

# DATA MODEL PATTERNS

A Metadata Map



DAVID C. HAY

TP311.13  
H412

# DATA MODEL PATTERNS

## *A Metadata Map*

David C. Hay  
*Essential Strategies, Inc.*



E200603919



AMSTERDAM BOSTON  
HEIDELBERG LONDON  
NEW YORK OXFORD PARIS  
SAN DIEGO SAN FRANCISCO  
SINGAPORE SYDNEY TOKYO



MORGAN KAUFMANN PUBLISHERS

Publisher	Diane D. Cerra
Assistant Editor	Asma Stephan
Editorial Intern	Marisa Crawford
Publishing Services Manager	George Morrison
Project Manager	Marilyn E. Rash
Cover Design	Hannus Design
Cover Image	<i>Aristotle with a Bust of Homer</i> , 1653 (Oil on canvas) Rembrandt Harmensz van Rijn (1606–1669 Dutch) Metropolitan Museum of Art, New York, USA
Composition	Cepha Imaging Pvt. Ltd.
Technical Illustrations	Dartmouth Publishing Inc.
Indexer	Kevin Broccoli
Interior printer	Courier Westford
Cover printer	Phoenix Color Corp.

Morgan Kaufmann Publishers is an imprint of Elsevier.  
500 Sansome Street, Suite 400, San Francisco, CA 94111

This book is printed on acid-free paper.

© 2006 by David C. Hay. Published by Elsevier Inc. All rights reserved.

Designations used by companies to distinguish their products are often claimed as trademarks or registered trademarks. In all instances in which Morgan Kaufmann Publishers is aware of a claim, the product names appear in initial capital or all capital letters. Readers, however, should contact the appropriate companies for more complete information regarding trademarks and registration.

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, scanning, or otherwise—without prior written permission of the publisher.

Permissions may be sought directly from Elsevier's Science & Technology Rights Department in Oxford, UK: phone: (+44) 1865 843830, fax: (+44) 1865 853333, E-mail: [permissions@elsevier.com](mailto:permissions@elsevier.com). You may also complete your request on-line via the Elsevier homepage (<http://elsevier.com>), by selecting "Support & Contact" then "Copyright and Permission" and then "Obtaining Permissions."

#### Library of Congress Cataloging-in-Publication Data

Hay, David C., 1947-  
Data model patterns : a metadata map / David C. Hay.  
p. cm.

1. Data warehousing. 2. Metadata. I. Title.  
ISBN-13: 978-0-12-088798-9 (pbk.: alk. paper)  
ISBN-10: 0-12-088798-3 (pbk.: alk. paper)

QA76.9.D37.H38 2006  
005.74—dc22 2006011123

For information on all Morgan Kaufmann publications, visit our Web site at [www.mkp.com](http://www.mkp.com) or [www.books.elsevier.com](http://www.books.elsevier.com)

Printed in the United States of America  
06 07 08 09 10 5 4 3 2 1

Working together to grow  
libraries in developing countries

[www.elsevier.com](http://www.elsevier.com) | [www.bookaid.org](http://www.bookaid.org) | [www.sabre.org](http://www.sabre.org)

