail regarding the real system. He can also use his knowledge about the application domain to quickly discover underlying structures and match them to conceptual data constructs.

To address these problems, Goldstein and Storey (1991) developed a Commonsense Business Reasoner (CBR) for their database design expert system, the View Creation System (VCS). CBR can store commonsense business knowledge and reason about it. Although CBR provides a framework for further studies in commonsense knowledge reasoning, it was not fully implemented and not connected with the VCS. So far, no other commonsense knowledge module has been developed for any database design expert system. As a result, the usefulness of commonsense knowledge in a database design expert system has not been studied.

1.1 Objective of the Thesis

The objectives of this thesis are to develop a new VCS with a commonsense knowledge module and to evaluate the effectiveness of this commonsense module. The organization of the commonsense knowledge base is based on Naive Semantics and domain of relevance ontology. This knowledge base was integrated into the original VCS. The effectiveness of the new system was evaluated in a lab experiment focusing on efficiency, quality of design and perceived ease of use.
1.2 Outline of the Thesis

The next Chapter describes some basic concepts in conceptual database design and two of the most often used conceptual design models. The process model for conceptual database design is also presented and adopted as the research framework. Chapter 3 introduces expert systems developed for conceptual database design and their limitations. Chapter 4 describes the concept of the commonsense knowledge and provides the theoretical foundation of the commonsense knowledge base developed in the subsequent Chapters. Also included is a discussion of domain of relevance ontology. Chapter 5 introduces the structure and implementation of the new VCS. Chapter 6 to Chapter 8 describe the experiment. Chapter 6 proposes the research hypotheses. Chapter 7 discusses and justifies the research methodology chosen. The experimental design is also described in Chapter 7. Chapter 8 reports data analysis results. The conclusion and future research suggestions are presented in Chapter 9.
2. CONCEPTUAL DATABASE DESIGN

2.1 Overview of Conceptual Database Design

Conceptual database design is the process of representing the information content in constructs independently of the way in which the data is physically stored (Goldstein, 1985; McFadden and Hoffer, 1988; Elmasri and Navathe, 1989; Date, 1990). The purpose of conceptual design is to understand the system's basic elements, their relationships and constraints. The input to a conceptual database design is the user requirement specification and the result is a global data model.

In the business domain, a conceptual data model represents the entities in an organization and the relationships among them. One of the advantages of using a conceptual data model is its independence of implementation consideration (Korth and Silberschatz, 1986). This allows a conceptual model to be used in many application programs, and on the other hand, allows an application domain to be modeled using various conceptual data models. A conceptual model's absence of implementation detail also makes it easier to understand and forms the basis for communication with end users.

2.2 E-R Model

The most widely used conceptual data model is the Entity-Relationship (E-R) model (Chen 1976, 1977). The E-R data model is based on the assumption that the real world consists of a set of basic objects and relationships among these objects.
There are three basic constructs in the E-R model: entity, relationship between entities and attribute. An entity is defined as “a thing which can be distinctly identified” (Chen, 1976). It is usually represented by a noun. For example, in a library database, book and borrower can be treated as entities. A relationship is defined as “an association among entities” (Chen, 1976). It corresponds to a verb in natural language. For example, borrow is a relationship among borrower and book in the above library database. An attribute is a kind of property of the entity or relationship it is related to. All the instances of an entity or relationship share its attributes. For example, if employee number and name are defined as attributes for entity employee, then all the employees would have an employee number and a name.

One advantage of using the E-R model as a conceptual design tool is its simplicity and naturalness (Brodie, 1984 and Konsynski, 1979). There are only three constructs in the model and they can be easily understood by novice users. The E-R model provides a communication medium between the users and the designer.

The E-R model was chosen as the basis for the VCS’s design approach for exactly the same reason (Storey, 1988).

2.3 Relational Model

While the E-R model is regarded as natural to the real world, the relational model (Codd, 1970) is the model used by most commercial products. The relational model represents the database using a collection of tables (Codd, 1970; Korth and Silberschatz, 1986). Each table is called a relation and has a unique name and set of attributes. The relation is the only construct in the relational model.
Relation models are based on relational algebra and relational calculus which give them a solid theoretical foundation (Codd, 1970). Most of the current commercial database products use the relational model. The VCS, although using the E-R model as its design basis, generates a relational model in 3rd Normal Form (3NF) as its final result. This can be then easily implemented using commercial products.

2.4 Process Model of Conceptual Database Design

As discussed above, although there has been a large amount of research done in building the theoretical foundation of conceptual database models, relatively little is known about the process whereby a novice designer applies these concepts and rules. Because of this lack of research in the design process, the factors that might affect a novice designer’s performance are not identified and the dynamics of the process are not studied. To study if and why commonsense knowledge might help in the design process, the process itself needs to be understood.

Batra and Davis (1992) studied the verbal protocol and written trace of several experts and novices designing a data base. Based on this data, they developed a three level process model of conceptual database design (Figure 1).
The specific subject activities at each level are the following:

**Enterprise Level** During the enterprise level, the subject reads, contemplates the requirements, comments, elicits user requirements, seeks clarifications or establishes connections. The focus at this level is on developing a reasonable understanding of the problem domain.

**Recognition Level** At this level, the subject focuses on some specific aspect of the user requirements and tries to understand the sub-problem at hand, and this triggers the appropriate knowledge structures in his repertoire.

**Representation Level** The representation phase constitutes the operationalization of the subject’s understanding of the structure into a conceptual data representation using a
specific data model. This phase also includes the verification of the representation to ensure it satisfies the user requirement.

It was found that a subject would typically stay at one level for some time before moving on to another. However, there were substantive differences in the pattern.

The three level process model, building on previous work on information process theory and empirically tested, is adopted as the research framework for the experiment on the effectiveness of commonsense knowledge module.

2.5 Design Process of Expert vs. Novice

Batra and Davis (1992) found human experts use commonsense knowledge in all the three levels. At the enterprise level, experts start using commonsense to understand basic objects and relations in the problem domain. Commonsense knowledge at this level helps an expert quickly establish a general picture of the application domain. For example, when an expert is told that he will be designing a database for a school, he will immediately know that there are basic objects like teachers, students, courses and so on.

The recognition phase sees the biggest difference between experts and novices. Experts tend to spend much less time at this stage than novices. During this phase, experts use commonsense knowledge to help identifying entities and attributes. For example, “student takes course” is a commonsense fact which should be modeled as a relationship because both student and course are entities and the mapping ratio between the two entities are many to many. An expert who has this commonsense knowledge can match it with the patterns in his database design knowledge base and represent it accordingly. An expert system without this
commonsense fact will rely on the user to identify both 'student' and 'course' are entities and there is a relationship between them.

It is often observed at the recognition stage, that an expert starts with a fuzzy picture of the real world obtained from the previous stage and uses his commonsense knowledge to map it to a data structure. Then the expert will actively elicit information from the user to try to verify and complete the selected data structure. Commonsense knowledge and database design knowledge together guide the expert during the recognition phase. For a novice designer, although he has commonsense knowledge for the problem domain, the lack of database design expertise makes it hard for him to use it.

At the last stage, the designer generates the data constructs identified in the previous level and ensures that they satisfy the syntax and constraints of the data model.

Based on the above findings, it is argued that the recognition phase is the most difficult phase and where commonsense knowledge can play an important role. For an expert system to be as effective as a real expert designer, it must be able to provide both strong database design knowledge and commonsense knowledge support to the user.
3. EXPERT SYSTEMS FOR CONCEPTUAL DATABASE DESIGN

3.1 Overview
Because of the complexity of conceptual design, a lot of effort has been put into the study of the conceptual design process and especially ways to improve the design result of computer-novice end-users. One major approach is the development of expert systems that automate the design process or help the designer in the design process (Storey and Goldstein, 1988; Choobineh et al., 1988; Dogac et al., 1989).

3.2 The View Creation System (VCS)
The VCS is a conceptual database design expert system developed by Storey and Goldstein (1988). It is based on a formalized methodology for generating and integrating user views. It uses the E-R model discussed in Section 2.2 as the view modeling tool. The design process with the VCS system can be divided into two stages:

3.2.1 Creation of the Initial E-R Model
In this stage, the user expresses his view of the database in terms of entities and relationships. For each entity, the user enters its name, attributes, and keys, and for each relationship, its cardinality and attributes.

3.2.2 Creation of the Final Relational Model
In this stage, the system checks the user's initial E-R model, identifies problems, seeks more information to fix the problems and converts it into a relational model.
According to Batra and Davis’ process model (1992), the VCS facilitates the design process basically at the representation level. Instead of directly representing the system structure using a relational model, which is a less direct representation of the real world, the user can start with the more natural E-R model and later let the VCS convert it to a relational one. However, the user has to capture and model the real system before he can represent it in an E-R model. So the enterprise and recognition level processes are essentially not supported in the original VCS.

3.3 Problems with the VCS

Gilbert (1993) conducted a lab experiment comparing users’ performance working with the VCS and human experts. Several problems of the VCS were identified. Among them, the lack of commonsense knowledge and poor user interface are the most important.

3.3.1 Lack of Commonsense Knowledge

As discussed in Section 2.5 about the difference between experts and novices, experts uses commonsense knowledge in the enterprise and recognition phases to help understand the user requirements and recognize data constructs. But so far, no expert system has used commonsense knowledge in the design process. The lack of commonsense reasoning ability in expert systems in general has been criticized by many researchers (Kolata, 1982; McCarthy, 1983; Meltzer, 1985; Buchanan, 1988). The VCS also suffers from this problem (Goldstein and Storey, 1991; Gilbert, 1993). It has rules and facts about database design process but nothing about the application domain. This creates an obstacle in the communication between
the user and the system. The user has to routinely enter "trivial" information and answer "obvious" questions. The system’s image as an expert is damaged as a consequence.

