



**UNIVERSITI PUTRA MALAYSIA**

**TRANSLATING RELATIONAL CONCEPTUAL SCHEMA  
TO OBJECT-ORIENTED SCHEMA**

**SOON LAY KI**

**FSKTM 2001 8**

**TRANSLATING RELATIONAL CONCEPTUAL SCHEMA  
TO OBJECT-ORIENTED SCHEMA**

**By  
SOON LAY KI**

**Thesis Submitted in Fulfilment of the Requirement for the Degree of  
Master of Science in Faculty of Computer Science and Information Technology  
Universiti Putra Malaysia**

**December 2001**



## DEDICATION

*To my parents,  
Lay Khim & Tat Kwang, Lay Im & Tat Mun,  
Sea Yeow, Woon Yar and Kee Leong.*

*... Your loves and supports are my greatest inspiration.*

Abstract of the thesis submitted to the Senate of Universiti Putra Malaysia in  
fulfilment of the requirement for the degree of Master of Science

**TRANSLATING RELATIONAL CONCEPTUAL SCHEMA  
TO OBJECT-ORIENTED SCHEMA**

By

**SOON LAY KI**

**December 2001**

**Chairman : Dr. Hamidah Ibrahim**

**Faculty : Computer Science and Information Technology**

A multidatabase is a confederation of preexisting distributed, heterogeneous, and autonomous database system. The integration process is essential in the effort of forming a distributed, heterogeneous database system. This process generally consists of two main phases, which are conceptual schema translation phase and followed by the integration phase. In our research, we have proposed an alternative translation approach to convert relational database schema to object-oriented database schema.

The translation approach consists of a set of translation rules, which are based on inclusion dependencies, key attributes and types of attributes. A database schema translation tool prototype, called RETOO (Relational-to-Object-Oriented) is then developed based on the proposed translation approach. RETOO receives a relational database schema as input data and generate an object-oriented database schema as the output data.

RETOO operates semi-automatically, especially in the process of identifying operations for each class. This is because relational data model does not provide the behavioural information of every entity.

The translation approach and RETOO database translation tool prototype are not only able to maintain the semantics of the relational database schema, but also enhance the semantics of the translated object-oriented schema via object-oriented data modelling concepts.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia untuk memenuhi keperluan ijazah Master Sains

**PENTERJEMAHAN SKEMA KONSEPSI PANGKALAN DATA HUBUNGAN  
KEPADA SKEMA BERORIENTASIKAN OBJEK**

Oleh

**SOON LAY KI**

**Disember 2001**

**Pengerusi : Dr. Hamidah Ibrahim**

**Fakulti : Sains Komputer dan Teknologi Maklumat**

Pangkalan data pelbagai jenis merupakan gabungan sistem pangkalan data sedia ada yang teragih serta berupaya untuk beroperasi secara automatik. Sesungguhnya, proses integrasi diperlukan untuk membentuk satu sistem pangkalan data yang teragih dan pelbagai jenis. Proses ini biasanya terdiri daripada dua fasa, iaitu fasa penterjemahan skema konsepsi, diikuti dengan fasa integrasi.

Dalam projek ini, kami telah mencadangkan satu pendekatan penterjemahan untuk menterjemahkan skema pangkalan data hubungan kepada skema pangkalan data berorientasikan objek. Syarat-syarat penterjemahan yang digunakan adalah berpandukan kepada kebergantungan lingkungan, atribut kunci dan jenis atribut-atribut kunci tersebut. Sejurus itu, satu sistem penterjemahan skema konsepsi pangkalan data hubungan kepada yang berorientasikan objek (RETOO) telah dibangunkan. RETOO ini akan menerima skema konsepsi pangkalan data hubungan sebagai data input dan menghasilkan data output di dalam bentuk skema konsepsi pangkalan data berorientasikan objek.