ELSEVIER

BOOK AID  
International

Sabre Foundation

# **DATA MODEL PATTERNS**

**The Morgan Kaufmann Series in Data Management Systems**

Series Editor: Jim Gray, Microsoft Research

- Data Model Patterns: A Metadata Map*  
David Hay
- Data Mining: Concepts and Techniques, Second Edition*  
JiaWei Han and Micheline Kamber
- Querying XML: XQuery, XPath, and SQL/XML in Context*  
Jim Melton and Steve Buxton
- Joe Celko's SQL for Smarties: Advanced SQL Programming, Third Edition*  
Joe Celko
- Moving Objects Databases*  
Ralf Güting and Markus Schneider
- Foundations of Multidimensional and Metric Data Structures*  
Hanan Samet
- Joe Celko's SQL Programming Style*  
Joe Celko
- Data Mining, Second Edition: Concepts and Techniques*  
Ian Witten and Eibe Frank
- Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*  
Earl Cox
- Data Modeling Essentials, Third Edition*  
Graeme C. Simsion and Graham C. Witt
- Location-Based Services*  
Jochen Schiller and Agnès Voisard
- Database Modeling with Microsoft® Visio for Enterprise Architects*  
Terry Halpin, Ken Evans, Patrick Hallock, Bill Maclean
- Designing Data-Intensive Web Applications*  
Stephano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, and Maristella Matera
- Mining the Web: Discovering Knowledge from Hypertext Data*  
Soumen Chakrabarti
- Advanced SQL: 1999—Understanding Object-Relational and Other Advanced Features*  
Jim Melton
- Database Tuning: Principles, Experiments, and Troubleshooting Techniques*  
Dennis Shasha and Philippe Bonnet
- SQL: 1999—Understanding Relational Language Components*  
Jim Melton and Alan R. Simon
- Information Visualization in Data Mining and Knowledge Discovery*  
Edited by Usama Fayyad, Georges G. Grinstein, and Andreas Wierse
- Transactional Information Systems: Theory, Algorithms, and Practice of Concurrency Control and Recovery*  
Gerhard Weikum and Gottfried Vossen
- Spatial Databases: With Application to GIS*  
Philippe Rigaux, Michel Scholl, and Agnes Voisard
- Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*  
Terry Halpin
- Component Database Systems*  
Edited by Klaus R. Dittrich and Andreas Geppert
- Managing Reference Data in Enterprise Databases: Binding Corporate Data to the Wider World*  
Malcolm Chisholm
- Understanding SQL and Java Together: A Guide to SQL, JDBC, and Related Technologies*  
Jim Melton and Andrew Eisenberg
- Database: Principles, Programming, and Performance, Second Edition*  
Patrick and Elizabeth O'Neil
- The Object Data Standard: ODMG 3.0*  
Edited by R. G. G. Cattell and Douglas K. Barry
- Data on the Web: From Relations to Semistructured Data and XML*  
Serge Abiteboul, Peter Buneman, and Dan Suciu
- Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*  
Ian Witten and Eibe Frank
- Joe Celko's SQL for Smarties: Advanced SQL Programming, Second Edition*  
Joe Celko
- Joe Celko's Data and Databases: Concepts in Practice*  
Joe Celko
- Developing Time-Oriented Database Applications in SQL*  
Richard T. Snodgrass
- Web Farming for the Data Warehouse*  
Richard D. Hackathorn
- Management of Heterogeneous and Autonomous Database Systems*  
Edited by Ahmed Elmagarmid, Marek Rusinkiewicz, and Amit Sheth
- Object-Relational DBMSs: Tracking the Next Great Wave, Second Edition*  
Michael Stonebraker and Paul Brown, with Dorothy Moore
- A Complete Guide to DB2 Universal Database*  
Don Chamberlin
- Universal Database Management: A Guide to Object/Relational Technology*  
Cynthia Maro Saracco
- Readings in Database Systems, Third Edition*  
Edited by Michael Stonebraker and Joseph M. Hellerstein
- Understanding SQL's Stored Procedures: A Complete Guide to SQL/PSM*  
Jim Melton
- Principles of Multimedia Database Systems*  
V. S. Subrahmanian
- Principles of Database Query Processing for Advanced Applications*  
Clement T. Yu and Weiyi Meng
- Advanced Database Systems*  
Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T. Snodgrass, V. S. Subrahmanian, and Roberto Zicari
- Principles of Transaction Processing*  
Philip A. Bernstein and Eric Newcomer
- Using the New DB2: IBM's Object-Relational Database System*  
Don Chamberlin
- Distributed Algorithms*  
Nancy A. Lynch
- Active Database Systems: Triggers and Rules For Advanced Database Processing*  
Edited by Jennifer Widom and Stefano Ceri
- Migrating Legacy Systems: Gateways, Interfaces, & the Incremental Approach*  
Michael L. Brodie and Michael Stonebraker
- Atomic Transactions*  
Nancy Lynch, Michael Merritt, William Weihl, and Alan Fekete
- Query Processing for Advanced Database Systems*  
Edited by Johann Christoph Freytag, David Maier, and Gottfried Vossen
- Transaction Processing: Concepts and Techniques*  
Jim Gray and Andreas Reuter
- Building an Object-Oriented Database System: The Story of O<sub>2</sub>*  
Edited by François Bancilhon, Claude Delobel, and Paris Kanellakis
- Database Transaction Models for Advanced Applications*  
Edited by Ahmed K. Elmagarmid
- A Guide to Developing Client/Server SQL Applications*  
Setrag Khoshafian, Arvola Chan, Anna Wong, and Harry K. T. Wong
- The Benchmark Handbook for Database and Transaction Processing Systems, Second Edition*  
Edited by Jim Gray
- Camelot and Avalon: A Distributed Transaction Facility*  
Edited by Jeffrey L. Eppinger, Lily B. Mummert, and Alfred Z. Spector
- Readings in Object-Oriented Database Systems*  
Edited by Stanley B. Zdonik and David Maier

To my mother, Henrietta Hay, who taught me to write well  
and to appreciate good writing.