3.3.2 Poor User Interface
The user interface is another major factor affecting the end-user's performance on the VCS (Gilbert, 1993). The input of data is not direct and intuitive. For example, to enter a new entity or relationship into the system, the user first goes to the main "Entity and Relationship" screen where all the entities and relationship are displayed. Then instead of typing directly, the user has to press [INSERT] key first to open a separate input box and enter the new entry there. When finished, he has to press [ESC] to go back to the main menu. Procedures like this make it hard for novice users to remember and often discourage them from learning how to use it at all.

To address above problems, especially the lack of commonsense knowledge, various effort was made. This will be introduced in the following Chapters.
4. COMMONSENSE KNOWLEDGE

4.1 Overview

The idea of incorporating commonsense knowledge into computer programs has a long history (McCarthy and Hayes, 1969; Hobbs and Moore, 1985; Lenat et al., 1986).

There has been no common definition for commonsense knowledge. Dahlgren (1989) describes commonsense knowledge as ‘a set of naive belief, at times vague and inaccurate, about the way the world is structured’. Ein-Dor (1985) defines commonsense as ‘what any participant in a culture expects any other participant in that culture to know when meeting for the first time and before any exchange takes place between them’. Goldstein and Storey (1991) define commonsense as ‘general knowledge about the way the world works’. It can be summarized that commonsense knowledge is everyday knowledge everybody knows about the world.

Because of the popularity of expert systems in recent years, there has been a lot of attention paid to expert knowledge. Commonsense knowledge, although considered important to overcome the brittleness of these systems (Winograd and Flores, 1987), remains largely untouched. A comparison between commonsense knowledge and expert knowledge will illustrate why it is so hard to model commonsense knowledge.
4.2 Commonsense vs. Expert Knowledge

Although they are both important components of a human being’s knowledge of the world, commonsense knowledge and expert knowledge have some distinguishing characteristics.

4.2.1 Structure

The accumulation process of commonsense knowledge is not structured and often unintentional. It is gathered through one person’s personal interaction with the environment around him. This interaction is constant, informal, and incomplete. Usually no effort is made to analyze and organize the information gathered and memorize it as formal knowledge.

Expert knowledge, on the other hand, is developed through intentional cognitive process (Winograd and Flores, 1988). An expert learns directly from his own experience and/or indirectly from other people’s knowledge. During the learning process, all of the input information is processed and used to enhance or modify the existing knowledge framework. Expert knowledge is processed in an effort to better understand and form a clear view of the subject. Therefore, expert knowledge has formal structures lacked by commonsense knowledge.

This unstructured nature makes commonsense knowledge very difficult to represent using formal language.

4.2.2 Generality

Commonsense knowledge comes from people’s everyday life and is easily shared by a large number of people through communication.
On the other hand, an expert usually has only in-depth knowledge about his own domain. Because the effort required to acquire expert knowledge is so great, it is understandable that expert knowledge is not shared by most people.

4.2.3 Availability

Because commonsense knowledge is obtained from everyday life, it is not confined to a small number of experts. It is readily available from ordinary people. In some sense, everybody can be treated as an expert on commonsense knowledge.

4.2.4 Usability

Commonsense knowledge is used in almost any place where human thinking is required. Analogy is used to bridge the more general commonsense knowledge and the specific question at hands.

Expert knowledge is usually generated to solve problems in a specific domain. This focus limits its use. Furthermore, the representation of the expert knowledge in most expert systems is domain dependent, which makes the transferring of knowledge across domains even harder.

4.3 Naive Semantics

Because of the above characteristics of commonsense knowledge, it is understandable that so far there is little success in the AI community in developing a simple, elegant formalism to represent it. One of the most ambitious project in this area is the CYC project (Lenat et al., 1986; Lenat et al., 1990; Guha and Lenat, 1990). They are trying to build a comprehensive commonsense knowledge base that can be accessed by various expert systems. Although the
COMMONSENSE KNOWLEDGE

CYC project may eventually accumulate enough knowledge and provide general knowledge support to domain specific expert systems, the feasibility and the cost effectiveness of thus a system can not be proved.

Naive Semantics (Dahlgren, 1989) provides a much more concise and implementable framework. The most important feature of Naive Semantics is that it represents word meanings as commonsense knowledge and not sets of primitives. In traditional AI research, word meaning has been treated as critical. For example, if ‘flying’ is defined as an attribute for ‘bird’, then all the creatures that can not fly would be automatically disqualified. Naive Semantics, representing word meaning in commonsense, treats ‘flying’ as a typical rather critical description of ‘bird’. In other word, Naive Semantic representations are probabilistic and non-monotonic.

Because Naive Semantics tries to include all commonsense knowledge into each word’s meaning, the number of features needed is theoretically limitless. To keep the size of the knowledge base manageable, Dahlgren (1989) developed a Kind Type system which uses an ontology to organize and represent commonsense knowledge. A set of 54 feature types of nouns and verbs is compiled based on psycholinguistic experiments in prototype theory (Rosch et al., 1976; Ashcraft, 1976; Fehr and Russell, 1984; Dahlgren, 1985). NewSelector System, an expert system built by Lord and Dahlgren (1990) stores business and financial knowledge using the Kind Type system.
4.4 Commonsense Business Reasoner

Based on Naive Semantics theory and ontology, Goldstein and Storey developed a Commonsense Business Reasoner (CBR) (1991) as a framework to capture commonsense knowledge in the business domain to facilitate conceptual database design.

In the CBR, two distinct types of knowledge are represented: classificatory and generic. The first type provides meanings of words and the second type relationships among objects. The ontology from Naive Semantics is used to provide a framework for the generic commonsense knowledge stored.

The 54 feature types proposed by Dahlgren (1989) provide a comprehensive list for commonsense knowledge encoding. Obviously, not all the feature types can be populated and there are some additional feature types specific to database design need to be added. Some of the most important feature types used in the CBR are discussed here.

4.4.1 Has Attribute

Attributes have a very important role in E-R models as they actually store the information about entities and relationships. Entering attributes for each entity is a routine task when using the VCS. The ‘Has Attribute’ feature type is created to store common attributes for a given word, saving input time and improving the human computer communication.

For example, the commonsense knowledge base may indicate that ‘person has attributes name and address’
4.4.2 Has Key
 Determining the key for an entity sometimes is obvious, other times tricky. It would be
very useful if the commonsense knowledge base can store some possible candidate keys and
prompt the user when needed. As other pieces of commonsense knowledge, the key stored in
the 'has key' feature type may not be the best key in a particular case, but it can still help the
user by providing an example of a common key.

One example of the 'Has Key' feature type is that a person's Social Insurance Number
(SIN) is a key.

4.4.3 Synonym
 Synonym is included to reflect the similarity between different expressions. For
example, a 'teacher' may be called 'instructor', 'professor' or 'lecturer'. These different
phrases share most attributes and commonsense knowledge about one of them often apply to
others as well.

4.4.4 Ako (A Kind Of)
 Inheritances exist in everyday life and in computer languages. It's also an important
concept in database design. Feature type 'Ako' captures inheritance by specifying that one
entity is a sub-class of another entity.

4.4.5 Subject
 Verbs usually represent relationships in the E-R model. A relationship in the VCS
involves two entities. The first entity is the subject of the relationship and the second entity is
the object of the relationship. To store this information, two feature types are used for verb in
the CBR. Among them, the subject feature type is included to store the information that a word is the subject in a relationship.

For example, 'university' is the subject of relationship 'university offers course'.

4.4.6 Object

The object feature type is included to store the information that a word is the object in a relationship.

For example, 'course' is the object of relationship 'university offers course'.

Using the subject and object feature types, both parties involved in a relationship are identified.

4.5 Domain of Relevance for Commonsense Knowledge

While Naive Semantics helps to provide a framework for storing commonsense, the ontology is still limited in that it provides only a rigid classification of terms. The major weakness of this approach is that the "relevance" of facts is not accounted for. Because a person learns commonsense knowledge of the world from his experience, the context and the relevance of the facts he observes become part of the knowledge. To fully capture his commonsense knowledge, the relevance of a fact must be 'tagged' to the fact. For example, the fact 'providing pension plan' may be related to the term 'organization'. However, this only applies to certain industries, or in certain countries. Although Naive Semantics' ontology attempted to provide a useful tool in organizing people's commonsense, the application is limited in terms of its ability to fully capture the relevance of facts.
By examining the relevance of terms, it is possible to define "dimensions of relevance" that are pertinent for enhancing the original ontology in the database design. For example, "sales tax" may be different in terms of relevance according to differences in political body. In some states or provinces, no sales tax is collected, but in some other places sales tax is collected. If the database design system has this domain of relevance knowledge stored, it will prompt the user about this commonsense knowledge only when it is relevant. Obviously this will make the design process more natural and efficient.

Before we identify possible dimensions for the domain of relevance knowledge, the organization of this knowledge needs to be defined.