RETOO beroperasi secara separa-automatik, terutamanya dalam proses mengenal pasti operasi untuk setiap kelas. Hal ini demikian memandangkan maklumat tentang operasi sesuatu objek tidak wujud dalam model pangkalan data hubungan.

Pendekatan penterjemahan dan sistem penterjemahan kami bukan sahaja berupaya untuk mengekalkan semantik atau ciri-ciri skema pangkalan data hubungan, malah juga berupaya menambahkan semantik skema yang diterjemahkan dengan konsep and ciri-ciri model berorientasikan objek.

## ACKNOWLEDGEMENTS

First and foremost, I would like to express my most sincere gratitude to my supervisors, Dr. Hamidah Ibrahim, Associate Professor Dr. Ali Mamat and Dr. Pua Chai Seng. The invaluable knowledge, experiences and patience they have given to me cannot be measured in words. Neither can the appreciation I have for them.

My heartiest appreciation is expressed to my parents, sisters, brothers, Kee Leong and Babbu, for giving me so much loves and supports throughout the duration of this project. I would also like to extend my indebtedness to several people who have directly or indirectly contributed to this dissertation work. The following list, by no means exhaustive, is an attempt to acknowledge at least a few of these people.

To all lecturers, staffs and  
labmates in the department:

*Puan Lilly Suriani  
Kak Yati  
Kak Umu  
Kak Iza  
Kak Maya  
Shamala  
Umi Zurina  
Maslinda  
Maizatul  
Ika  
Auntie Thevanan*

My housemates:

*Lee Ai  
Hui Lian  
Wee Peng  
Hung Chee  
Seok Har  
Sook Leng  
Pei Ten  
Sing Yi  
Lay Fung  
Lian Pey  
Geok Hun*

My friends:

*Brother Tan Ho Soon  
Residents of Buddhist Houses  
Lei Won  
Poh Cheang  
Boo Ping and Hui Khim  
Shau Meng  
Eddie Chow  
Stephen Ho  
Chong Fang  
Hui Ling*

I share the joy and happiness with all of you and may all of you be well and happy always.



# TABLE OF CONTENTS

	<b>Page</b>
DEDICATION	ii
ABSTRACT	iii
ABSTRAK	v
ACKNOWLEDGEMENTS	vii
APPROVAL SHEETS	viii
DECLARATION FORM	x
LIST OF TABLES	xiv
LIST OF FIGURES	xv
LIST OF ABBREVIATIONS	xvi
 <b>CHAPTER</b>	
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement	3
1.2 Problem Solving	4
1.3 Objectives	5
1.4 Research Scope	5
 <b>2 PRELIMINARIES</b>	<b>6</b>
2.1 Distributed Database	6
2.2 ANSI/SPARC Architecture	10
2.3 Distributed Database Reference Architecture	11
2.4 Distributed Multidatabases Architecture	13
2.4.1 Models with a GCS	13
2.4.2 Models without a GCS	14
2.5 Alternative Design Strategies	15
2.6 Database Integration	18
2.6.1 Schema Translation	18
2.6.2 Schema Integration	19
2.7 Relational Database Conceptual Schema to Object-Oriented Database Conceptual Schema	20
2.8 Relational Database	23
2.8.1 Relational Model Concepts	23
2.8.2 Relational Model Constraints	25
2.9 Object-Oriented Database Model	28
2.9.1 Object-Oriented Concepts	28
2.9.2 Identifying Classes or Objects	31
2.9.3 Identifying the Operations	31
2.9.4 Object-Oriented Abstraction	33