*CHRONO-SYNCLASTIC INFUNDIBULA—Just imagine that your Daddy is the smartest man who ever lived on Earth, and he knows everything there is to find out, and he is exactly right about everything, and he can prove he is right about everything. Now imagine another little child on some nice world a million light-years away, and that little child's Daddy is the smartest man who ever lived on that nice world so far away. And he is just as smart and just as right as your Daddy is. Both Daddies are smart, and both Daddies are right.*

*Only if they ever met each other they would get into a terrible argument, because they wouldn't agree on anything. Now, you can say that your Daddy is right and the other little child's Daddy is wrong, but the Universe is an awfully big place. There is room enough for an awful lot of people to be right about things and still not agree.*

*The reason both Daddies can be right and still get into terrible fights is because there are so many different ways of being right. There are places in the Universe, though, where each Daddy could finally catch on to what the other Daddy was talking about. These places are where all the different kinds of truths fit together as nicely as the parts in your Daddy's solar watch. We call these places chrono-synclastic infundibula.*

*...*

*Chrono (KROH-no) means time. Synclastic (sin-CLASS-tick) means curved toward the same side in all directions, like the skin of an orange. Infundibulum (in-fun-DIB-u-lum) is what the ancient Romans like Julius Caesar and Nero called a funnel. If you don't know what a funnel is, get Mommy to show you one.*

*—Kurt Vonnegut Jr.*

*From The Sirens of Titan, © copyright 1988*

*by Kurt Vonnegut*

*Used by permission of Dell Publishing,  
a division of Random House, Inc.*

# PREFACE

## ABOUT METADATA

Twenty years ago, when I started working as a consultant with the Oracle Corporation, I learned a particular style of data modeling. I had done database design for many years before that, and often illustrated my designs with drawings. The particular flavor of modeling I learned at Oracle, however, was very different. For the first time, I was modeling the structure—the language—of a company, not just the structure of a database. How does the organization understand itself and how can I represent that so that we can discuss the information requirements?

Thanks to this approach, I was able to go into a company in an industry about which I had little or no previous knowledge and, very quickly, to understand the underlying nature and issues of the organization—often better than most of the people who worked there. Part of that has been thanks to the types of questions data modeling forces me to ask and answer. More than that, I quickly discovered common patterns that apply to all industries.

It soon became clear to me that what was important in doing my work efficiently was not conventions about *syntax* (notation) but rather conventions about *semantics* (meaning). This was the source of my first book, *Data Model Patterns: Conventions of Thought*. I had discovered that nearly all commercial and governmental organizations—in nearly all industries—shared certain semantic structures, and understanding those structures made it very easy to understand quickly the semantics that were unique to each.

The one industry that has not been properly addressed in this regard, however, is our own—information technology. This is at least partly because the patterns that address most businesses are not as helpful to the understanding this one. Where a business model represents the semantics of a business, what we need are



models that represent semantics itself. We need models of the models we use to describe the business. This is more difficult.

Our industry also has not been properly addressed for the same reason many companies do not have data models: we have not seen the need. Just as the idea of modeling an organization's data seems a little too arcane for many business people, so too the idea of modeling information technology data seems too strange for many of us. But the need is definitely there. Just as it is essential for an organization to better understand the underlying nature of its data (and through that the underlying nature of its own structure) if it is to acquire and use systems successfully to meet its customers' needs, so too is it essential for us to understand the underlying nature of our data (and through that the underlying nature of our industry's own structure) if we are to be successful in producing information systems products for our customers.

As you will see, some of the semantic patterns are in fact the same for information technology as they are for any other industry. Most notably, people and organizations are components of the information technology world, just as they are at the heart of any business. Similarly, locating programs and data in the information technology world is not that different from locating products and customers in the business world. Beyond these topics, however, the model in this book is very different from a typical commercial model. Whereas a business is concerned with modeling products and processes, our model is concerned with modeling the *concepts for describing* a product or process.