4.5.1 Organization of the Domain of Relevance Knowledge

The domain of relevance knowledge is different from normal commonsense knowledge in a sense that it is a kind of meta knowledge, i.e. knowledge about knowledge. The Naive Semantics' ontology introduced in Section 4.3 was created only to store commonsense knowledge itself. A new ontology is required to store domain of relevance knowledge. One natural way to organize the new ontology is to divide it along several relevance dimensions. For example, geographic, industry, organization and application are frequently used to define the relevance of a term, and can be treated as relevance dimensions. Under each of these dimension, there are many nodes representing sub-classes in that dimension. For example, under the geographic dimension, there are Canada, US, Mexico sub-classes. Under Canada, there are different provinces, etc.
After the framework of the ontology has been set up, the actual relevance knowledge can be put it. The facts in the original ontology will be attached to different nodes in these dimensions in the new ontology based on their relevance. For example, the fact 'GST' will be attached only to node Canada in the geographic dimension, and to all industries except charity industry node in the industry dimension. For other dimensions that are not relevant to this fact, it will simply not be attached.

Using this structure, the domain of relevance information about commonsense knowledge is stored completely. There are two ways this information can be used. First, given a specific domain, all the relevant commonsense knowledge can be retrieved. This is done through searching all the dimensions for the given domain and recording the facts attached to them. If a fact is attached to all of the dimensions, then it is relevant in the given domain. Secondly, for a particular fact, its domain of relevance can be easily identified by searching all the dimensions for the attachment of this fact and the product of these domains is the result.

4.5.2 Dimensions of Relevance

Goldstein and Storey (1991) identified some contexts such as functions and organizations. The organization context can be further divided into industry, organization and user. Following this line of argument, more dimensions of relevance are proposed. These dimensions are part of the foundation upon which we construct our ontology of domain of relevance.
These dimensions are grouped into several categories. It should be noted that these classifications, like other commonsense knowledge, are somehow vague and insufficient. In addition, because of the parallel existing in business world, cross-classification is used.

We categorize all the dimensions into three groups, ENVIRONMENT, BUSINESS and PEOPLE. They correspond to three levels of domain of relevance.

4.5.2.1 Relevance of Environment

Besides the relevance related to a business itself, the economy and geography environment where the business exists also provide relevant commonsense information.

4.5.2.1.1 Economy

There are still centralized economies where most of the planning and controls are done by a central authority. The mechanism is so different that many economic terms such as price, tax and profit have different meaning. For example, the price of a product will be set by the authority instead of the market forces. Much of the business commonsense knowledge in a free market economy will not apply.

Dimension Stage of Development (Developed, Developing)

The stage of development also affects the relevance of knowledge. For example, in developing countries, the fact "using credit card" may be not relevant.

Dimension Type (Centrally Planned, Free Market)
4.5.2.1.2 Market

Markets can be classified into monopoly, oligopoly or competition markets. Though the real world doesn't fit in perfectly, this classification still reflects some distinct facts about markets. For example, the price is determined by the monopoly and the profit is higher in a monopoly market, while the price is more determined by the supply and demand in a competition market. Considering that the government may step in and introduce regulations, the regulated market was added.

Dimension Type of Market (Regulated, Monopoly, Oligopoly, Competition)

Market can be divided by their scopes, too. There are global market, domestic market and regional market.

Dimension Scope of Market (Global, Domestic, Regional)

4.5.2.1.3 Political Body

Different regions have different regulations regarding business activities. This can be reflected in tax, minimal wage, pension, etc. Examples of political bodies are:

Dimension Political Body (Canada (Provinces), US (States))

There is no need to start with a complete list of political bodies in the world. Instead, they can be added to the ontology whenever needed.

4.5.2.1.4 Geography

The geography environment is less important today than before. However, it still has strong direct or indirect impact on a business. Some of the geography dimensions are listed below.
COMMONSENSE KNOWLEDGE

Dimension Topography (Coastal, Mountain, Prairie, Desert, Metropolitan, Town, Countryside, West coast, East coast)

Dimension Orientation (North, South, East, West)

Dimension Continent (America, Asia, Africa, Europe)

4.5.2.2 Relevance of Business

The first category of dimensions deals directly with the particular business. It includes industry, company and product.

4.5.2.2.1 Industry

One of the most obvious and important dimensions identified in this category is industry. There are many different ways to classify industries (Krahn and Lowe, Mitchell, 1987; 1988; Reinecke and Schoell, 1977). However, they tend going into detail instead of capturing the general knowledge about industries.

One of the intention of our study is to build a frame that is general and flexible enough to facilitate further expansion. After carefully examining several industries, we identify the division of service and manufacturing industries as the top nodes in the industry dimension. These two kinds of industries are different in several ways. While service industries provide services that are usually non-physical processes, manufacturing industries produce products in both physical and non-physical forms. Service industries usually don't need raw materials, but manufacturing industries will need raw material to be transformed to the products. These two kinds of industries may also pay different taxes.
Service industries can be divided into transportation, communication, information, medical, tourism, insurance, banking and accounting. They can be further divided when necessary. The structure of the industry domain is listed below:

Dimension Industry (Service (Transportation (Airline, Shipping, ...), Communication (Mail, Telephone, ...), Information, Medical, Tourism, Insurance, Banking (Accounting, ...), Manufacturing (Automobile, Computer, ...))

The lower level details are not fully defined because they are less important and can be added whenever that particular industry in used.

4.5.2.2.2 Company
Another important aspect of relevance of business is company. Companies can be categorized by the ownership into sole proprietor, partnership and corporation. Corporations can be further divided into private and public owned corporations. These companies are different in terms of their employees and owners. Sole proprietor, partnership companies’ owners usually are the managers themselves. A corporation will pay tax on its profit while a sole proprietor or partnership company doesn't need to pay.

Dimension Ownership (Sole Proprietor, Partnership, Corporation, Private owned, Public owned)

Corporations can be distinguished by whether they sell stock to the public. This dimension will be useful in deciding the control in the company.

Dimension Stock (Open, Close)
Companies can also be classified by their goals. Profit and non-profit companies differ in their products, employees, taxes and profits.

**Dimension**  
*Goal (Profit, Non-Profit)*

One interesting dimension about the company may be the existence of trade union. In a company with strong trade union, the employees wages may be higher, so the cost of operation will be higher.

**Dimension**  
*Union Existence (Unionized, Non-unionized)*

4.5.2.2.3 Department/Division

Inside almost every companies, there are several different departments or divisions that have different but related functions. They played different roles in the operation of the company. Although we identified this as an dimension, we will not provide any value or sub-dimension because of the variety across companies. As in Section 4.1.1, they can be added incrementally.

**Dimension**  
*Department/Division*

4.5.2.2.4 Functional Area

Different functional areas have different terminology and different processes. They are usually clearly defined.

**Dimension**  
*Functional Area (Finance, Accounting, Marketing, Personnel)*
4.5.2.2.5 Product

Products differ greatly across industries, but they still can be roughly divided by their users: a product is either used by consumers directly or it may be input of another industry.

Products also may different in their taxation, some products like food are tax free, and others not. Products imported and exported are subject to duties and quota.

Dimension Product (End-user (Food, Clothes, ...), Intermediate Product (Mining product, ...)

4.5.2.3 Relevance of People

People play a central role in business. There are many dimensions upon which people can be classified. We will only select those relevant to business commonsense.

4.5.2.3.1 Occupation

A person's occupation is an important aspect of his characteristics. For example, professionals are usually paid on a salary basis, while unskilled workers are usually paid hourly. Maria Hirszowicz (1981) roughly categorized occupations into four groups:

Dimension Occupation (White-collar (Professional and technical, Managers and proprietors, Clerical, Sales), Blue-collar (Skilled workers, Semi-skilled workers, Unskilled workers), Service workers, Farm workers)

Obviously, one person's role, function and position in a business setting are closely related with his occupation. The domain of relevance information will help to construct the model of the business from human resource's perspective.
There are many more detail defined occupations (Reinecke and Schoell, 1977; Krahn and Lowe, 1988; Mitchell, 1987). However, as mentioned before, we are only interested in identifying a frame which will be the basis of further expansion.

4.5.2.3.2 Personal Attributes
People's personal attributes differ greatly and the facts in the ontology may only apply to certain kinds of people. Following are some personal attributes that representing different dimensions.

Dimension Sex (Male, Female)

Dimension Age (Child, Adult, Senior)

Dimension Marriage Status (Single, Married, Divorced)

Among above dimensions, the gender difference exists in most of today's business in terms of salary, occupation, etc. Age affects a person's role in the business and the society and marriage status has direct impact on taxes.

4.5.2.3.3 Personal Background
Like personal attributes, personal background affects a person’s position in a business, but its effect is more indirect.

Dimension Education (high school, college, university)

Dimension Income (below average, average, above average)
4.5.3 Some Examples

In the following, some business terms is examined according to the above dimensions to show the usefulness of the domain of relevance knowledge.

**Tax** Taxes are relevant to most of the dimensions identified earlier in this Section. A company making profit must pay taxes, and taxes are different from one place to another place. Some countries don't collect taxes while other do. People with different income will pay different taxes.

**GST** GST's relevance can be first easily identified in the POLITICAL BODY domain, i.e., only Canada collects GST. It is also has relevancy to most of the products except food in the PRODUCT domain.

**Pension** Pension's relevance can mainly be identified by INDUSTRY, OCCUPATION, POLITICAL BODY and UNION.

**Interest** Interest is usually the same across INDUSTRIES and COMPANIES, but different in different countries. In different types of ECONOMY, knowledge about interest will be different, too.

**Salary** Salary varies greatly in almost all the dimensions in term of amount and form. In some INDUSTRIES and OCCUPATIONS, salary is more hourly calculated. The amount of salary will also be different across COMPANY, POLITICAL BODY, ECONOMY and PEOPLE.
So far, a new framework has been developed to store both commonsense knowledge and its domain of relevant information. In the next Chapter, we will see how a new VCS is build on the framework developed here.
5. THE NEW VCS

The new VCS was developed aimed at fixing the problems of the original VCS. It incorporates the commonsense knowledge base described in the previous Chapter. Major changes are also done to improve the user interface.

