

**D A T A B A S E
T R A N S A C T I O N
M O D E L S**

For Advanced Applications

Edited

by

AHMED K. ELMAGARMID

9461527

Database Transaction Models for Advanced Applications

Edited by

Ahmed K. Elmagarmid



E9461527

MORGAN KAUFMANN PUBLISHERS

San Mateo, California

Editor: *Bruce M. Spatz*
Production Manager: *Yonie Overton*
Cover Mechanical Artist: *Patty King*
Indexing/Electronic Output: *Superscript Typography*

QA

005.75

Database manag.

distributed databases

Morgan Kaufmann Publishers, Inc.
Editorial Office:
2929 Campus Drive, Suite 260
San Mateo, CA 94403

©Morgan Kaufmann Publishers, Inc.
All rights reserved
Printed in the United States of America

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise—without the prior written permission of the publisher.

95 94 93 92 5 4 3 2 1

Library of Congress Cataloging in Publication Data is available for this book.

ISBN 1-55860-214-3
ISSN 1046-1698

Database Transaction Models for Advanced Applications

THE MORGAN KAUFMANN SERIES IN
DATA MANAGEMENT SYSTEMS

Series Editor, Jim Gray

DATABASE TRANSACTION MODELS FOR ADVANCED APPLICATIONS

Edited by Ahmed K. Elmagarmid (Purdue University)

A GUIDE TO DEVELOPING CLIENT/SERVER SQL APPLICATIONS

Setrag Khoshafian (Portfolio Technologies, Inc.),

Arvola Chan (Versant Object Technology),

Anna Wong (CLaM Associates), and

Harry K. T. Wong (Nomadic Systems)

THE BENCHMARK HANDBOOK FOR DATABASE AND TRANSACTION
PROCESSING SYSTEMS

Edited by Jim Gray (Digital Equipment Corporation)

CAMELOT AND AVALON: A DISTRIBUTED TRANSACTION FACILITY

Edited by Jeffrey L. Eppinger (Transarc Corporation),

Lily B. Mummert (Carnegie Mellon University),

and Alfred Z. Spector (Transarc Corporation)

DATABASE MODELING AND DESIGN: THE ENTITY-RELATIONSHIP
APPROACH

Toby J. Teorey (University of Michigan)

READINGS IN OBJECT-ORIENTED DATABASE SYSTEMS

Edited by Stanley B. Zdonik (Brown University) and

David Maier (Oregon Graduate Center)

READINGS IN DATABASE SYSTEMS

Edited by Michael Stonebraker (University of California at Berkeley)

DEDUCTIVE DATABASES AND LOGIC PROGRAMMING

Jack Minker (University of Maryland)

*To beautiful little
Najd Mohammed Elmagarmid,
April 7, 1988–December 15, 1990*

Foreword

The transaction concept has emerged as the key structuring technique for distributed data and distributed computations. Originally developed and applied to database applications, the transaction model is now being used in new application areas ranging from process control to cooperative work. Not surprisingly, these more sophisticated applications require a refined and generalized transaction model. The concept must be made recursive, it must deal with concurrency within a transaction, it must relax the strict isolation among transactions, and it must deal more gracefully with failures.

This book collects, for the first time, the many generalizations of the basic transaction model. Taken together it represents the latest and best thinking on how to structure distributed and heterogeneous computations. The authors represent the spectrum from research universities to industry. The applications they discuss encompass planning, design, engineering, manufacturing, and commerce. Their approaches range from theoretical models to practical tool kits. In sum, this book covers the entire spectrum of thought on the topic of generalized transaction models and transaction processing.

This book stands as a milestone, collecting the many intellectual threads developed over the decade of the 1980s. Researchers are now working to merge these many approaches into a new unified transaction concept able to handle all applications. This new emerging concept will likely be one of the major contributions to Computer Science in the coming decade.

Jim Gray
Digital Equipment Corporation

Preface

What constitutes a transaction model? This innocent question from an uninformed colleague of mine stopped me, for the first time in ten years, to really think about this issue. It is easy to understand that a data model describes the structure of the database, i.e., data types and the constraints related to the data. But what components one should find in a transaction model? After mumbling something about how to specify the correctness of the database, and the type and structure of the logical operations to manipulate it, I quickly realized that giving a satisfactory answer is not as easy I thought. Several more hours of pondering upon the topic gave more refined, but equally unsatisfactory answers - especially in the light of the new developments in the transaction processing research. All my attempts to give a definition centered not only on the description of properties of the notion of a transaction, but also on the external mechanisms that guarantee such properties. This is exactly what the different chapters of this book are all about.

