Object Relationship Notation (ORN) for Database Applications

Enhancing the Modeling and Implementation of Associations

Object Relationship Notation (ORN) for Database Applications

Enhancing the Modeling and Implementation of Associations

by

Bryon K. Ehlmann
Southern Illinois University Edwardsville
Edwardsville, IL, USA

Springer

Bryon K. Ehlmann
Department of Computer Science
Southern Illinois University Edwardsville
Edwardsville, IL 62026
behlman@siue.edu

ISBN 978-0-387-09553-0 e-ISBN 978-0-387-09554-7
DOI 10.1007/978-0-387-09554-7
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009929228

© Springer Science+Business Media, LLC 2009
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Barbara,
Bethany and Jonathan

Preface

Conceptually, information that is relevant to a database application consists of objects and relationships. Since the advent of object-orientation in the 1980s, much emphasis has been placed on objects and, in my opinion, too little on the relationships that exist among them. Many books have been written about objects. This book is about relationships.

The emphasis on objects in software systems analysis, design, and implementation has improved the productivity of developing and maintaining software systems. I believe that placing more emphasis on relationships can have a similar effect, especially for database systems. This book is founded on that belief.

Motivation

The motivation for this book is the same as the motivation driving the research and its results that are the subject of this book. This motivation originated in 1990 when I attended a class on Ontos, an early object database management system (ODBMS). Despite much support in the system for the storing of objects, i.e., object persistence, I saw little support for relationships, specifically the types of relationships that bind different objects together. A classic example of this type of relationship is the one-to-many relationship that binds each object representing a department to the objects representing the employees who work for the department. The support for such relationships improved little in later ODBMSs and in a subsequent ODBMS standard. I viewed the support for relationships in ODBMSs to be slightly less than that provided by relational database management systems (RDBMSs).

And this latter support has, I believe, been dismal, especially when one considers the age and pervasiveness of RDBMSs and the fact that for decades relational databases have been developed based on the Entity-Relationship (ER) model. This belief has provided even more motivation for the contents of this book. When the relationships defined in ER models are implemented in relational databases, much of their meaning—expressed by their one-to-one, one-to-many, many-to-many, and cardinality notations—is lost or very difficult to resurrect. Support for these concepts is lacking in RDBMSs, which makes implementation unnecessarily more difficult and error-prone.

I will “come clean” on another motivation for my research and this book. I confess I have a passion for relationships. I find them fascinating! There are so many different kinds, and it is often a challenge to try to discover their essence. What is it

that makes one type of relationship different from or similar to another? Some aspects of a relationship's nature can be defined mathematically, but others often seem only "vague notions." For example, the notions of "containment" contribute to the nature of many relationships, e.g., a car contains, or is composed of, many parts. But what exactly is containment? It seems to come in different "flavors"—e.g., a car also contains materials like plastic, which aren't really "removable," and a car may contain passengers, which certainly must be removable. Can these flavors of containment somehow be generically defined and distinguished so that they can be ultimately and properly "appreciated," i.e., managed, by a DBMS?

Purpose

The purpose of this book is to highlight in more detail the lack of support for relationships in DBMSs but, more importantly, to point a way toward improvement. To this end, I describe in this book a simple, yet powerful notation that modestly extends the basic ER model and its more modern, standard incarnation, the UML class diagram. This notation, the Object Relationship Notation (ORN), allows the true nature of relationships—more specifically, "associations" using UML terminology—to be more precisely defined. It also allows these relationship definitions to be included in a data definition language (DDL), like SQL. This permits a more direct mapping from model to implementation and facilitates better support for relationships in DBMSs, both object and relational.

In this book, I also describe patterns and software tools that demonstrate how ORN can be used to more productively model and implement databases. The patterns are given in Chapter 4, and the software tools are available via the Web. (See the Downloads section later in this Preface.) The patterns, called *association patterns*, assist in developing a better understanding of relationships and in modeling data, regardless of whether ORN is used. The software tools—ORN Simulator, Object Relater *Plus*, and ORN Additive—can be used as research tools to verify the examples given in this book, to test others, and to serve as prototypes for development efforts that would integrate ORN into a commercial modeling tool or DBMS. In addition, the software tools can be used to assist in the development of real database applications, and the ORN Simulator can serve as a pedagogical tool for learning the concepts of data modeling and transaction processing.