5.1 Architecture of the New VCS

Because both the original VCS and the CBR module are relatively self contained systems, it was decided that some run-time message passing mechanism would be established between the two systems instead of making structural changes to either of them.

Figure 2 shows the structure of the new VCS enhanced with CBR. At present, the control flow is from the VCS to the CBR and the information flow from the CBR back to the VCS. The VCS sends a request to the CBR module requesting commonsense knowledge on one or a group of terms. After the CBR module receives this request, it searches in the commonsense knowledge base and return all the relevant information it finds.

5.2 Consultation of Commonsense During Design Process

![Figure 2 Structure of the new VCS](image)
5.2.1 Stage One

As discussed in Chapter Two, the recognition phase is the most difficult part for a novice user (Batra and Davis, 1992). To support the user in this stage, the commonsense knowledge base stores some of the most important and universal structures in the application domain and their mappings into conceptual data constructs. By accessing the commonsense knowledge base, the user can quickly identify the underlying structure and modify the solutions provided by the system to fit the task’s specific requirements.

Commonsense knowledge can also be used in the enterprise and representation phases by providing relevant information. When the user reads a term from the case or enters an entity or relationship into VCS, he can access the commonsense knowledge base through CBR to get related facts. This will facilitate the communication between the user and the system because it reduces the need to enter trivial information and makes the system look ‘smart’. The user’s attitude towards the system is also likely to be changed. The reasoning process in these two stages is largely based on commonsense knowledge stored.

Commonsense reasoning in this context can be defined as the process of retrieving relevant commonsense knowledge based on a given term. ‘Relevant’ means the knowledge retrieved should be related to the term in the database design context. A term can be the name of either an entity or a relationship. Obviously, the simplest reasoning process in the CBR is retrieving the attributes and keys given entity or relationship names. For example, it is often a tedious task for the user to add the attribute ‘name’ to every ‘person’ related entity. With commonsense reasoning ability, the system can automatically add ‘name’ as an attribute
whenever the user creates such an entity. This reasoning pattern can be incorporated into all other reasoning processes.

All the entities must be related by relationships, and all the relationships involve entities. So the next reasoning process is retrieving relevant relationships for the entities and relevant entities for the relationships. This will further reduce the user input needed. By now the user only needs to enter a name for his entities or relationships. Then the system will search through the commonsense knowledge base and add attributes and related relationships and entities.

5.2.2 Stage Two

In this stage the E-R model is translated into a relational model. Commonsense knowledge plays a less important role here because most of the knowledge used in this stage is database design expert knowledge. However, commonsense knowledge still can help to provide missing attributes and keys and identify multiple values of entities. For example, the user doesn’t need to tell the system that a person cannot have multiple SIN’s.

5.3 Other Improvements

Besides lack of commonsense knowledge, the original VCS also suffers from poor interface design (Gilbert, 1993). Users found it unintuitive and incompatible with other software on the market. To identify interface problems for the original VCS, a similar pilot was done at UBC. The user was trained to use the VCS. All of the users in the testing had difficulties with the interface and some people even could not complete the design process. It was observed that the interface’s effect is so big, it may confound the result of an experiment
on the commonsense knowledge effect. A great amount of work was done to improve the user interface for the VCS.

5.3.1 Direct Input Method
The ‘Entity and Relationship’ box used to be a display-only screen, which means the system displays current entities and relationships in the box but the user can not change anything in the box. It is now converted to a fully functional editing box. This means the user can enter, delete and modify any entity or relationship directly in one box. This saves the trouble of pressing [INSERT] to go to the edit box every time a new entry is entered.

5.3.2 Smart Cursor Positioning
When the user wants to enter a new entity, he first enters the name of the entity in the ‘Entity and Relationship’ box, and then presses [ENTER] to invoke the detailed entity screen to enter attributes and keys for the entity. In the original VCS, when the user got into the entity screen, the cursor stopped at the entity name field, the first field on the screen. The user then had to press [TAB] to go to the attributes field. In the new VCS, when the entity screen is first displayed, the cursor is automatically positioned in the attributes field, and the user can type immediately.

Similarly, when the VCS asks for a missing key from the user, the cursor is now positioned at the key attributes field directly.

5.3.3 Context Sensitive Message
There are three kinds of message in the system. All of them have been made context sensitive.
5.3.3.1 Help Message

It can be accessed by pressing [F1]. This feature was available in the original VCS, but the help message contained only the usage of function keys. The new help message provides explanations of the choices available, depended on the current procedure.

5.3.3.2 System Message

There is always a system message box on the screen. While the user spends most of the time in the entity and relationship screen, the original VCS did not display the attributes, key and mapping ratio of the entries on the screen. The user had to constantly go into the detailed entity or relationship screen to check this information. In the new VCS, the system message box is used to display the complete information about the current cursor entry.

5.3.3.3 Dialogue Message

The dialogue message displayed during the solving session now contains context sensitive information retrieved from the commonsense knowledge base.

5.3.4 Configurable Features

To allow maximum flexibility, all the new features and some of the old features were made configurable through a configuration menu. The user can define whether to consult the commonsense knowledge base, the reasoning level, whether to check for problems after modification. This allows the empirical study of different features of the system to be easily carried out.
6. HYPOTHESIS DEVELOPMENT

6.1 The Research Model

6.1.1 Dependent Variables

Evaluation is an integral part of the system development cycle. As a result, system effectiveness has been an important issue over the years. Effectiveness of an information system is generally defined as the extent to which the system accomplishes its objectives (Hamilton and Chervany, 1981). Therefore, assessing system effectiveness is first to determine the objectives of the system, then to develop measures to evaluate how well the objectives have been achieved. However, different systems tend to have different objectives. As Melone (1990) suggests, "the bases for identifying criteria of effectiveness are many and diverse".

Melone (1990) divides effectiveness into an output-oriented component and an affect-oriented component. The output-oriented component emphasizes system performance in terms of the quantity and quality of outputs and the efficiency of the process. This line of measurement was widely used in system effectiveness studies (Hamilton and Chervany, 1981). User perceptions are now widely used as a surrogate for system effectiveness (Melone, 1990).

In this experiment, the goal is to test the effectiveness of the commonsense knowledge base in the new VCS system. Both output-oriented and affect-oriented measurements are needed to fully evaluate the changes of effectiveness caused by adding CBR. The intended role of the VCS is to support the conceptual design process of the end user. When interacting with the user, the VCS improves productivity by producing better conceptual designs and reducing the user's information processing requirement. The commonsense knowledge
module further enhances VCS's effectiveness by providing commonsense knowledge needed in the design process. Based on this observation, user performance was chosen as the dependent variable. This is supported by past studies on system effectiveness (Gilbert, 1993). To fully measure the effect of the independent variables, the construct of user performance was operationalized by three variables -- the time taken to finish the task, the quality of the design and the perceived ease of use.

6.1.2 Independent Variables

As discussed in Chapter 2, the major constructs in conceptual database design are entities, relationships and attributes. The key task in the design process then is to identify these constructs from the user requirements and map them to the formal modeling language. This is defined as the recognition level in Batra and Davis' process model (1992). Because of the lack of experience of the end user, the design is often error prone and time-consuming. Although many researchers have criticized the lack of commonsense knowledge of existing expert systems (McCarthy and Hayes, 1969; Hobbs and Moore, 1985; Lenat et al., 1986; Goldstein and Storey, 1991) and many empirical studies have been done to investigate user performance using different data models (Hoffer, 1990), no study has ever addressed the role commonsense knowledge plays in the design process. The development of the new VCS with commonsense knowledge represents a concrete step towards more intelligent expert systems. The experiment was designed mainly to test the effectiveness of such effort, so the existence of commonsense knowledge is treated as an independent variable.
The complexity of the design task also has a direct impact on the performance of the
designer (Batra and Antony, 1992a; Gilbert, 1993). Studying the effect of task complexity will
provide valuable information on the generality of an expert system. If the system only
produces good results for simple tasks, then the usefulness of the system in a real world
setting will be seriously limited. On the other hand, if the system performs consistently well
under both easy and difficult tasks in the test, it can be expected to deal with a wide range of
problems. Therefore, the complexity of the task was treated as another independent variable.

However, the relevance of task complexity does not stop here. It can also be a
potential moderating variable for the relationship between the existence of commonsense
knowledge and user performance. The interaction between the adding of commonsense
knowledge and task complexity would be a valuable source of information when designing
future expert systems. If it is proved that commonsense knowledge only helps in more
complex tasks then the system should take this into account and try to consult commonsense
knowledge only when complicated structures are modeled or the size of the model exceeds
certain limit.

The research model used in this study is illustrated in Figure 3.
6.2 Commonsense Knowledge Effect

The commonsense knowledge base incorporated into the VCS contains commonsense knowledge about the application domain expressed in entities, relationships and attributes. By entering these constructs for the user directly instead of letting the user discover them himself, the new VCS facilitates the recognition of not only specific constructs, but also the understanding of the entire problem domain.

Therefore, the following hypotheses are proposed (stated in null form):

**H1** There will be no difference between using the VCS with the commonsense knowledge base and using the VCS without the commonsense knowledge base in terms of time taken to finish the task.

**H2** There will be no difference between using the VCS with the commonsense knowledge base and using the VCS without the commonsense knowledge base in terms of the quality of the design.
H3 There will be no difference between using the VCS with the commonsense knowledge base and using the VCS without the commonsense knowledge base in terms of the perceived ease of use.