Aristotle called his work "Meta" physics, simply because it was the one he wrote after writing the one on physics, the word *meta* being Greek for "after". This book could also be about a "meta" model simply because I am creating it after years of creating business models. But it is more than that. Because of the strange nature of Aristotle's metaphysics, the word *meta* came to mean "above" or "beyond". Because of the strange nature of this model, I am sure no one will argue against applying the word in this more cosmic sense.

## **ABOUT THIS BOOK**

The "data" in *metadata* means that this description of our industry will be expressed as a data model. The concepts are presented here using semantic

data constructs. But data are not the sole *subject* of this model. The book is intended to be more *comprehensive* than prior efforts, in that it will cover more facets of our industry. Because this is a comprehensive view of metadata, a comprehensive view of the world is required. The book describes not just the structure of data modeling but also models of activities, people and organizations, locations, events and timing, and motivation.

Yes, those of you familiar with John Zachman's Framework for Enterprise Architecture will recognize these topics. They are the what, how, who, where, when, and why columns in his approach to understanding the body of knowledge that is the information systems development world. These columns indeed form the basis for chapters in the book.

While the model is intended to be comprehensive, by the way, I am acutely aware that it probably is not comprehensive enough. First, not all rows are covered. To model the builder's world (the fifth row) requires a model of every different brand of relational database, programming language, and new tool for addressing business rules and other areas. Those models alone would require several books this size. It seems reasonable, therefore, to start "small".

Second, as suggested by the title of the book, these really are just patterns for modeling metadata. This is not a comprehensive design for a "metadata repository". Rather, this model is an attempt to identify the most fundamental and widely applicable concepts that must be present in such a repository. I am acutely aware of the fact that if you are building a repository in a particular environment you will need more specific details in many areas. My only hope is that this model will make it clear where to add those details.

In addition to addressing the columns in John Zachman's Architecture Framework, this book addresses the different points of view taken by various people in the systems development process: the CEOs, the people who run the business, the information architects, the designers, the builders, and the users of systems. By addressing the different perspectives described in the framework, this book should be more *comprehensible* than previous efforts, as well. It describes metadata as seen by business owners, system architects, and designers—in their terms.

Because each row of the Architecture Framework described here represents a particular perspective, and the part of the model describing that row is presented in terms of language appropriate to that perspective, both business metadata and technical metadata are included. Each is intended to be readable and understandable by its intended audience. Moreover, the model is presented one small

piece at a time to ensure that the structures described can be understood by any educated—even if not technologically savvy—reader.

The subject of the book is a single *conceptual data model* (an *entity-relationship model*) of the metadata that control systems development and management. It is a conceptual data model in that it is a unified description of the business we are in, not of a specific database design. Indeed, it is not the design of a metadata repository at all, although it does describe what should be in such a repository, and any designer would be well advised to understand it thoroughly before taking on such a design. It is fundamentally a Row Three model.

This book itself uses a particular vocabulary (as close to educated English as possible) to describe the concepts contained here. One of the things described, however, is itself the idea of vocabulary. This means that the models used are themselves examples of what is being described. (For those who, in spite of your author's best efforts, do not find the meaning of the models intuitively obvious, all the entity classes and attributes presented are defined both in the text and in the glossary at the back of the book.)

When a company develops a data model of its operations, the model is a useful product for the development of a new database. The effort of producing the model itself, however, often reveals to the people involved profound insights into the nature of their business. These insights often represent a direct benefit to the enterprise, over and above any improved systems obtained from the model. It gives them the opportunity to understand the *implications* of what they do for a living—on their systems, their colleagues, and on the business as a whole.

So, what you have here is a model of the principal concepts behind what we do when we try to improve the information management of an organization. The interesting thing is that once we understand these concepts the major controversies that have plagued our industry for the last couple of decades (such as object-oriented versus relational, the entity-relationship diagram versus the UML class model, and so forth) become less heated. It turns out that there is no real disagreement about the merit of any particular technological change, but only on the perspectives of the contenders. Understand these differences of perspective, and the arguments disappear.

What the model in this book shows is just what such technological changes mean. Does this new tool change the way we write programs? Does it change the way we construct (or carry out) processes? Does it change how we analyze requirements? Correctly placing the technology in the framework goes a long

way toward understanding its significance—and, indeed, increases our ability to implement it effectively.