The research results I describe in this book do not relate to a "sizzling" area of research in computer science. Rather, they relate to a traditional area—namely, data modeling and database definition—where little research is occurring, where the major advancements were made in the 1970's, where much effort since then has been on standardization, and thus where any breakthrough is slow to be adopted. Nevertheless, this book is an effort on my part to present and promote some practical research results that I feel can advance the state-of-the-art in this traditional area. I believe that these results can be adopted with little cost and can significantly improve the productivity of developing database systems and improve their integrity.

Readers

The intended readers for this book are researchers in database systems, developers of DBMSs or data modeling tools, practitioners of database systems development, and students of database management. Others who may be interested are software engineering researchers or anyone having an interest in data modeling, database development, or simply learning more about relationships. The book can be used as a supplemental text for courses in database management or database modeling where students use association patterns or one or more of the tools that are discussed.

The prerequisite knowledge for the reader is a basic understanding of data structures, files, and databases. The database knowledge needed is that normally obtained in an introductory course in database management—most importantly, a familiarity with the ER model, relational databases and SQL, and to a lesser extent object databases. In Chapter 1, I provide a brief overview of all of these topics. I use UML class diagrams extensively throughout the book. In Chapter 1, I explain these diagrams so that a reader familiar with ER diagrams should develop a sufficient understanding.

Structure

This book is divided into three parts. The first part contains introductory material about relationships in general and ORN in particular. This material is intended for all readers. The second part contains material for readers interested in using ORN to develop database applications or for readers merely interested in developing a better understanding of the benefits and capabilities of ORN. The third part is for readers interested in including support for ORN in database modeling tools or DBMSs or for readers merely interested in investigating the algorithms required for implementing ORN. Here is a brief summary of each chapter:

Part I About Relationships and ORN

- Chapter 1, **Introduction: Including a Brief History of Relationships**, provides a general introduction to relationships, gives a brief history of relationships in terms of how we have understood and recorded them over the years, and discusses some of the problems encountered today in modeling and recording, i.e., implementing, them using DBMSs. The historical account provides a short review of relational databases and UML class diagrams, which should prove helpful in understanding the remaining contents of the book.
- Chapter 2, **Object Relationship Notation (ORN)**, provides the syntax and semantics for ORN—i.e., its form, both textual and graphical, and its meaning—and gives examples of associations to illustrate the meaning of the notation's symbols.
- Chapter 3, **ORN Simulator: A Modeling Tool Where Associations Come Alive**, presents a modeling tool that the reader can access via the Web. This tool allows its user to easily experiment with ORN and thus develop a better under-

standing of its semantics. The user can easily create an ORN-extended database model and then manipulate a prototype database in the context of the model. By creating and deleting objects and creating, destroying, and changing associations in the database, the user readily sees the effects of different ORN specifications.

- Chapter 4, **Association Patterns: Emerging from a Variety of Association Types**, provides examples of the variety of associations that can be defined using ORN and identifies among them some association patterns that can be used to guide database modeling.
- Chapter 5, **Comparing ORN to Similar Declarative Schemes**, compares ORN to similar schemes for declaring relationship semantics in terms of simplicity and expressive power. It also discusses how ORN relates to efforts to better define the semantics of whole-part relationships.

Part II Using ORN to Develop a Database System

- Chapter 6, **ORN Additive: A Tool for Extending SQL Server with ORN**, discusses a tool that allows its user to add ORN support to Microsoft's SQL Server. The tool automatically generates the T-SQL triggers and stored procedures that are required to implement ORN-defined associations in a database application.
- Chapter 7, **Object Relater *Plus* (OR+): An ORN-Extended Object DBMS**, discusses a tool that allows its user to add ORN support to Progress's Object Store. The tool automatically generates the C++ methods that are required to implement ORN-defined associations in an object database application.
- Chapter 8, **Mapping Database Models to DDLs: From ORN-Extended Class Diagrams to ORN-Extended DBMSs**, shows how one can easily map the ORN-defined associations in a class diagram to an ORN Additive/T-SQL definition of a database or an OR+ Object Database Definition Language (ODDL) definition of a database.
- Chapter 9, **Association Semantics: Dealing with the Subtleties, Inconsistencies, and Ambiguities**, discusses some of the finer points about association semantics and how certain association definitions can lead to associations and combinations of associations that are mathematically inconsistent, likely inconsistent, or ambiguous. Database developers can better identify and deal with such associations when their semantics are defined by ORN.