It is predicted that all the above hypotheses will be rejected.

6.3 Task Complexity Effect

Traditionally, there are two measurements of the complexity of a design task: the size of the problem and the nature of the relationships among entities (Date, 1990). The size of the problem is usually represented by the number of entities and relationships in the model. The more entities and relationships a model has, the more complex it is. Relationships as an important factor in task complexity is confirmed by the empirical study by Gilbert (1993). It shows that subjects had great difficulties modeling unary and category relationships, while they had little trouble with binary relationships.

This leads to the following hypotheses (stated in null form):

H4 There will be no difference between using the VCS for a complex task and using the VCS for an easy task in terms of time taken to finish the task.

H5 There will be no difference between using the VCS for a complex task and using the VCS for an easy task in terms of the quality of the design.

H6 There will be no difference between using the VCS for a complex task and using the VCS for an easy task in terms of the perceived ease of use.

It is predicted that all the above hypotheses will be rejected.
6.4 Interaction Effect

In the easy task, there are three entities and relationships and all of the relationships are simple binary relationships. In the complex task, there are seven entities and relationships and some are unary and category relationships. As described in Chapter 4, commonsense knowledge is general knowledge about the world. Compared with expert knowledge, it is 'shallow' and 'common'. In designing the experiment, the amount of the commonsense support in the two tasks is made the same, i.e., same number of entities and relationships are prompted to the subject in each task. For the easy task, commonsense knowledge covers a large portion of the design. But for the complex task, commonsense knowledge only helps in the easier portion, leaving difficulties like unary and category relationships completely to the subject.

Therefore, it is expected that when the task becomes more and more complex, commonsense knowledge's effect will become less obvious.

This discussion suggests the following hypothesis (stated in null form):

\[ H_7 \] There will be no interaction effect between the existence of commonsense knowledge and the complexity of the task.

will be rejected.
7. RESEARCH METHODOLOGY

As stated before, the main objective of this study is to test if the VCS’s effectiveness will be affected by the existence of commonsense knowledge or task complexity. However, there are many other features besides these two factors that may affect the user’s performance. Examples are subject’s understanding of the task, system design concepts and familiarity of the VCS. To find the real relationship between the independent variables and dependent variables, other factors’ effects must be ruled out. Because of this focus on causal relationship, a laboratory experiment was selected as the research method (Emory and Cooper, 1991). Benbasat (1989) states ‘high internal validity is one of the major advantages of laboratory experiments’. By using a laboratory experiment, various kind of threats to the internal validity can be reduced or avoided. Also more information can be captured in a controlled manner by the experimenter.

7.1 Experimental Design

A 2x2 factorial design was used with existence of commonsense knowledge and the complexity of the task as the two factors. There are four treatment groups: the group working on a complex task using the VCS with commonsense knowledge; the group working on an easy task using the VCS with commonsense knowledge; the group working on a complex task using the VCS without commonsense knowledge; the group working on an easy task using the VCS without commonsense knowledge. This design allows the effect of commonsense knowledge and the complexity of the task to be examined separately. The interaction between these two factors can also be studied.
Currently, commonsense knowledge support is mainly provided during the creation of the E-R model, which corresponds to the recognition level in Batra and Davis's process model introduced in Section 2.4. This has been identified as the step that is most difficult for a novice designer and commonsense knowledge can possibly play an important role. It is therefore the main focus of this experiment.

7.2 Subjects

44 UBC students volunteered as subjects. With $\alpha=.05$, effect size=.4, this sample size gives a statistic power of .7 (Coven, 1977; Coven and Coven, 1983). The subjects were randomly assigned to one of the four groups. The four groups are of equal size, with 11 subjects in each of them.

Monetary award was given to encourage participation. Extra money was given to the top 5 performers.

Although no formal database design knowledge was required, the subject did need to know how to use the VCS, especially its interface. A tutorial was administered before the experiment to familiarize the subjects with the system.

One of the assumptions of the VCS is that the user may have no formal database design knowledge but be an expert in the problem domain. The tasks used in the experiment were both structured around the academic environment. Therefore, university students are appropriate subjects in this context.
7.3 **Independent Variables**

The two independent variable were the existence of commonsense knowledge in the VCS and the complexity of the task.

7.3.1 **Commonsense Knowledge**

Because the experiment was conducted using university students as subjects and education as the application domain, the commonsense knowledge base used is in education domain as well. Based on the Naive Semantics discussed in Section 4.3, the following Generic Education Model was developed and used as the commonsense knowledge base in the experiment.

Obviously, the facts included in this model are ‘commonsense’ and trivial to most people.

**Entities (Attributes):**

- school (NAME, phone, address),
- course (COURSE_NUMBER, name, credit),
- person (SIN, name, phone, address),
- teacher (),
- professor (),
- student (STUDENT_NUMBER),

**Relationships:**

- teacher is_a person,
- professor is_a teacher,
- student is_a person,
- school employs professor,
- school enrolls student,
- school offers course,
- teacher teaches course,
- student takes course.
Although a domain of relevance knowledge base discussed in Section 4.5 can be built and appended to the above knowledge base, it is not necessary in this experiment because of the focus of the experiment. All the knowledge stored in the commonsense knowledge base is in the education domain and therefore always available to the subject during the design session.

7.3.2 Tasks

Two cases were developed based on the pilot study by Gilbert (1993). The university selection case is the simple one, with only 3 entities and 4 relationships. The student advisory case is more difficult and has 7 entities and 7 relationships (Appendix F).

In the university selection case, the subject was asked to design a database to support students in selecting their universities. All the entities in the design are easily identified and the relationships are simple binary relationships.

The student advisory case required the subject to create a database to store information such as courses and professors for an academic program. One of the difficulties in this task is the category relationship between elective courses, core courses, and courses. The other difficult point lies in the unary relationship between course and its prerequisites.

7.4 Dependent Variables

7.4.1 Time Taken to Finish the Task

This is defined as the time from the moment the subject starts working on the model till the moment he finishes the E-R model.
7.4.2 Design Quality

Design quality can be regarded as how well the conceptual data model captures reality and conforms to the data model syntax. Measuring the design quality is a difficult task. A review of previous literature (Batra and Antony, 1992a, 1992b; Shoval and Even-Chaime, 1987; Batra, et. al., 1990, Gilbert 1993) indicates there is no standard grading scheme for design quality.

It is generally accepted that the key elements of a E-R model are the basic constructs: entities, relationships and attributes (Chen, 1976; Date, 1990), so a grading scheme was designed based on the design’s accuracy in using these constructs (Appendix A). A standard design (Appendix B) was first developed by the researchers and the subjects’ scores were determined by matching entities, relationships and attributes in his design with this standard design. To make the results comparable across tasks, scores were converted to percentage of correctness by dividing each subject’s score by the perfect score.

7.4.3 Perceived Ease of Use

User perception has been increasingly used as a surrogate for system effectiveness (Melone, 1990). Davis (1989) developed and validated two measures of user perceptions: perceived usefulness and perceived ease of use. Moore and Benbasat (1991) further refined these measures and established an instrument to measure the perceptions of adopting an information technology innovation. Because these measures have been widely used by researchers and showed good convergence and reliability, we adopted Moore and Benbasat’s scale in this experiment. However, the perceived usefulness would be hard to measure because most subjects don’t design any database on their job. The validity of the result would be
compromised under this circumstance. Only the perceived ease of use measure was used in this experiment.

7.5 Experimental Procedure and Data Collection

The experiment was conducted in an office with a computer and a video camera. The computer is a 486 IBM-PC compatible with color monitor. The experimenter started the VCS before the subject entered the room. The video camera was used to video tape the computer screen during the design session. The complete design session was video taped except when the tutorial and questionnaires were administrated.

7.5.1 Pre-Test Questionnaire

The subject first filled out a questionnaire on his background, including database design experience and computer experience. This information was used in the data analysis.

7.5.2 Introduction and Tutorial

Before the experiment, the subject read a one and half page introduction to the whole experiment, database design principles and, especially, the VCS system. This was followed by a tutorial. The goal of the tutorial was to show the subject how to apply the design principles and how to use the system. The tutorial also helped to remove the learning effect. A sample library information system was used in the tutorial (Appendix E). The experimenter explained and demonstrated the usage of function keys through a complete design process. The subject was encouraged to ask questions about the use of VCS during the tutorial.

The tutorial took approximately 30 minutes.
7.5.3 Understanding the Test Case Description

In this step, the subject read the case and was allowed to ask the experimenter questions regarding the description in the case or the design requirements. No time limit was specified to allow the subject fully understanding of the case.

7.5.4 Creation of the E-R Model

After the subject indicated he had finished reading the case and had a good understanding of the task required, he was allowed to use the VCS to start to design the database using the E-R model. No questions of any kind were allowed to eliminate human intervention.

The process was video taped and time stamped. The time to finish this step was later calculated based on the video tape.

After the subject finished the task, his design was saved as a file. The design was later graded by the researcher using the pre-defined grading scheme.

7.5.5 Post-Test Questionnaire

After the user finished designing the E-R model, his perceived ease of use of the VCS regarding the creation of the E-R model was measured by a questionnaire.
8. DATA ANALYSIS

The experimental data collected were entered into computer files. There are three dependent variables in this study:

1. Time taken to finish the design task;

2. Quality of the design;

3. User perceived ease of use.

The summary of the results is shown in Table 1 to 3.