For example, UML has been trumpeted as a great innovation in modeling. It is true that it is more expressive in some areas than has been seen before. But it is important to understand what is really new about it and what is simply a new notation for things that can also be represented in other ways. The models in this book should make this distinction clear.

This book is intended for the data management community—data administrators, database administrators, data modelers, and the like. But it should also be useful to system developers, helping them to more readily understand both the meaning of what they are doing and where that fits into the larger scheme of things. It should also be useful in an academic setting for teaching any and all of these people. This may be asking a lot, but it would be valuable also if information technology managers at least understood the broad strokes of these models, again to ensure that they understand the context of what they are doing.

The model in these pages attempts to show the information processing world from many different perspectives. With luck, perhaps we actually have a chronosynclastic infundibula.

## **ACKNOWLEDGMENTS**

I must begin by expressing my thanks to Allan Kolber, who not only encouraged me throughout this effort but provided invaluable insights into the Zachman Framework, and in particular into the real meaning of the Business Owner's View. I still reserve the right to disagree with him on specifics, but his insights have been vital to this book.

And of course thanks go to my Business Rules Group colleagues who provided a wonderful place for the incubation of ideas on business rules and the Architecture Framework. Their movement to become collaborators in the Business Rules Team has been a significant step forward, and I appreciate the publication of the "Semantics of Business Vocabulary and Business Rules". I sincerely hope that this book can be a proper complement to that work. In particular, Cheryl Estep has spent many hours helping clarify where my work has diverged from the Business Rule Team's efforts.

Please note that I have borrowed extensively from early drafts of the BRT work. While I want to give them credit for members' contributions, any errors of interpretation or divergence from the eventual final draft are entirely my responsibility.

Thanks must also go to Bob Seiner for publishing *The Data Administration Newsletter* ([www.tdan.com](http://www.tdan.com)) faithfully for all these years. In addition to providing the world with a wonderful source of knowledge about all things data administration, it provided me with a wonderful vehicle for exploring the ideas that ultimately resulted in this book.

Thanks to Meiler Page-Jones for writing the best book I have found about object-oriented design [Page-Jones 2000]. It was clear enough to form the basis for my model of object orientation.

From ANSI to John Zachman, I am always indebted to all the authors in my bibliography (and others) who have formed the intellectual basis for anything I may have done here.

I am fortunate to have been able to present this model to various organizations. I want to thank The Texas Modeling and User Group in particular, and specifically Lee Leclair for his insightful comments. Also my thanks to the Information Resource Management Association of Canada for their hospitality; especially the comments of Dorothy Russel, Kerel Vitrofsky, and Deborah Henderson were most helpful. Comments from the Boston and Rochester DAMA chapters are also appreciated. The people attending the Metadata 2005 Conference in London were also helpful in showing me some remaining errors in the model.

None of this would have been possible without the help of Diane Cerra, my advocate at Morgan Kaufmann, and Marilyn E. Rash, the project manager/editor who put it all together. In addition, the quality of the book was immeasurably improved by the suggestions from all the people who graciously gave of their time to review the contents of the manuscript: James Bean, Charles Betz, Malcolm Chisolm, Michael Eulenberg, Dagna Gaythorpe, Terry Halpin, Mike Lynott, Dave McComb, Ken Orr, and Graeme Simseon.

And of course, as always, thanks to my wife, Jola, and to Pamela and Bob, for being there.

## FOREWORD

Marcel Proust wrote that the real voyage of discovery lay not in finding new landscapes, but in having new eyes. In the information systems field, parts of the landscape change so rapidly that we can easily overlook the value that can be found in fresh perspectives. This is particularly true of systems specification, where advances have more often come from new ideas than from new technologies.

Systems specification is traditionally discussed in terms of its processes. Neophytes are taught the stages and tasks; debate centers on the merits of the different methodologies. Yet information systems professionals learned a long time ago that although process analysis might be the most obvious approach to understanding a business, it is not the only one. Looking at data—the *what* rather than the *how*—not only complements the process perspective, but also arguably results in a deeper, more concise and more stable description. Data modelers regularly achieve insights that process modelers miss.

In this book, Dave Hay brings the perspective of an expert data modeler to the business of systems specification. His starting point is the Zachman Framework for Enterprise Architecture, which itself owes its longevity at least in part to a focus on artifacts rather than activities.