Active research on database transaction processing dates back to the early '70s, and many theoretical and pragmatic issues of the "traditional transactions" have been explored. The key properties such as atomicity of transaction execution, isolation of intermediate results and a universal correctness criteria (serializability) have been discovered and studied intensively. However, as pointed out by many early researchers, much of this discussion was intended to be relevant for business oriented database applications, where transactions were assumed to be simple, short-lived and non-hierarchical in structure. Attempts to blindly apply this work to other contexts has almost in every case that I know of, produced impractical and sometimes even disastrous results. Indeed, one-by-one each one of the fundamental "ACID" properties has been challenged in various environments, in spite of the appeal of the theoretical elegance of such concepts as serializability or atomic commitment.

So what are the new issues that need to be addressed by these new

transaction models? This many faceted question is best answered by reading the various chapter in this book; my attempt here is only to illustrate how a single modification in the assumptions gives rise to several major questions to be answered.

Let's look at one simple (realistic) modification that the transactions are long-lived, their duration is not seconds but minutes, hours or weeks. This type of transactions are especially natural in Computer Aided Design (CAD) or cooperative environments (e.g., in Groupware). The most common commercially used concurrency control method, blocking the data via locks, becomes totally impractical as one would exclude all the access to data for hours. Actually this single modification severely challenges the principles of guaranteeing isolation and atomicity of the traditional transactions, and introduces naturally the notion of compensation used in S-transactions, Sagas, Open Nested transactions and transactions on Active Objects. Secondly, as such environments are decentralized in nature, it is natural to extend the traditional flat transaction control structure to a hierarchical one, i.e., introduce the notion of transaction nesting with subtransactions executing at various sites. Finally, for such long-living transactions it is also typical to allow several alternative execution threads which leads to incorporating multiple acceptable final states into the correctness criteria (e.g., Flex transactions).

The different chapters of this book provide an excellent snapshot of the various questions raised by the new application areas, where transaction processing is needed, but where the underlying assumptions make it impossible to apply the traditional models and algorithms. The book acts both as a reference for the state of the art in advanced transaction models, as well as a good starting point for anybody interested in addressing the important open questions in the area. Since the chapters are written by different authors, the book has unique value in collecting to a single volume various alternative views on the topics, fact that made the book especially appealing to me personally.

Henry Tirri
University of Helsinki

Acknowledgements

This book would not have come about without the contributions and dedication of the many authors involved. This book is an acknowledgement of their many and valuable contributions to the field.

The contributions have endured a roller coaster that lasted for over two years. What was promised to be a fast and painless task took five times longer than promised and went through far too many revisions. We all are indebted to the over 100 reviewers who did the hard job. I was pleased at the level of professionalism and desire for the highest levels of perfection exhibited by the contributors and reviewers.

This book would not have been possible without the help of many of the InterBase team at Purdue. A special mention goes to Sue Barton, Omran Bukhres, Jiansan Chen, Yungho Leu, James Mullen, and Aidong Zhang.

A great amount of input was provided by so many of my good friends. I will not list them all, but a special mention is given to Amit Sheth and M. Tamer Özsu. Special thanks are due to Henry Tirri for writing a preface to the book and to Jim Gray, the series editor, for writing the Foreword to the book.

Special thanks go to the staff at Morgan Kaufmann. Their dedication to this project has been unyielding. Special thanks go to my Production Manager Yonie Overton. She has put up with a tight and often unpredictable schedule. No amount of thanks or appreciation is enough for the force behind the project, Bruce Spatz, the Senior Editor for this book. Indeed he is solely responsible for the inception of the idea of writing the book on transaction models and receives all the credit for seeing the project through thick and thin. He has been gracious, friendly and accommodating at times when I may have been unreasonable. Thanks are due also to Elizabeth Essex, Associate Product Manager.

The push for writing a book started at a meeting with Mike Morgan in

my office at Purdue in the Autumn of 1988. A lot has happened in the field since that time, and we hope we have captured some of it in these pages.

This book does not particularly cover heterogeneous databases, as that is the topic of an upcoming book by myself, Amit Sheth and Marek Rusinkiewicz.

This book would not have been possible without the financial support I receive from the Purdue Research Foundation, The Indiana Business Modernization and Technology, The National Science Foundation, The Software Engineering Research Center, Bell Northern Research, and NASA.