<b>3</b>	<b>LITERATURE REVIEW</b>	<b>36</b>
3.1	Semantic Enrichment of Database Schemas: An Object Oriented Approach	36
3.2	Database Transformation from Relational to Object-Oriented Database and Corresponding Query Translation	40
3.2.1	Introduction of the Approach	40
3.2.2	Schema Transformation	40
3.3	Converting Relational to Object-Oriented Databases	42
3.3.1	Introduction of the Approach	42
3.3.2	Methodology for Data Conversion from Relational to Object-Oriented	43
3.4	Automatic Migration and Wrapping of Database Applications – A Schema Transformation Approach	47
3.4.1	Introduction of the Approach	47
3.4.2	Examples of ER Schemas and ER Transformation	49
3.5	A Data Dictionary System Approach for Database Schema Translation	50
3.5.1	Introduction of the Approach	51
3.5.2	Translation Mechanism	51
<b>4</b>	<b>RETOO TRANSLATION APPROACH</b>	<b>54</b>
4.1	Format of Relational and Object-Oriented Database Conceptual Schema	55
4.2	RETOO Translation Rules	58
4.3	Inclusion Dependency and Translation Rules	58
4.4	Relational to Object-Oriented Database Schema Translation Approach	60
4.4.1	Identifying Classes	61
4.4.2	Identifying the Operations	82
<b>5</b>	<b>TRANSLATION TOOL PROTOTYPE</b>	<b>83</b>
5.1	Relational Database Conceptual Schema Reading Module	85
5.2	Referential Integrity Constraints Reading Module	86
5.3	Composite Attributes Reading Module	87
5.4	Conceptual Schema Translation Module	88
5.5	Methods or Operations Generation and Reading Module	89

<b>6</b>	<b>RESULTS AND DISCUSSION</b>	<b>91</b>
6.1	Translating Relational to Object-Oriented Database Conceptual Schema	91
6.2	Translated OO Schemas Comparisons	105
6.3	RETOO System Implementation	111
6.3.1	Relational Database Conceptual Schema Reading Module	111
6.3.2	Referential Integrity Constraints Reading Module	109
6.3.3	Composite Attributes Reading Module	114
6.3.4	Conceptual Schema Translation Module	115
6.3.5	Methods or Operations Generation and Reading Module	117
6.4	Limitations	118
6.5	Significances of RETOO and the Translation Approach	119
<b>7</b>	<b>CONCLUSIONS AND FUTURE WORKS</b>	<b>120</b>
7.1	Conclusions	120
7.2	The Contributions of RETOO and the Translation Approach	121
7.3	Future Works	122
	<b>REFERENCES</b>	<b>123</b>
	<b>APPENDICES</b>	<b>127</b>
A1	Translating <i>Airline</i> Relational Schema into Object-Oriented Schema	128
	<b>BIODATA OF THE AUTHOR</b>	<b>133</b>

## LIST OF TABLES

Table		Page
2.1	A Relation in a Relational Database	23
4.1	Foreign Key	71
4.2	Subcategories of Case 1	72
6.1	Categories in Translation Step 4	96

## LIST OF FIGURES

Figure		Page
1.1	Relational- <i>to</i> -Object-Oriented Database Schema Translation Tool	2
2.1	A Centralized Database Accessed Remotely	7
2.2	Distributed Database System	9
2.3	The ANSI/SPARC Architecture	10
2.4	Distributed Database Reference Architecture	12
2.5	MDBS Architecture with a GCS	13
2.6	MDBS Architecture without a GCS	15
2.7	Top-Down Design Approach	17
2.8	Database Integration Process	19
3.1	Data Abstraction of BLOOM Data Model	37
3.2	Steps in Knowledge Acquisition Phase	39
3.3	Steps of the Conversion Phase	39
3.4	An Example of Schema Transformation	41
3.5	Examples of ER Schemas Transformation	49
4.1	Translating Relational into OO Database Conceptual Schema	54
4.2	Examples of the Format of Relational Conceptual Schema	55
4.3	RETOO Object-Oriented Conceptual Schema	56
4.4	Class with Extra Methods	58
4.5	Referential Integrity	59
4.6	RETOO Database Schema Translation Tool	61
4.7	Steps of Identifying Classes	62
5.1	RETOO Database Conceptual Schema Translation Tool	84
5.2	Action Diagram in Referential Integrity Constraints Reading Module	86
6.1	Relational Schemas Used for Comparisons	105
6.2	Translated OO Schema using RETOO Approach	107
6.3	Comparison of Translation Result on Relational Schema 2	107
6.4	Translated OO Schemas Comparison	108
6.5	Comparison on Translating Relational Schema 2	109
6.6	Comparison on Translating Relational Schema 1	110
6.7	Relational Database Conceptual Schema Reading Module	112
6.8	The Relational Database Conceptual Schema Entered by User	113
6.9	Specifying Referential Integrities in RETOO	113
6.10	Specifying Composite Attributes in RETOO	115
6.11	Applying RETOO Translation Schema	116
6.12	Reading Extra Methods for Classes from Users	117
6.13	Translated OO Schema	118