This book begins with some insightful and practical adjustments to the Framework. It is hardly surprising that after some twenty years, reflective users should have some suggestions for improvement, and it would be disappointing if purists saw these changes as anything other than a confirmation of the Framework's continuing relevance and adaptability. Dave then sets out, in considerable detail, the structure of the information used in each of the relevant cells. In keeping with his philosophy of data modeling, he seeks to describe rather than prescribe, to illuminate established approaches rather than proposing his own.

The result is a comprehensive picture of what constitutes an information systems specification, from business requirements to logical design. Dave calls it "an enterprise data model of the IT industry"—a description of the data that systems

professionals need to do their jobs. Whatever methodology we use, and regardless of whether we document the data or convey it informally, these are the things we need to know. For users of the Zachman Framework, the information here provides a clear “next level down”. For information systems practitioners, students, and academics in general, it offers a perspective likely to be of continuing value even as methods and technologies change.

The language of the book is data modeling—and this is one of its particular strengths. The use of data models, rather a reliance on text and examples, provides a level of completeness and rigor missing in much of the other work in the field. Dave’s background as a working data modeler is evident in the content and presentation of the models, and in the examples he uses to illustrate them. Readers of his earlier books on data model patterns and systems specification will know that he brings to the task a unique combination of subject matter knowledge and data modeling expertise.

Read this book, therefore, not only to gain a “new eye” on information systems specification, but also to see how an expert modeler uses the language of data modeling to represent and communicate a complex and important domain.

Graeme Simsion  
*Simsion & Associates and the  
University of Melbourne*

# CONTENTS

Preface .....	xiii
Foreword .....	xix
 <b>CHAPTER 1: ABOUT METADATA MODELS .....</b>	 <b>1</b>
WHAT ARE METADATA? .....	1
IN SEARCH OF METADATA .....	5
THE ARCHITECTURE FRAMEWORK .....	7
The Rows .....	8
The Columns .....	11
METAMODELS AND THE FRAMEWORK .....	13
Data .....	15
Activities .....	15
Locations .....	16
People and Organizations .....	16
Timing .....	16
Motivation .....	17
THE NOTATION: OBJECT AND ENTITY CLASSES .....	17
Class Model (UML) .....	18
Entity-Relationship Model .....	24
LEVEL OF ABSTRACTION .....	29
 <b>CHAPTER 2: DATA .....</b>	 <b>33</b>
DATA AND THE ARCHITECTURE FRAMEWORK .....	33
THE BUSINESS OWNER AND BUSINESS RULES .....	34
ROW TWO: BUSINESS TERMS, CONCEPTS, AND FACT TYPES .....	37
Terms and Concepts .....	38
Ontologies .....	45
Propositions .....	48
Fact Types .....	49
Characteristics .....	52
Resources .....	54



---

Authorship .....	57
Document Distribution .....	60
Coverage .....	60
The Dublin Core .....	63
Data Stewardship .....	65
ROW THREE: THE ENTITY-RELATIONSHIP DIAGRAM .....	65
Sample Entity-Relationship Model .....	66
Entity Classes and Attributes .....	67
Relationships and Relationship Roles .....	70
Entity Class Views .....	82
About Reference Data .....	86
ROW FOUR: DATA DESIGN .....	87
Relational Database Design .....	88
Object-Oriented Data Design .....	97
ROW SIX: THE PRODUCTION SYSTEM .....	115
 <b>CHAPTER 3: ACTIVITIES, FUNCTIONS, AND PROCESSES .....</b>	<b>123</b>
ACTIVITIES AND THE ARCHITECTURE FRAMEWORK .....	123
DEFINITIONS .....	125
TYPES OF PROCESS MODELS .....	127
ROW TWO: FUNCTIONS AND BUSINESS PROCESSES .....	129
Functions .....	130
Business Processes .....	133
Sample Process Model .....	138
Motivating Business Processes .....	140
Access Roles .....	140
ROW THREE: PROCESSING DATA .....	142
A Data Flow Diagram .....	144
System Processes and Data Flows .....	146
Access Roles and System Events .....	150
Essential System Processes .....	152
ROW FOUR: PROGRAM MODULES .....	157
Basic Module Structure .....	157
Program Roles .....	164
Object Modules .....	171
ROW SIX: PROGRAM INVENTORY .....	178
 <b>CHAPTER 4: LOCATIONS .....</b>	<b>183</b>
ABOUT LOCATIONS .....	183
Site .....	185
Geographic Location .....	189