Ahmed K. Elmagarmid
Associate Professor and Executive Director
Indiana Center for Database Systems
Department of Computer Sciences
Purdue University

Contents

FOREWORD	xvii
<i>Jim Gray</i>	
PREFACE	xix
ACKNOWLEDGEMENTS	xxi
1 TRANSACTION MANAGEMENT IN DATABASE SYSTEMS	1
<i>D. Agrawal and A. El Abbadi</i>	
1.1 Introduction	2
1.2 Execution atomicity	4
1.2.1 Motivation	5
1.2.2 Serializability	6
1.2.3 Conflict Serializability	8
1.3 Failure Atomicity	12
1.3.1 Transaction Failures	12
1.3.2 System Failures	15
1.4 Distributed Databases	16
1.5 Extensions to the Transaction Model	19
1.5.1 Multiversion Databases	19
1.5.2 Nested Transaction Model	20
1.5.3 Transaction Models for Abstract Objects	22
1.6 Concluding Remarks	23

2	INTRODUCTION TO ADVANCED TRANSACTION MODELS	33
	<i>Ahmed K. Elmagarmid, Yungho Leu, James G. Mullen, and Omran Bukhres</i>	
2.1	Introduction	34
2.2	Advanced Transaction Models	35
2.2.1	Cooperative Transaction Hierarchy	38
2.2.2	Cooperative SEE Transactions	39
2.2.3	DOM Transactions	40
2.2.4	A Transaction Model for an Open Publication Environment	41
2.2.5	ConTract Model	41
2.2.6	Split-Transactions	42
2.2.7	Flex Transaction Model	43
2.2.8	ACTA	44
2.2.9	Transaction Tool Kits	45
2.2.10	S Transactions	45
2.2.11	Multilevel and Open Nested Transactions	46
2.2.12	Polytransactions	46
2.3	Summary of Transaction Models	47
3	A COOPERATIVE TRANSACTION MODEL FOR DESIGN DATABASES	53
	<i>Marian H. Nodine, Sridhar Ramaswamy, and Stanley B. Zdonik</i>	
3.1	Introduction	54
3.2	Characteristics of the Transaction Model	57
3.2.1	Hierarchical Organization of Transactions	57
3.2.2	Correctness Criteria	58
3.2.3	Multi-copy versus Single-copy system	59
3.2.4	Operation-Based Recovery	59
3.3	The Model	60
3.3.1	Transaction Groups	61
3.3.2	Cooperative Transactions	63
3.3.3	Operations	63
3.3.4	Histories	64
3.4	Correctness	64
3.4.1	Patterns and Conflicts	64

3.4.2	LR(0) Grammars and DPDAs	66
3.4.3	Correct Transaction Group Histories	67
3.5	Example	67
3.6	Synchronization	71
3.6.1	Algorithm	72
3.6.2	Example	73
3.6.3	Checkpointing	73
3.7	Recovery	77
3.7.1	Dependency Maintenance and Logging	77
3.7.2	Algorithm	80
3.8	Related Research	81
3.9	Summary	83
4	A FLEXIBLE FRAMEWORK FOR TRANSACTION MANAGEMENT IN ENGINEERING ENVIRONMENTS	87
	<i>Sandra Heiler, Sara Haradhvala, Stanley Zdonik, Barbara Blaustein, and Arnon Rosenthal</i>	
4.1	Introduction	88
4.1.1	Motivation	89
4.1.2	Summary of the Approach	91
4.1.3	An Example of Transaction Management in a Simple Organization	92
4.1.4	Related Work	95
4.2	The Model	97
4.2.1	Overview	97
4.2.2	Semantics of Request Processing	100
4.2.3	Request Processing by the TMH	105
4.2.4	Framework Services and Their Interfaces	109
4.3	Protocols for Software Engineering Environments— Approaches and Idioms	110
4.3.1	Specifying Protocols	110
4.3.2	Deadlock Prevention/Detection	112
4.3.3	Limiting Sharing	113
4.3.4	Triggering Copies and Merges	115
4.4	Results and Status	117

