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 cirri-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:	My housemates:	My friends:
Puan Lilly Suriani	*Lee Ai*	*Brother Tan Ho Soon*
Kak Yati	*Hui Lian*	*Residents of Buddhist Houses*
Kak Umu	*Wee Peng*	*Lei Won*
Kak Iza	*Hung Chee*	*Poh Cheang*
Kak Maya	*Seok Har*	*Boo Ping and Hui Khim*
Shamala	*Sook Leng*	*Shau Meng*
Umi Zurina	*Pei Ten*	*Eddie Chow*
Maslinda	*Sing Yi*	*Stephen Ho*
Maizatul	*Lay Fung*	*Chong Fang*
Ika	*Lian Pey*	*Hui Ling*
Auntie Thevanan	*Geok Hun*	

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

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.

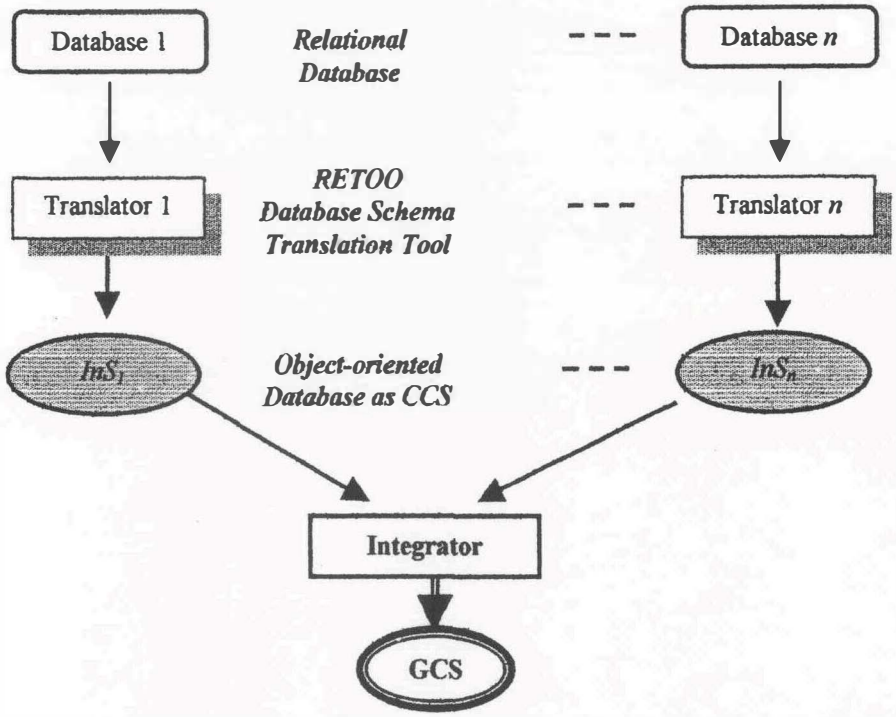

Figure 1.1: Relational-*to*-Object-Oriented Database Schema Translation Tool

InS_1 ... 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 Poulovassilis, 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.

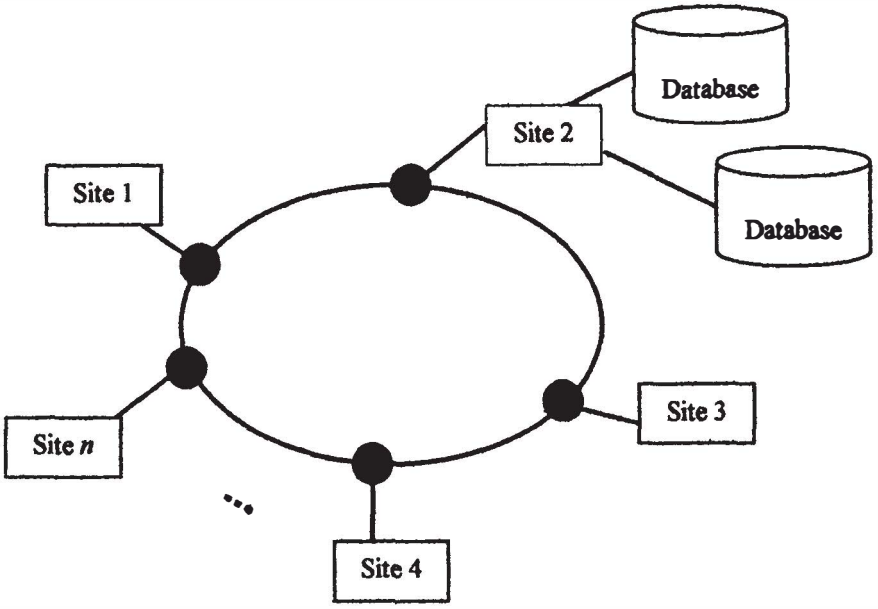

Figure 2.1: A Centralized Database Accessed Remotely

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.