Part III Adding ORN to the DBMS

- Chapter 10, **A Conceptual Implementation of ORN: Exploring Semantic Circularity and Ambiguity**, provides algorithms, which are independent of database type, for the implementation of ORN. Based on these algorithms, it also discusses the circularity and clarity of ORN semantics in the presence of link cycles within a database. A theorem is stated and proved about the clarity of ORN semantics.
- Chapter 11, **Adding ORN to the SQL Standard for RDBMSs**, provides the syntax and semantics for adding ORN to the SQL relational DBMS standard.
- Chapter 12, **Adding ORN to the ODMG Standard for ODMSs**, provides the syntax, semantics, and algorithms for adding ORN to the ODMG 3.0 standard for Object Data Management Systems (ODMSs).

The diagram in Fig. P.1 shows the dependencies between chapters and thus the order in which chapters can be read. A reader very knowledgeable of data modeling, class diagrams, and database management systems can skip Sections 1.1 and 1.2 of Chapter 1, but should read Sections 1.3 and 1.4 for a proper introduction to the subject matter of this book. If desired, Chapters 2 and 3 can be studied together to allow experimentation and perhaps a better understanding of the contents of Chapter 2.

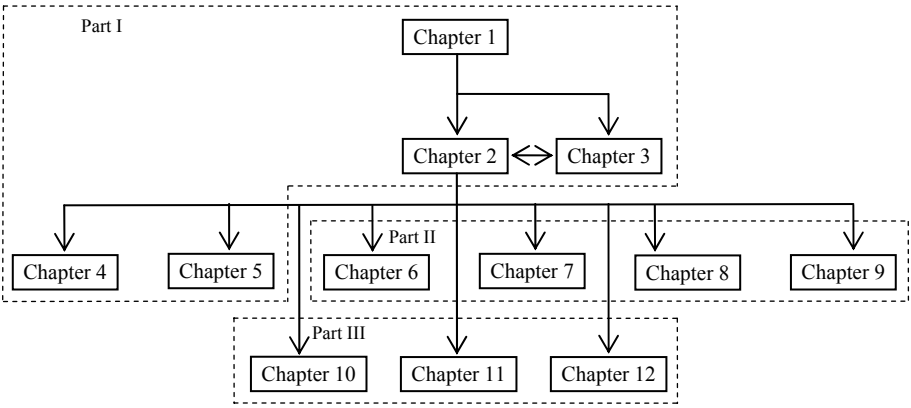

Fig. P.1. Dependencies between chapters

Downloads

The ORN Additive and Object Relater *Plus* (OR+) software tools, which are discussed in this book, can be downloaded from www.siue.edu/~behlman. Access the site and click on the "Download Software" link.

Acknowledgments

A number of people and organizations have contributed to the writing of this book and the work that produced its contents.

I thank the Computer Science Department and School of Engineering at Southern Illinois University Edwardsville and the Computer and Information Systems Department at Florida A&M University for their support. I particularly thank the students in my DBMS, software engineering, and independent study classes for the projects they completed over the course of many semesters that resulted in the extensive testing of the ORN Simulator, OR+, and ORN Additive software prototypes and the association patterns presented in Chapter 4. Special recognition in this regard should go to Anwesha Chattopadhyay, Brock Green, Jian Li, and Rajesh Vaddi.

I thank the co-authors on my previous papers related to ORN for their reviews and assistance. They are Larry Dennis, Siebert Hardeman, Erika Neal, Gregory Riccardi, Naphtali Rishe, Michael Stewart, Jinyu Shi, and Xudong Yu. Gregory Riccardi deserves special recognition for giving me the latitude and support that made my early efforts in developing ORN possible.

I thank my editor at Springer, Susan Lagerstrom-Fife, and especially her assistant, Sharon Palleschi, for working with me, reviewing my drafts, and answering my many questions.

I thank my wife Barbara Ehlmann and Professor William White for their efforts in reviewing and proofreading the entire book and finding my many errors. Still more errors were found by the much appreciated efforts of Bethany Ehlmann, my daughter, and three students—Gerald Jackson, Aaron Jansen, and John Jenkins—in carefully reviewing and proofreading selected chapters.

Finally, I am especially grateful to my wife for enduring my many moments of mental preoccupation while writing this book. Her patience, understanding, and tremendous support made this book possible and makes my work more enjoyable.

Early research related to the development of ORN and Object Relater *Plus* (OR+) was partially supported by the National Science Foundation under grant CDA-9313299 and the U.S. Dept. of Energy under contract DE-FG05-92ER40735.

Jim Melton, SQL standards editor and SQL standards representative for Oracle Corporation, provided helpful clarification on referential integrity for SQL:2003 reference types, which was relevant to Chapter 10.

The opening quotation in Chapter 1 is extracted with permission from (Zdonik and Maier 1990), © 1990, Morgan Kaufmann.