## **LIST OF ABBREVIATIONS**

ANSI	-	American National Standards Institute
BLOOM	-	Barcelona Object-Oriented Model
CCS	-	Canonical Conceptual Schema
DBMS	-	Database Management System
DDB	-	Distributed Databases
DDBMS	-	Distributed Database Management System
DDS	-	Data Dictionary System
EER	-	Extended Entity Relationship
ER	-	Entity-Relationship
ES	-	External Schema
FD	-	Functional Dependency
FK	-	Foreign Key
GCS	-	Global Conceptual Schema
GES	-	Global External Schema
GUI	-	Graphical User Interface
ID	-	Inclusion Dependency
InS	-	Intermediate Schema
LCS	-	Local Conceptual Schema
LES	-	Local External Schema
LIS	-	Local Internal Schema
MDBS	-	Multi Database System
OID	-	Object Identity

OMT	-	Object Model Technology
OO	-	Object-Oriented
OODB	-	Object-Oriented Database
OODBMS	-	Object-Oriented Database Management System
PK	-	Primary Key
SPARC	-	Standards Planning and Requirement Committee
SQL	-	Structured Query Language
WWW	-	World Wide Web

# **CHAPTER 1**

## **INTRODUCTION**

In today's information age, databases and database technology are having a major impact on the growing use of computers. The government, education, medicine, engineering, business and other areas have computerized all or part of their daily functions. Undoubtedly, these computerization processes often include database systems to model and store the information of the real-world entities involved in these functions.

The computing environment in most of these contemporary organizations contains distributed, heterogeneous, and autonomous hardware and software systems. Therefore, there is an increasing need to support the co-operations of the services provided by these different software and hardware.

Likely, the existence of multiple, heterogeneous and autonomous databases within an organization means the globally important information exists in separate local database management systems (DBMSs), thus making the existing data inaccessible to remote users. One solution is to integrate these databases to form a single cohesive definition of a multi-database. Most of the integration are made possible with the support of database translation, which is the task of translation from one database conceptual schema into another.

In this research, a set of translation rules used to translate relational database conceptual schema into object-oriented (OO) database conceptual schema is proposed.



Subsequently, this set of translation rules are applied in a database schema translation tool prototype, called RETOO (Relational-To-Object-Oriented), with the assumption that OO conceptual schema is used as the canonical conceptual schema (CCS). This canonical conceptual schema will then be integrated into the global conceptual schema (GCS) of the distributed, heterogeneous database system. Figure 1.1 briefly illustrates the system.