<table>
<thead>
<tr>
<th>N</th>
<th>Time</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Commonsense Easy Task</td>
<td>11</td>
<td>8 28 23 13 13 27 47 40 25 27 22</td>
</tr>
<tr>
<td>Commonsense Easy Task</td>
<td>11</td>
<td>17 20 12 21 10 22 9 19 25 18 10</td>
</tr>
<tr>
<td>No Commonsense Difficult Task</td>
<td>11</td>
<td>24 21 39 22 27 52 11 25 30 46</td>
</tr>
<tr>
<td>Commonsense Difficult Task</td>
<td>11</td>
<td>25 14 24 30 13 39 28 7 18 26 37</td>
</tr>
</tbody>
</table>

*Table 1 Time taken to complete task by groups (minutes)*

<table>
<thead>
<tr>
<th>N</th>
<th>Quality</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Commonsense Easy Task</td>
<td>11</td>
<td>78 85 66 85 63 66 29 68 73 76 61</td>
</tr>
<tr>
<td>Commonsense Easy Task</td>
<td>11</td>
<td>76 73 85 61 78 56 93 68 71 83 49</td>
</tr>
<tr>
<td>No Commonsense Difficult Task</td>
<td>11</td>
<td>53 63 66 69 81 56 63 57 51 53 59</td>
</tr>
<tr>
<td>Commonsense Difficult Task</td>
<td>11</td>
<td>87 78 69 62 82 44 76 46 82 79 78</td>
</tr>
</tbody>
</table>

*Table 2 Quality of design by groups (percentage)*

<table>
<thead>
<tr>
<th>N</th>
<th>EOU</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No Commonsense Easy Task</td>
<td>11</td>
<td>28 23 23 7 15 18 13 24 22 8 25</td>
</tr>
<tr>
<td>Commonsense Easy Task</td>
<td>11</td>
<td>23 28 15 23 21 9 25 26 19 26 14</td>
</tr>
<tr>
<td>No Commonsense Difficult Task</td>
<td>11</td>
<td>11 19 8 19 23 18 10 19 16 22 9 16</td>
</tr>
<tr>
<td>Commonsense Difficult Task</td>
<td>11</td>
<td>28 19 20 25 29 18 19 26 21 19 19 22</td>
</tr>
</tbody>
</table>

*Table 3 Perceived ease of use by groups (total score on the EOU questionnaire, ranges from 0 to 35)*
Because the experiment was a multiple dependent variable design, a Multivariate Analysis of Variance (MANOVA) was conducted. The SPSS/PC 4.0’s MANOVA procedure was used with time, design quality and user perceived ease of use as dependent variables. Existence of the commonsense knowledge module and the complexity of the task were the two independent variables. The printout from the MANOVA procedure is included in Appendix C.

8.1 Commonsense Knowledge Effect

8.1.1 Time and Perceived Ease Of Use

There are significant differences between the VCS without commonsense knowledge and the VCS with commonsense knowledge in terms of time taken to finish the task and perceived ease of use (for time: $F(1,40)=5.65, p<0.02$ and for perceived ease of use: $F(1,40)=5.86, p<0.02$). Therefore, H1 and H3 are rejected. This suggests that subjects using the VCS enhanced with the commonsense knowledge module used less time to finish the design and perceived it to be easier to use. This was supported by the discussion in Chapter 6 where the hypothesis is developed.

8.1.2 Design Quality

H2 is not rejected ($F(1,40)=2.07, p<0.16$). This shows adding the commonsense knowledge base did not improve the design quality significantly.

The content of the commonsense knowledge base provides a possible explanation why the design quality is not improved. The commonsense knowledge base used in the experiment only contains a minimal set of facts. This was intended to avoid slanting the results in favor of
commonsense reasoning. This approach, however, can also weaken the effect of commonsense knowledge base.

Another possible explanation for this may be found in the way commonsense knowledge is retrieved in the current system. When the commonsense knowledge module receives a term that the VCS wants to find more about, it only does a simple string search on the text of the term. If the term appears in a fact in the commonsense knowledge base, that fact is regarded as relevant to the term and returned to the VCS. For example, if the VCS wants to find out all the commonsense knowledge about ‘students’, it will pass the term ‘student’ to the commonsense knowledge module. The commonsense knowledge module then finds all the facts it knows where this term appears. If the user enters a misspelled word or a synonym not stored in the commonsense knowledge base, the system will not be able to recognize it and will not retrieve any information. In this case, the system can not be of any help to the user.

Another reason is that some subjects accepted whatever was suggested by the commonsense knowledge base without judging its relevance. The commonsense knowledge retrieved in this experiment is always relevant to the education domain, but not always relevant to the specific case. Sometimes, the entities or relationships retrieved were even incorrect from the case’s point of view. The user, as the designer, has to decide what is needed and what is not. The VCS as an expert system can help the user in the process, but can not replace the user. If the user relies completely on the system and its commonsense
DATA ANALYSIS

knowledge, then some irrelevant or even incorrect entities, relationships and attributes will be included.

This was clearly shown in the number of extra entities or relationships entered by the subjects during the experiment (Figure 4). Subjects using the VCS with the commonsense knowledge module entered more extra relationships than their counterparts using the VCS without the commonsense knowledge module. However, the number of extra entities was the same for the two group. The way most people work with the E-R model helps to shed some light on this contradiction. It was observed in the experiment that most people started the model with entities. They only moved to the relationships after they had finished entering all the entities. For each user entered entity, the VCS’s commonsense reasoning module could retrieve related relationships and add them to the design automatically. For each user entered relationship, the VCS only added the two entities mentioned in the relationship if they did not exist. Because most subjects started with entities, only extra relationships were added by the commonsense knowledge module. It was very rare when a subject entered a relationship before he entered the two entities.
8.2 Task Complexity Effect

8.2.1 Time

Statistical analysis shows task complexity has an effect on the time required to finish the task (F(1,40)=4.06, p<0.05). Therefore, H4 is rejected. This means the subjects used more time on the complex task, which is expected.

8.2.2 Design Quality

The effect of task complexity on design quality is not significant (F(1,40)=0.29, p<0.47). Therefore H5 is not rejected.

This was a surprise because the complex task contained not only more constructs but also more difficult relationships. Novice designers usually have great difficulties modeling the
DATA ANALYSIS

unary and category relationships as found by Gilbert (1993) and confirmed in the pilot testing at UBC. They have trouble both identifying and representing these constructs.

A closer look at the experimental procedure helped to explain the result. In the tutorial given at the beginning of the experiment, the concepts of unary and category relationships and the techniques to model them were introduced. The subjects had a fresh memory about this knowledge when they started the experiment. They often looked actively in the experiment task for unary and category relationships because they thought what was taught in the tutorial must be used somewhere in the experiment. Under this circumstance, it was relatively easy for them to recognize these relationships and apply the techniques they just learned. This helped the subjects given the complex case to perform better than they otherwise might have. In real design tasks, the identification and modeling of unary and category relationships depend on many more factors than that those can be taught in a 30 minute tutorial, and the performance of novice designers may not be as good as in the experiment.

8.2.3 Perceived Ease Of Use

The effect on perceived ease of use is not significant ($F(1,40)=0.22, p<0.64$). Therefore, H6 is not rejected which means the subjects perceived the system to be equally easy to use regardless of the task.

One explanation of this result is that the subjects were asked questions regarding the system, not the experiment task, so the complexity of the task did not affect their impressions. When the subjects evaluated the system, they very likely considered other aspects of the
system like the user interface which was exactly the same for all groups. The strong effect of
the interface might dominate the subjects' perceived ease of use of the system.

8.3 Interaction Effect

There is only one possible interaction effect, the interaction between the existence of
commonsense knowledge base and the complexity of the task in this study. Statistical analysis
shows this interaction effect is not significant (for time: $F(1,40)=0.11$, $p<0.74$; for perceived
ease of use: $F(1,40)=1.47$, $p<0.23$; for quality of design: $F(1,40)=0.52$, $p<0.47$).

As discussed in Chapter 6, commonsense knowledge is only 'shallow' knowledge
about the world, so it is not as effective in a complex task as in a simple task. However, even
though commonsense can not provide a large percentage of the knowledge used in the final
design, it can be used as a framework by some users to construct their own model. Entities
and relationships added by the system, although simple, may inspire the user to think about the
task from new directions and discover other structures. In that sense, commonsense can help
even in complex tasks.
9. CONCLUSION AND FUTURE RESEARCH

In this study, a new database design expert system enhanced with commonsense knowledge was developed and the effectiveness of the commonsense knowledge module was tested using a laboratory experiment. An ontology was also proposed to store domain of relevance information for the commonsense knowledge. One of the major contributions of this experiment is that it showed that even a minimal amount of commonsense knowledge improves the time taken to finish a design task and the perceived ease of use. The establishment of the causal relationship between commonsense knowledge and improved design process can be used as the basis for further research into other aspects of commonsense reasoning support.

The results of this study suggest that commonsense knowledge can play an important supportive role in the database design process. The commonsense knowledge provided to the user, however, should be relevant to the application domain and design task. The domain of relevancy information can be used to select facts from the commonsense knowledge base. Commonsense knowledge’s effect is apparent in both simple and complex situations. This shows that commonsense knowledge will be an important help in various design tasks in the real world.

9.1 Limitation of the Study

The structure of commonsense knowledge in the commonsense knowledge base and the retrieval process are still in the prototype stage, thus limiting the effectiveness of the commonsense knowledge module.
Although a domain of relevance ontology was created, it was not tested in the experiment because of the single problem domain. The Generic Business Model (Goldstein and Storey 1991) was not implemented for the same reason.