Edwardsville, Illinois Bryon K. Ehlmann

About the Author

Bryon K. Ehlmann is currently Professor of Computer Science at Southern Illinois University Edwardsville. He earned his B.S. and M.S. degrees in Computer Science from the University of Missouri at Rolla (now the Missouri University of Science and Technology) and his Ph.D. degree in Computer and Information Sciences from Florida State University. Professor Ehlmann's research interests include database management, object databases, data modeling, software engineering, and user interfaces. He has twelve years of industry experience with Burrough's Corp. and Unisys Corp. in developing software systems in the areas of database management, report generation, and user interface development. In addition, he has twenty-seven years of academic experience in teaching programming, data management, and software engineering and has published over twenty-five papers in the areas of database management and data modeling. Professor Ehlmann can be contacted at behlman@siue.edu and would appreciate any comments, questions, or corrections related to this book.

Contents

Preface ..... **vii**
Motivation vii
Purpose viii
Readers ix
Structure ix
Downloads xi
Acknowledgments xi

About the Author ..... **xiii**

Contents ..... **xv**

List of Abbreviations ..... **xxi**

Part I About Relationships and ORN **1**

Chapter 1 Introduction **3**
Including a Brief History of Relationships
1.1 Relationships 3
1.2 A Brief History of Relationships 4
1.2.1 The paper-based system 5
1.2.2 The computerized file-based system 6
1.2.3 Early data models and DBMSs 8
1.2.4 The relational model and RDBMS 15
1.2.5 The entity-relationship model and database 19
1.2.6 Semantic models, the object model, and the ODBMS 23
1.2.7 The object-relational DBMS 31
1.3 Problems in Modeling and Implementing Relationships 32
1.4 Preview of a Solution 35

Chapter 2 Object Relationship Notation (ORN) **39**
2.1 Syntax 39
2.2 Graphical Representation 40
2.3 Semantics 41
2.4 Examples 42
2.4.1 <*-to-*> 44

2.4.2 <1-to-*> ... 45
2.4.3 <0..1-to-*>|- ... 47
2.4.4 <0..1-to-*>|-X- ... 47
2.4.5 <0..1-to-1..*>? ... 48
2.4.6 !<0..1-to-*> ... 49
2.4.7 '<*-to-1..*> ... 50
2.5 Flashback to the Company Database ... 51

Chapter 3 ORN Simulator ... 53
A Modeling Tool Where Associations Come Alive
3.1 Creating a Database Model and a Database ... 53
3.2 Verifying Association Semantics ... 60
3.3 Architecture ... 66
3.4 Benefits ... 67

Chapter 4 Association Patterns ... 69
Emerging from a Variety of Association Types
4.1 Context ... 69
4.2 Pattern Descriptions ... 71
4.2.1 “is defined by” pattern ... 73
4.2.2 “is recorded for” pattern ... 75
4.2.3 “is a realization of” pattern ... 79
4.2.4 “is associated by” pattern ... 80
4.2.5 “is an update of” pattern ... 81
4.2.6 “is a part of” pattern ... 82
4.2.7 “is a” pattern ... 89
4.3 Patterns in Database Modeling and Implementation ... 90
4.4 Associations That Don’t Conform to a Pattern ... 95
4.5 Conclusion ... 96

Chapter 5 Comparing ORN to Similar Declarative Schemes ... 97
5.1 Schemes for Relational Databases and Object Relations ... 98
5.1.1 Null constraint ... 98
5.1.2 Key constraints ... 99
5.1.3 Referential integrity rules ... 100
5.1.4 Surjectivity constraints ... 102
5.1.5 Additional relationship semantics ... 103
5.2 Schemes for the ER model ... 105
5.3 Schemes in ODBMSs ... 105
5.4 Whole-Part Properties and Dimensions for Class Diagrams ... 107
5.4.1 Proposed association dimensions for class diagrams ... 108
5.4.2 Primary characteristics for whole-part relationships ... 111
5.4.3 Secondary characteristics for whole-part relationships ... 112
5.5 Conclusion ... 113