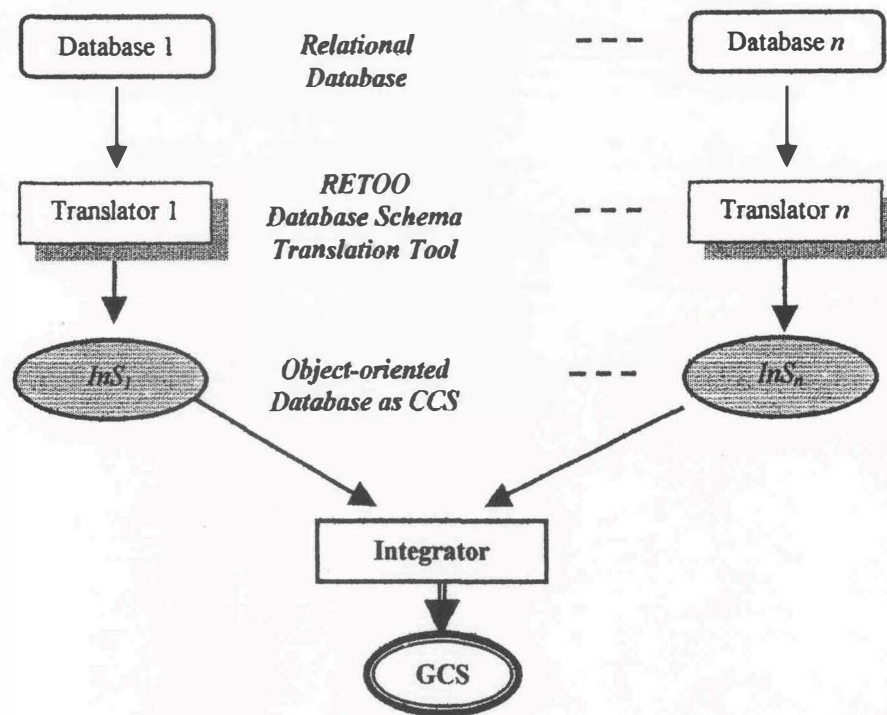


Figure 1.1: Relational-to-Object-Oriented Database Schema Translation Tool

$InS_1 \dots InS_n$  shown in Figure 1.1 are intermediate schemas or known as canonical conceptual schemas. As mentioned above, in this project, the CCS or  $InS_i$  where  $1 \leq i \leq n$ , is the OO conceptual schema.

## **1.1 Problem Statement**

Distributed, heterogeneous and automated database systems are playing an important role to support the global-accessibility of information. Since majority of these databases are already developed, a bottom-up integration process is needed to integrate all the local database schemas into the global conceptual schema. Database integration consists of two steps:

- i. schema translation
- ii. schema integration

The translator will translate the participating local database conceptual schemas to a common canonical intermediate representation or canonical conceptual schema (CCS). Then, each intermediate schema is integrated into a global conceptual schema (GCS).

Most works on schema translation deal with conversion from the entity-relationship (ER) model to the relational model or some extension of it (Castellanos et al., 1994; Castellanos and Saltor, 1991). There are many works on translation from ER model into relational model or vice versa (Huang et al., 1997; Seol, 1997; Lukovic and Mogin, 1996). Besides, works on general frameworks for schema translation were also carried out (McBrien and Poullovassilis, 1998).

Nevertheless, only a few works have been done on translating relational schema into OO schema (Stanisic, 1999; Fong, 1997; Castellanos et al., 1994; Castellanos and Saltor, 1991). Stanisic focused his work not only on schema translation, but query

translation as well. While Castellanos et al. proposed a methodology to translate the relational model into Barcelona Object-Oriented Model, namely BLOOM model. However, these works have their limitations respectively, especially in terms of translated OO model representation. All these works are discussed in more detail in Chapter 3.

Two main motivations of RETOO are the lack of translation from relational database conceptual schema into object-oriented database conceptual schema and object-oriented technology is the future direction. Section 2.7 explains the rationale behind the decision of choosing relational model to be translated into object-oriented model, which is the intermediate schema.

## **1.2 Problem Solving**

This research has proposed a set of translation rules to translate relational conceptual schema into object-oriented conceptual schema using inclusion dependency, key attributes and types of attributes. These translation rules were then implemented in a translation tool prototype, called RETOO. RETOO, which operates semi-automatically, is able to translate the relational into OO database conceptual schema. The translation rules, the architecture of RETOO and its functionality are discussed thoroughly in Chapter 4 and 5.

### **1.3 Objectives**

The objectives of this research project are:

- i. To propose an alternative approach to translate relational database conceptual schema into object-oriented database conceptual schema.
- ii. To develop a prototype for relational-to-object-oriented database conceptual schema translation tool based on the approach proposed in this research.