9.2 Future Research

Several possible research directions come from this study. After the causal relationship between commonsense knowledge and system effectiveness has been established, protocol analysis should be done to analyze the dynamics of the design process. By doing so, the exact role played by commonsense knowledge can be identified. This should provide guidance to further development of an intelligent database design expert system which uses commonsense knowledge to the best extent.

The content of the commonsense knowledge base should be further refined. For the purpose of the experiment, a very small amount of knowledge was put into the system. In the future, more commonsense facts and their domain of relevance information should be identified and stored. An experiment with an expanded commonsense knowledge base will allow researchers to fully study the interaction between the VCS and the commonsense knowledge module.

The current commonsense knowledge module uses simple string search to retrieve related commonsense facts. More reasoning patterns should be developed to provide more relevant commonsense knowledge support.

Learning ability will enable the system to learn new commonsense knowledge as it is used by different users. The system should be able to reorganize its commonsense knowledge
CONCLUSION AND FUTURE RESEARCH

ontology and domain of relevance ontology when knowledge is accumulated. This will simplify the maintenance of the commonsense knowledge base.
10. BIBLIOGRAPHY


BIBLIOGRAPHY


Meltzer, B. Knowledge and Experience in Artificial Intelligence. AI Magazine. Spring 1985, 6:1, 40-42.


Appendix A  Grading Scheme for Design Quality

<table>
<thead>
<tr>
<th>ITEM</th>
<th>POINTS ASSIGNED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Entities</td>
<td>2</td>
</tr>
<tr>
<td>Correct Attribute</td>
<td>1</td>
</tr>
<tr>
<td>Correct Primary Key</td>
<td>1</td>
</tr>
<tr>
<td>Extra Attribute</td>
<td>0</td>
</tr>
<tr>
<td>Incorrect Entities</td>
<td>0</td>
</tr>
<tr>
<td>Correct Relationships</td>
<td>2</td>
</tr>
<tr>
<td>Correct Cardinality One</td>
<td>1</td>
</tr>
<tr>
<td>Correct Cardinality Two</td>
<td>1</td>
</tr>
<tr>
<td>Correct Attribute</td>
<td>1</td>
</tr>
<tr>
<td>Incorrect Relationship</td>
<td>0</td>
</tr>
</tbody>
</table>
Appendix B  Standard Design

Standard E-R design for the advisory case

ENTITIES

Course      [COURSE_NUMBER, description, Section_per_year]
Core_Course [waiver_allowed]
Elective_Course [enrollment_limit]
Section     [SECTION_NUMBER, time, location, no_student]
Student     [STUDENT_NUMBER, name, address, phone, gpa, major]
Professor   [NAME, OFFICE_NUMBER, area, rank]
Pre_requisite []

RELATIONSHIPS

Core_Course(1,1)  is_a  Course(0,1)
Elective_Course(1,1)  is_a  Course(0,1)
Pre_requisite(1,1)  is_a  Course(0,1)
Student(0,n)        takes  Course(1,n) [grade]
Professor(1,n)      teaches  Course(1,n) [Section_number]
course(0,n)         has    Pre_requisite(0,n) [must_take_before]
Course(1,n)         has    Section(1,1)
Standard E-R design for the university selection case

ENTITIES
University [NAME, address, phone, tuition]
Program [NAME, credit_hours, year_started, gmat_required]
Alumni [NAME, ADDRESS, phone, year_graduation, major]
Professor [NAME, ADDRESS, phone, area]

RELATIONSHIPS
University(1,n) has Program(1,1)
Program(1,n) has Professor(1,n)
Program(1,n) has Alumni(1,1)
Standard E-R design for the tutorial case

ENTITIES

collection  [CALL_NUMBER, title, status]
book        [author]
serial      [issue_number]
reference   []
borrower    [CARD_NUMBER, name, phone, address]

RELATIONSHIPS

book(1,1)   is_a  collection(0,1)
serial(1,1)  is_a  collection(0,1)
reference(1,1)  is_a  book(0,1)
book(0,n)    has   reference(0,n)
borrower(0,n)  borrows  book(0,1)  [due_date]
Appendix C  Statistical Results

* * ANALYSIS OF VARIANCE -- DESIGN 1 * *

EFFECT .. TASK
Multivariate Tests of Significance ($S = 1, M = 1/2, N = 18$)

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Value</th>
<th>Approx. F</th>
<th>Hypoth. DF</th>
<th>Error DF</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pillais</td>
<td>.12233</td>
<td>1.76546</td>
<td>3.00</td>
<td>38.00</td>
<td>.170</td>
</tr>
<tr>
<td>Hotellings</td>
<td>.13938</td>
<td>1.76546</td>
<td>3.00</td>
<td>38.00</td>
<td>.170</td>
</tr>
<tr>
<td>Wilks</td>
<td>.87767</td>
<td>1.76546</td>
<td>3.00</td>
<td>38.00</td>
<td>.170</td>
</tr>
<tr>
<td>Roys</td>
<td>.12233</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Univariate F-tests with (1,40) D. F.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Hypoth. SS</th>
<th>Error SS</th>
<th>Hypoth. MS</th>
<th>Error MS</th>
<th>F</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>408.09091</td>
<td>4017.27273</td>
<td>100.43182</td>
<td>.06336</td>
<td>.51</td>
<td></td>
</tr>
<tr>
<td>EOU</td>
<td>7.36364</td>
<td>1312.36364</td>
<td>7.36364</td>
<td>.22444</td>
<td>.638</td>
<td></td>
</tr>
<tr>
<td>QUALITY</td>
<td>79.11364</td>
<td>10809.2727</td>
<td>79.11364</td>
<td>.29276</td>
<td>.591</td>
<td></td>
</tr>
</tbody>
</table>
**ANALYSIS OF VARIANCE -- DESIGN 1**

**EFFECT : COMMON**

Multivariate Tests of Significance \((S = 1, M = 1/2, N = 18 )\)

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Value</th>
<th>Approx. F</th>
<th>Hypoth. DF</th>
<th>Error DF</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pillais</td>
<td>.18792</td>
<td>2.93118</td>
<td>3.00</td>
<td>38.00</td>
<td>.046</td>
</tr>
<tr>
<td>Hotellings</td>
<td>.23141</td>
<td>2.93118</td>
<td>3.00</td>
<td>38.00</td>
<td>.046</td>
</tr>
<tr>
<td>Wilks</td>
<td>.81208</td>
<td>2.93118</td>
<td>3.00</td>
<td>38.00</td>
<td>.046</td>
</tr>
<tr>
<td>Roys</td>
<td>.18792</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Univariate F-tests with \((1,40) \) D. F.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Hypoth. SS</th>
<th>Error SS</th>
<th>Hypoth. MS</th>
<th>Error MS</th>
<th>F</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>567.36364</td>
<td>4017.27273</td>
<td>567.36364</td>
<td>100.43182</td>
<td>5.64924</td>
<td>.022</td>
</tr>
<tr>
<td>EOU</td>
<td>192.36364</td>
<td>1312.36364</td>
<td>192.36364</td>
<td>32.80909</td>
<td>5.86312</td>
<td>.020</td>
</tr>
<tr>
<td>QUALITY</td>
<td>560.20455</td>
<td>10809.2727</td>
<td>560.20455</td>
<td>270.23182</td>
<td>2.07305</td>
<td>.158</td>
</tr>
</tbody>
</table>
ANALYSIS OF VARIANCE -- DESIGN 1 *

EFFECT .. COMMON BY TASK
Multivariate Tests of Significance (S = 1, M = 1/2, N = 18)

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Value</th>
<th>Approx. F</th>
<th>Hypoth. DF</th>
<th>Error DF</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pillais</td>
<td>.05656</td>
<td>.75944</td>
<td>3.00</td>
<td>38.00</td>
<td>.524</td>
</tr>
<tr>
<td>Hotellings</td>
<td>.05946</td>
<td>.75944</td>
<td>3.00</td>
<td>38.00</td>
<td>.524</td>
</tr>
<tr>
<td>Wilks</td>
<td>.94344</td>
<td>.75944</td>
<td>3.00</td>
<td>38.00</td>
<td>.524</td>
</tr>
<tr>
<td>Roys</td>
<td>.05656</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Univariate F-tests with (1,40) D. F.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Hypoth. SS</th>
<th>Error SS</th>
<th>Hypoth. MS</th>
<th>Error MS</th>
<th>F</th>
<th>Sig. of F</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>11.00000</td>
<td>4017.27273</td>
<td>11.00000</td>
<td>100.43182</td>
<td>.10953</td>
<td>.742</td>
</tr>
<tr>
<td>EOU</td>
<td>48.09091</td>
<td>1312.36364</td>
<td>48.09091</td>
<td>32.80909</td>
<td>1.46578</td>
<td>.233</td>
</tr>
<tr>
<td>QUALITY</td>
<td>141.84091</td>
<td>10809.2727</td>
<td>141.84091</td>
<td>270.23182</td>
<td>.52489</td>
<td>.473</td>
</tr>
</tbody>
</table>
Appendix D  Tutorial

THE UNIVERSITY OF BRITISH COLUMBIA
The Faculty of Commerce and Business Administration
Management Information Systems Division

Database Design Expert System Study

TUTORIAL -- ONE
DATABASE DESIGN BASICS

Database design involves describing the structure of the portion of the real world that you want to have information about in your database. We design a database by defining the entities, relationships and attributes that are to be included. This is not very different from describing something in ordinary English:

**Entities** are like nouns—they refer to persons, places, or things we want to have represented in the database. In deciding what entities to have in your database design, think about what things you are going to want information about. For example, the entities for a library database would surely include book and borrower, and might also include branch, author, subject, etc., depending on exactly what purposes the database was intended to serve.