Part II Using ORN to Develop a Database System 115

Chapter 6 ORN Additive 117
A Tool for Extending SQL Server with ORN
6.1 Capabilities, Operations, and Architecture 117
6.1.1 Capabilities 118
6.1.2 Operations and architecture 119
6.1.3 The +ornddl command 121
6.1.4 The +orndml command 121
6.2 ORN Additive DDL Statements 122
6.2.1 USE statement 122
6.2.2 CONSTRAINT statement 123
6.2.3 DELETE statement 123
6.2.4 SET ORN_MESSAGE_NUMBER_BASE statement 125
6.2.5 Example of a query file with ORN Additive DDL statements 125
6.3 ORN Additive DML Statements 125
6.3.1 USE statement 125
6.3.2 BEGIN TRANSACTION statement 127
6.3.3 SAVE TRANSACTION statement 127
6.3.4 COMMIT TRANSACTION statement 127
6.3.5 ROLLBACK TRANSACTION statement 127
6.3.6 SET RXC_MODE statement 128
6.3.7 ENABLE/DISABLE ORN_TRIGGERS statement 129
6.3.8 Example of query file with ORN Additive DML statements 130
6.4 Conclusion 132

Chapter 7 Object Relater *Plus* (OR+) 133
An ORN-extended Object DBMS
7.1 Capabilities and Compatibilities 133
7.2 ODDL 135
7.2.1 ODDL specification 135
7.2.2 <association> 138
7.2.3 Control commands 140
7.3 ODML 141
7.4 Architecture, Implementation, and Extensibility 143
7.4.1 Architecture 143
7.4.2 Implementation 145
7.4.3 Extensibility 146
7.5 Conclusion 147

Chapter 8 Mapping Database Models to DDLs 149
From ORN-Extended Class Diagrams to ORN-Extended DBMSs
8.1 Mapping an ORN-Extended Model to an ORN-Extended SQL 149
8.1.1 Transforming the model for a relational database 149
8.1.2 Mapping classes to tables 153

8.1.3 Mapping associations to foreign keys 154
8.2 Mapping an ORN-Extended Model to an Object DDL 157
8.2.1 Transforming the model for an object database 158
8.2.2 Mapping classes to class definitions and extents 159
8.2.3 Mapping associations to object-based attributes 160
8.3 Conclusion 163

Chapter 9 Association Semantics 165
Dealing with the Subtleties, Inconsistencies, and Ambiguities
9.1 Inconsistencies 165
9.1.1 Within an *<association>* 166
9.1.2 Involving *<association>* combinations 170
9.2 Inconsistency Detection 171
9.3 Ambiguities 172
9.4 Associations as Functions 174
9.5 Conclusion 177

Part III Adding ORN to a DBMS 179

Chapter 10 A Conceptual Implementation of ORN 181
Exploring Semantic Circularity and Ambiguity
10.1 Algorithms 182
10.1.1 Algorithm CreateObject 182
10.1.2 Transaction operations 183
10.1.3 Supporting pseudocode for complex object operations 184
10.1.4 Algorithm CreateLink 185
10.1.5 Algorithm DeleteObject 186
10.1.6 Algorithm DestroyLink 187
10.1.7 Algorithm ChangeLink 187
10.2 Link Cycles 188
10.2.1 Circularity 189
10.2.2 Ambiguities 190
10.2.3 The Theorem for ORN Semantic Clarity and its proof 196
10.3 Conclusion 203

Chapter 11 Adding ORN to the SQL Standard for RDBMSs 205
11.1 Motivation 205
11.2 Overview of SQL Association Capabilities 206
11.3 Proposed ORN Extension to SQL 209
11.4 Conclusion 213

Chapter 12 Adding ORN to the ODMG Standard for ODMSs 215
12.1 Motivation 215
12.2 Adding ORN to ODL 217
12.2.1 Associations in ODL 217

12.2.2 Adding ORN syntax 218
12.2.3 Adapting ORN semantics to ODL 219
12.3 Algorithms 221
12.3.1 Class Transaction 223
12.3.2 Method new() 225
12.3.3 Method delete() 226
12.3.4 Method _enforce_binding() 227
12.3.5 Method form_tp_R() 229
12.3.6 Method drop_tp_R() 230
12.3.7 Method _enforce_explicit_binding() 230
12.3.6 Method change_tp_R() 231
12.3.9 Example 232
12.4 Conclusion 234

Bibliography 237

Index 241

List of Abbreviations

API	application program interface
DBMS	database management system
DBTG	Database Task Group
DDL	data definition language
ER	Entity-Relationship
MDD	model-driven development
ODBMS	object database management system
ODDL	Object Database Definition Language
ODL	Object Definition Language
ODM	object-to-database mapping
ODMG	Object Data Management Group
ODML	Object Database Manipulation Language
ODMS	object data management system
OML	Object Manipulation Language
OO	object-oriented
OQL	Object Query Language
OR+	Object Relater *Plus*
ORN	Object Relationship Notation
RDBMS	relational database management system
RXC	Relationship eXChange
SQL	Structured Query Language
UML	Unified Modeling Language