5	A TRANSACTION MODEL FOR ACTIVE DISTRIBUTED OBJECT SYSTEMS	123
	<i>Alejandro Buchmann, M. Tamer Özsu, Mark Hornick, Dimitrios Georgakopoulos, and Frank A. Manola</i>	
5.1	Introduction	124
5.2	A Characterization of Transaction Schemes	128
5.2.1	Correctness Criteria	129
5.2.2	Transaction Models	133
5.3	The DOM Transaction Model	138
5.3.1	Example of a DOM Transaction	139
5.3.2	Multitransactions	141
5.3.3	Nested Transactions	142
5.3.4	Compensating Transactions	142
5.3.5	Contingency Transactions	143
5.3.6	Vital and Non-vital Transactions	144
5.4	Formal Specification of the Model	145
5.4.1	Summary of the ACTA Formalism	145
5.4.2	Multitransactions	146
5.4.3	Nested Transactions	148
5.4.4	Contingency Transactions	149
5.4.5	Compensating Transactions	150
5.5	Conclusions and Future Work	151
6	A TRANSACTION MODEL FOR AN OPEN PUBLICATION ENVIRONMENT	159
	<i>Peter Muth, Thomas C. Rakow, Wolfgang Klas, and Erich J. Neuhold</i>	
6.1	Overview	160
6.2	Introduction	160
6.3	The Architecture of the Publication Environment and its Transaction Needs	165
6.3.1	Architecture	165
6.3.2	Requirements for the Transaction Model	171
6.4	Transaction Model	173
6.4.1	Object-Oriented Serializability	176
6.4.2	Object-Oriented Concurrency Control	186
6.4.3	Recovery	190
6.5	Transactions in the Publication Environment	200

6.5.1	Transaction Execution	200
6.5.2	The Impact of Distribution	203
6.5.3	The Impact of Heterogeneity	208
6.6	Conclusion	212
7	THE ConTract MODEL	219
	<i>Helmut Wächter and Andreas Reuter</i>	
7.1	Introduction and Overview	220
7.2	Transaction Support for Large Distributed Applications	221
7.3	ConTracts	225
7.3.1	Modelling Control Flow: Scripts and Steps	227
7.3.2	ConTract Programming Model	230
7.3.3	Transaction Model	230
7.3.4	User Interface for Controlling Large Distributed Applications	233
7.3.5	Forward Recovery and Context Management	234
7.3.6	Consistency Control and Resource Conflict Resolution	240
7.3.7	Compensation	241
7.3.8	Synchronization with Invariants	246
7.4	Implementation Issues	250
7.4.1	Flow Management	251
7.4.2	Transaction Management	251
7.4.3	Logging	253
7.4.4	Synchronization	253
7.4.5	Transactional Communication Service	254
7.5	Comparison with Other Work	254
7.5.1	Structural Extensions	254
7.5.2	Embedding Transactions in an Execution Environment	255
7.6	Conclusions	256
7.7	Sample Script "Business Trip Reservations"	258
8	DYNAMIC RESTRUCTURING OF TRANSACTIONS	265
	<i>Gail E. Kaiser and Calton Pu</i>	
8.1	Introduction	267
8.2	Requirements	269
8.3	Programmed Transactions	272
8.3.1	Definitions	272
8.3.2	Nested Transactions	275

8.4	User-Controlled Transactions	277
8.5	Applications	281
8.5.1	Editing	281
8.5.2	Design Environments	281
8.5.3	Multi-User Design Environments	283
8.6	Implementation Issues	284
8.7	Comparison to Related Work	287
8.8	Conclusions	289
9	MULTIDATABASE TRANSACTION AND QUERY PROCESSING IN LOGIC	297
	<i>Eva Kühn, Franz Puntigam, and Ahmed K. Elmagarmid</i>	
9.1	Introduction	298
9.2	Representation of MDBS Queries in Prolog	299
9.2.1	Dynamic and Static Integration	300
9.3	Transaction Control with Logic Programming	317
9.3.1	The Flex Transaction Model	318
9.3.2	Parallel Logic Programming	320
9.4	Query and Transaction Processing in VPL	324
9.4.1	Architecture	324
9.4.2	Operational Semantics of the VPL Language	326
9.4.3	Mapping Transactions into VPL Queries	337
9.5	Extending the Power of Flex Transactions	341
9.6	Conclusions	343
10	ACTA: THE SAGA CONTINUES	349
	<i>Panos K. Chrysanthis and Krithi Ramamritham</i>	
10.1	Introduction	350
10.2	The Formal ACTA Framework	351
10.2.1	Preliminaries	352
10.2.2	Effects of Transactions on Other Transactions	356
10.2.3	Objects and the Effects of Transactions on Objects	360
10.3	Characterization of Atomic Transactions	368
10.4	Characterization of Sagas	372
10.4.1	A Special Case of Sagas	382
10.5	Variations of the Sagas Model	385
10.5.1	Sagas with no Special Relation with Last Component	385
10.5.2	Sagas with Vital Components	386
10.5.3	Sagas of Sagas	389