**Relationships** correspond to verbs in ordinary language. They represent some kind of connection between entities. The most common kind of relationship connects two entities, as in "borrower borrows book". Borrows is the verb in this phrase and it states a relationship between the two entities, book and borrower.

Although relationships such as the one just shown are the most common, it is possible to have a relationship that involves only one kind of entity or more than two kinds. For example, "book references book" expresses the fact that one book can reference another. It is still a relationship between two things, but in this case, they are both of the same kind. Similarly, "borrower borrows book from branch" is a relationship among three entities. This is fairly rare in practice.

One of the characteristics of a relationship which must be specified is referred to as its "mapping ratios" or "cardinalities". This describes the number of instances of the relationship a single instance of that entity can participate in. It is not necessary to know the exact number—just whether it is "at most one", "at least one", "exactly one" or "any number". As an
example, consider the relationship "borrower borrows book". A single borrower can participate in any number of "borrow" relationships, but a

single book can participate in at most one since if it has been borrowed by one person, it can't be borrowed by anyone else at the same time. Mapping ratios are sometimes expressed as a pair of numbers in parentheses. The first number is the minimum and the second, the maximum. In this notation, we usually use "n" to mean any number greater than 1. In this notation, the mapping ratios for borrower would be (0,n) and those for book would be (0,1).

A particularly useful type of relationship is "is_a", as in "novel is_a book". The meaning of this relationship is that a novel is a kind of book. Therefore, anything that the database system knows about books in general must also apply to novels.

Attributes in a database design are used to describe entities and relationships something like adjectives and adverbs in English. Attributes of an entity (noun) describe properties of that kind of thing. For example, the Borrower entity might have attributes such as card_number, name, address, no._of_overdue_books, etc. We should include in a database design only those attributes that are necessary for the task. Every entity will have some attributes specified for it or there is no point in having it in the database at all!

A key is a special kind of attribute. It refers to an attribute (or combination of attributes) that can be used to uniquely identify a particular entity. For example, "card_number" is a key for the Borrower entity, since no two borrowers should ever have the same card number. "Name" would not be a good key for Borrower because it is possible to have two borrowers with exactly the same name. In such a case we could use a combination of attributes as a key, such as "name, address".

Relationships can also have attributes. The relationship "borrower borrows book" might have the attribute "date_due". Not all relationships have attributes.
**APPENDIX**

**TUTORIAL – TWO**

**USING THE VIEW CREATION SYSTEM (VCS)**

**IMPORTANT:** There is a prompt line at the bottom of screen at all time to show you all the keys available to you at that time. Please read them carefully. You can also access the system’s context sensitive help by pressing F1 at any time. The experimenter will not answer questions during the experiment.

---

**Stage One**

In this stage, you will use the VCS to create an Entity-Relationship model for your database. Please read the Task Description carefully. Your design should include all and only information mentioned in the Task Description.

1. Enter entities, relationships and attributes: Type on the screen directly. After each line, press [\[\[\]\]] to go to the next line. Press [ENTER] on a line to enter attributes, keys and mapping ratios.
   - Entity and attribute names can be only one word. Relationship will be entered in the form ‘entity_one relationship entity_two’, as in ‘customer buys product’.
   - All names must be in lower case. All the entity names should be in singular form. i.e. use ‘car’ instead ‘cars’ in all places.
   - Use a single word for name. If you want to use multiple words, join them with an underline (not dash). For example, ‘insurance_num’ for ‘insurance number’.

2. After you are satisfied with your design, please notify the experimenter.

**Stage Two**

In this stage, you will use the VCS to modify your original Entity-Relationship model and translate it into a Relational model. The VCS will ask you some questions regarding your design. You should always choose ‘Continue’ when presented with the choice ‘Continue/Go back to the main menu’. As in Stage One, you have access to the prompt line and context sensitive help. Read them carefully before making your choice.

1. Enter missing information: The most common case is missing key attribute. When the VCS finds an entity without a key attribute, it will alert you and enter the entity screen and place the cursor at the first key attribute field. To add a key attribute, press [Insert]. Use the [Space Bar] to identify key attribute(s) in the pop-up window. Then press [F10] to exit.

2. Answer other questions from the VCS. For example, clarifying some words’ meaning, identifying multiple value, functional dependency. The terms will be explained in the ‘System Message’ and Help.

3. After the VCS solves all the problems and returns to the main menu, please notify the experimenter.
Appendix E  Tutorial Task

TUTORIAL — THREE

CASE DESCRIPTION

You were asked by your community library’s director to develop a database for her library. After talking to her, you found out that the library has two categories of holdings: books and serials. Some information, like title, call number, status, should be kept for both kinds of collections. However, only books have author and only serials have issue number. Some books also have references which are also books.

Besides book, you were told to store information on borrowers as well. This includes name, card number, phone and address.

Finally, the director required that she be able to check who borrowed which books (only books can be borrowed) and the due date, so the database has to store this information as well.

You were told the database you design should be able to answer questions like the following:

1. What is the status of a particular book?

2. Which books are borrowed by a particular borrower and when is the due date?

3. What is the reference book(s) for a particular book?
Appendix F  Experiment Tasks

THE UNIVERSITY OF BRITISH COLUMBIA
The Faculty of Commerce and Business Administration
Management Information Systems Division

Database Design Expert System Study

TASK DESCRIPTION

As a major in Entrepreneurship, you want to set up a one-person consulting practice whereby you will provide information to students who are selecting business programs for their graduate studies. In order to do so you will need a database with information on universities and their business programs.

First you would like to store some information for each university, for example, name, address, phone, and tuition. Each university offers various types of degrees including MBA, M.Sc. and Ph.D. The number of credit hours required is different for each program. The number of credit hours required can vary from one university to another, too. Some programs require GMAT scores for admission while others do not. It is important to identify whether a GMAT score is required. Students would also be interested in the year a university started a program.

Universities have a number of professors who may teach in more than one programs. Some potential students will request information on certain professors so that they can contact them directly. Thus you will need to retain information on a professor that consists of name, address, phone, and area (for example, finance, marketing, etc.)
Recently, universities have started to engage alumni who are willing to talk to prospective students in the same program. Universities will provide you with a list of each program's alumni contacts which includes their name, address, phone, year of graduation and their major.

Questions:

The following is a list of questions that the database should be able to provide answers for:

1. What is the tuition at a particular university?
2. What program(s) does a particular university offer (for example, MBA, M.Sc., etc.)?
3. How long has a particular program at a particular university been established?
4. Who is teaching at a particular university for a particular major?
5. Who graduated from a particular program at a particular university?
TASK DESCRIPTION

You have been assigned the task of developing a database for a student advisory function. The purpose of the database is to provide students with information on the courses for their program, the pre-requisites for each of their courses; whether or not they are eligible for waiver exams; information on course offerings, etc.

Courses may be core courses for some programs or elective courses. Obviously some information you will need will be the same for both types of courses. For example, each course should have a course number and a brief description and there should be an indication of the number of times it is offered throughout the year. The main difference between the two types of courses is that elective courses have enrollment limits and core courses may allow waiver examinations.

Each course has one or more Sections. For those courses that have only one Section, the Section number will always be ‘001’. When a course has multiple Sections, each Section need not be taught by the same professor (although this is possible of course).

Courses can have one or more pre-requisites and it is necessary to retain information on what the pre-requisites are. A pre-requisite is usually taken before the course that requires it, although some may be taken concurrently with the higher level course.
For each course offering the database should keep track of which professor is teaching which Section, plus the time and location (room number) where each Section meets.

Assume that a student has just one major area (for example, accounting, marketing, etc.) which should be stored in the database along with the student’s name, student number, address, phone number, and grade point average. For every course a student takes, his or her mark should be retained.

For each professor, information is needed on the professor’s name, area of expertise (for example, accounting, marketing, etc.), office number and rank (assistant, associate, full).

Questions

The following is a list of questions that the database should be able to provide answers for:

1. What is the description of a particular course?
2. What are the pre-requisites for a particular course?
3. What grades did a student receive in his or her courses?
4. Who is teaching a particular Section of a given course?
5. How many students were enrolled in a particular Section of a course?
6. What is the seating limit for a particular elective course?
QUESTIONNAIRE

Please circle the appropriate words

Using the VCS enhances my effectiveness on the job.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Using the VCS enables me to accomplish tasks more quickly.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

I believe that the VCS is cumbersome to use.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Using the VCS improves my job performance.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Learning to operate the VCS is easy for me.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Overall, I find using the VCS to be advantageous in my job.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Using the VCS gives me greater control over my work.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

My interaction with the VCS is clear and understandable.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

Using the VCS increases my productivity.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

My using the VCS requires a lot of mental effort.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely

It is easy for me to remember how to perform tasks using the VCS.
Likely: extremely | quite | slightly | neither | slightly | quite | extremely | unlikely
Using the VCS is often frustrating.

Likely | extremely | quite | slightly | neither | slightly | quite | extremely |
UNLIKELY

I believe that it is easy to get the VCS to do what I want it to do.

Likely | extremely | quite | slightly | neither | slightly | quite | extremely |
UNLIKELY

Using the VCS improves the quality of work I do.

Likely | extremely | quite | slightly | neither | slightly | quite | extremely |
UNLIKELY

Using the VCS makes it easier to do my job.

Likely | extremely | quite | slightly | neither | slightly | quite | extremely |
UNLIKELY

Overall, I believe that the VCS is easy to use.

Likely | extremely | quite | slightly | neither | slightly | quite | extremely |
UNLIKELY