### **1.4 Research Scope**

This translation tool is designed to translate the pre-existing, well-designed relational database conceptual schema to object-oriented database conceptual schema. Thus, it is developed based on the bottom-up design of database integration process. Section 2.5 discusses the bottom-up design in more detail. Users, particularly database system developers, who wish to translate the local relational database schema to a canonical intermediate schema, which is the object-oriented schema, can get the intended result with this easy-operable tool.

The database integration does not only require the translation from one database schema into another; it may also lead to the necessity of query mapping for database transaction. However, since the mapping between query languages is another large part in database integration, we do not include it in this research due to time constraint.

## **CHAPTER 2**

### **PRELIMINARIES**

In this chapter, we shall discuss the preliminaries, which is the overall background of this research. These include the concepts of distributed database system, relational database system and object-oriented database system.

#### **2.1 Distributed Database**

In centralized database system, all system components reside at a single computer or site. The components include the data, the DBMS software, and the associated secondary storage devices such as disks for on-line database storage and tapes for backup. A centralized database can be accessed remotely via terminals connected to the site even though the data and DBMS software principally reside at a single site.

Figure 2.1 shows a centralized database, which is accessed remotely by other sites. The database resides and centrally managed by computer system at *Site 2*. This means all the requests are routed to that site. Undoubtedly, this will lead to the problem of transmission delays consequently. On account of this, many organizations have decided to migrate their existing centralized database system to distributed database system.

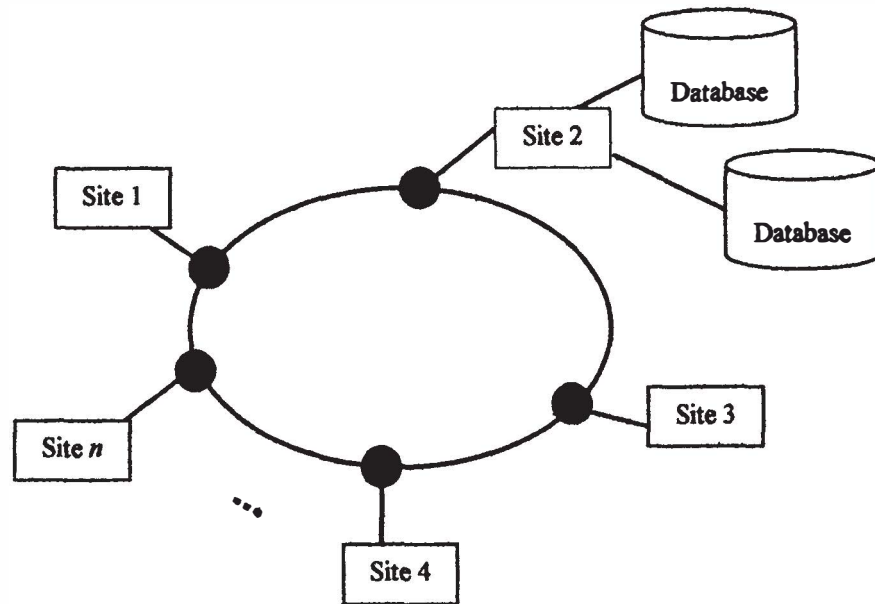


Figure 2.1: A Centralized Database Accessed Remotely

In recent years, there has been a rapid trend towards the distribution of computer systems over multiple sites that are interconnected via a communication network (Brodie, 1993). Besides distributed enterprise collaboration applications and internal business operation applications, many organizations distribute their databases to network servers. These distributed databases (DDB) can reside on network servers on the World Wide Web (WWW), on corporate intranets or extranets, or on other company networks (O'Brien, 1999). Several factors have led to the development of distributed database system, such as the improved performance in terms of transaction time, allowing data sharing while maintaining some measure of local control, and increasing reliability and availability of databases.